# SOFTALK

VOLUME 3 NOVEMBER 1982 $3.00

## Thanksgiving Turkey –Raised on Apples

## A Dream of Walking

## Campaign '82

## Exec Software Publishing Corporation

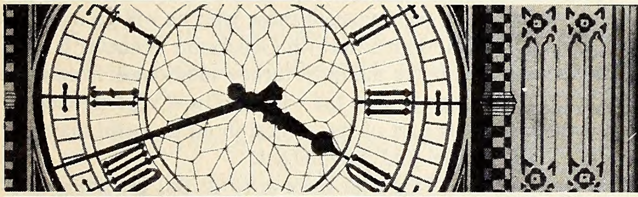
# SOFTALK



# FEATURES · DEPARTMENTS



# IN NEXT MONTH'S SOFTALK



Digging December ... *Softalk*'s Third Annual Holiday Gift Guide ... The Muppets and their friends at Henson Associates ... Exec: TG Products ... The Life of a Programmer for Hire ... The Cat (purring kind) House Apple ... Word Processing: *Word Juggler* ... Seven Flags ... and more ...

# INDEX OF ADVERTISERS

| | |
|---|---|
| Chairman | John Haller |
| Publisher | Al Tommervik |
| Editor | Margot Comstock Tommervik |
| Art Director | Kurt A. Wahlner |
| Managing Editor | Patricia Ryall |
| Associate Editors | Jean Varven |
| | Andrew Christie |
| Newspeak Editor | David Hunter |
| Special Assignments | Jonathan Miller |
| Open Discussion Editor | Tommy Gear |
| News and Trade Editor | Matthew T. Yuen |
| Reviews Editor | Roe Adams |
| Assistant Editors | Michael Ferris |
| | David Durkee |
| Proofreader | Harry McNeil |
| Contributing Editors | |
| *Assembly Language* | Roger Wagner |
| *Pascal* | Jim Merritt |
| *Business* | Peter Olivieri |
| *Apple CP/M* | Greg Tibbetts |
| *Apple III* | Taylor Pohlman |
| *Hardware* | Jeffrey Mazur |
| *Applesoft* | Doug Carlston |
| *Investing* | Kenneth Landis |
| *DOS* | Bert Kersey |
| *Graphics* | Mark Pelczarski |
| *Financial Modeling* | Joe Shelton |
| *Beginners* | Christopher U. Light |
| *Basic Solution* | Wm. V. R. Smith |
| Art Manager | Kevin McKeon |
| Art Production | Don Robertson |
| Art Assistants | Michael G. Pender |
| | Roxanne Rogers |
| | Jill Schultz |
| Associate Publisher | Mary Sue Rennells |
| Accounting | Evelyn Burke |
| Accounting Assistant | Mary Milam |
| Marketing | Jim Salmons |
| Marketing Assistant | Dave Fitzgerald |
| Assistant to the Publisher | Dan Yoder |
| Purchasing | Steven M. Poncar |
| Dealer Sales | Judy Weinstein |
| Book Sales | Lois Stickelmaier |
| Advertising Coordinator | Linda McGuire |
| West Coast Sales | Mike Antich |
| | Softalk |
| | 11160 McCormick Street |
| | Box 60 |
| | North Hollywood, CA 91603 |
| | (213) 980-5074 |
| East Coast Sales | Paul McGinnis |
| | Ian Ross |
| | Paul McGinnis Company |
| | 60 East 42nd Street |
| | New York, NY 10017 |
| | (212) 490-1021 |
| Midwest and Rocky Mountain Sales | Ted Rickard |
| | John Sienkiewicz |
| | John Bollweg |
| | Kevin Sullivan |
| | Market/Media Associates |
| | 435 Locust Road |
| | Wilmette, IL 60091 |
| | (312) 251-2541 |
| Circulation | |
| *Trial Subscriptions* | Hal Schick |
| | Dierdre Booth |
| *Paid Subscriptions* | Pam Kelley |
| *Back Issues* | Ron Rennells |
| *List Maintenance* | Pat Adams |
| | Harry Richardson |
| *Assistant* | Cliff Martinez |
| Systems | Sam Pawley |
| Guest Reviewers | Dave Albert, Jock Root. |

**Cover Photo:** Tom the Turkey, courtesy of John Johnson. Photo by Kurt Wahlner.

**Credits:** Composition by Photographics, Hollywood, California. Printing by Volkmuth Printers, Saint Cloud, Minnesota.


 ISSN: 0274-9629. *Saftalk* is published monthly by Softalk Publishing Inc., 11160 McCormick Street, North Hollywood, California; telephone (213) 980-5074. Second-class postage paid at North Hollywood, California, and additional mailing offices.

**Free Subscription:** Complimentary trial subscription to all owners of Apple computers in the USA. If you own an Apple, but you've never received *Saftalk*, send your name, address, and Apple serial number with a request for subscription to Softalk Circulation, Box 60, North Hollywood, CA 91603. *Saftalk* is totally independent of Apple Computer Inc.; sending your warranty card to Apple Computer will not inform *Softalk* of your existence.

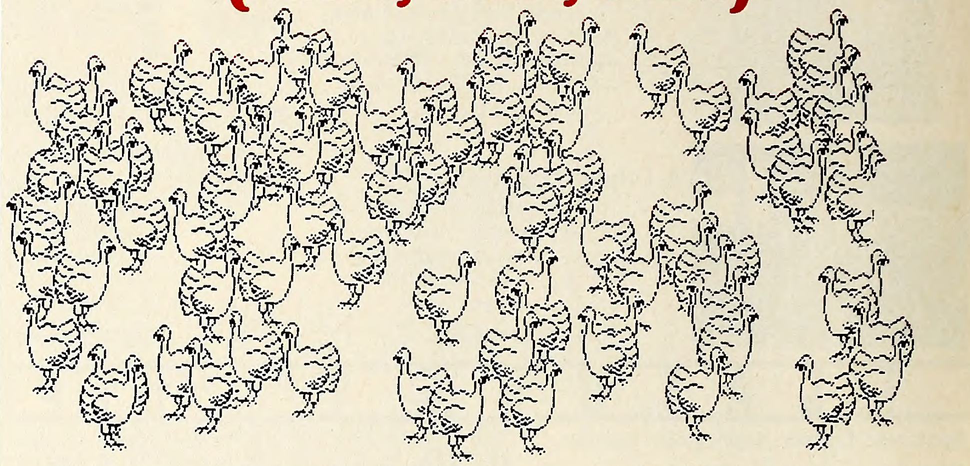
**Paid Subscription:** $24 per year without sponsor, $18 per year with sponsor. At the end of trial period, each subscriber will be notified; response is required only if you wish to continue receiving *Softalk*. Lack of response will be taken as your choice to discontinue the magazine. Special rates for schools and libraries, $8; multiple subscriptions for schools and libraries, $5 each.

**Back Issues:** $2 through February 1981; $2.50 through July 1981; $3.50 through September 1982; $4 thereafter. November and December 1980 and January, February, March, September, October, and November 1981 are sold out.

**Problems?** If you haven't received your *Saftalk* by the fifteenth of the month, or if you have other problems with your subscription, Hal Schick or Pam Kelley can help out. Call (213) 980-5074.

**Moving?** Send new address and old to Softalk Circulation, Box 60, North Hollywood, CA 91603; telephone (213) 980-5074.

# Contest: Poultry Panic! or A Time To Gather Gobblers (turn, turn, turn)



No, the cover of this issue has absolutely nothing to do with *Softalk* employees. Well, okay, maybe a little. You see, we were planning to have the First Annual Softalk Thanksgiving Banquet right before the Applefest in San Francisco at the end of this month. The guest list included everyone who works at *Softalk,* our columnists, and a few derelicts who hang out at the hamburger joint across the street.

If you know us like we know us, you know that we just love to eat. So, we went out to the local North Hollywood turkey farm and traded them some magazines and *BasiCalc* disks for a whole bunch of turkeys—enough for everyone to have at least two. An even trade, wouldn't you say? Then, *Softalk*'s construction department got right to work and built a turkey pen to hold the little gobblers until we were ready to eat.

For your viewing pleasure, here they are, all one hundred and. . . . Hey, wait a minute. Some of them have escaped! Most of them have escaped! Word has it that our trustworthy, yet hungry, writers saw some of them strutting around and began to hoard them, stashing most of them away in their columns. Boy, as if an invitation to this epicurean extravaganza wasn't enough!

If we can't recover the birds, we're all going to go hungry.

So, we're calling on you to give us some assistance. Help us round up the turkeys that got away, and you'll be handsomely rewarded. All you have to do is peruse the magazine and capture any stray turkeys you see aimlessly tiptoeing through the typefaces. Once you think you've caught them all, count them up and submit your tally to us.

Whoever catches all the turkeys will be eligible to win $100 in goods made by *Softalk* advertisers.

Here are the rules:

1. Any photographs of turkeys don't count. We're just looking for escaped hi-res roasters like the ones on this page.

2. Then, to make sure you have the right total, add the number of turkeys you captured to the number of turkeys on this page. We're looking for the total number of turkeys before the massive breakout.

3. In the case of more than one correct entry, the names of those who tie will go into the Random Sandwich Generator. Whoever is able to make the most turkey sandwiches from leftovers will be the winner. The RSG will oversee this part of the contest in all its awful majesty, cruel and magnificent.

That's all there is to it. So, get going and have a ball!

Name: ____________________

Address: ____________________

City, State, Zip: ____________________

Phone: ____________________

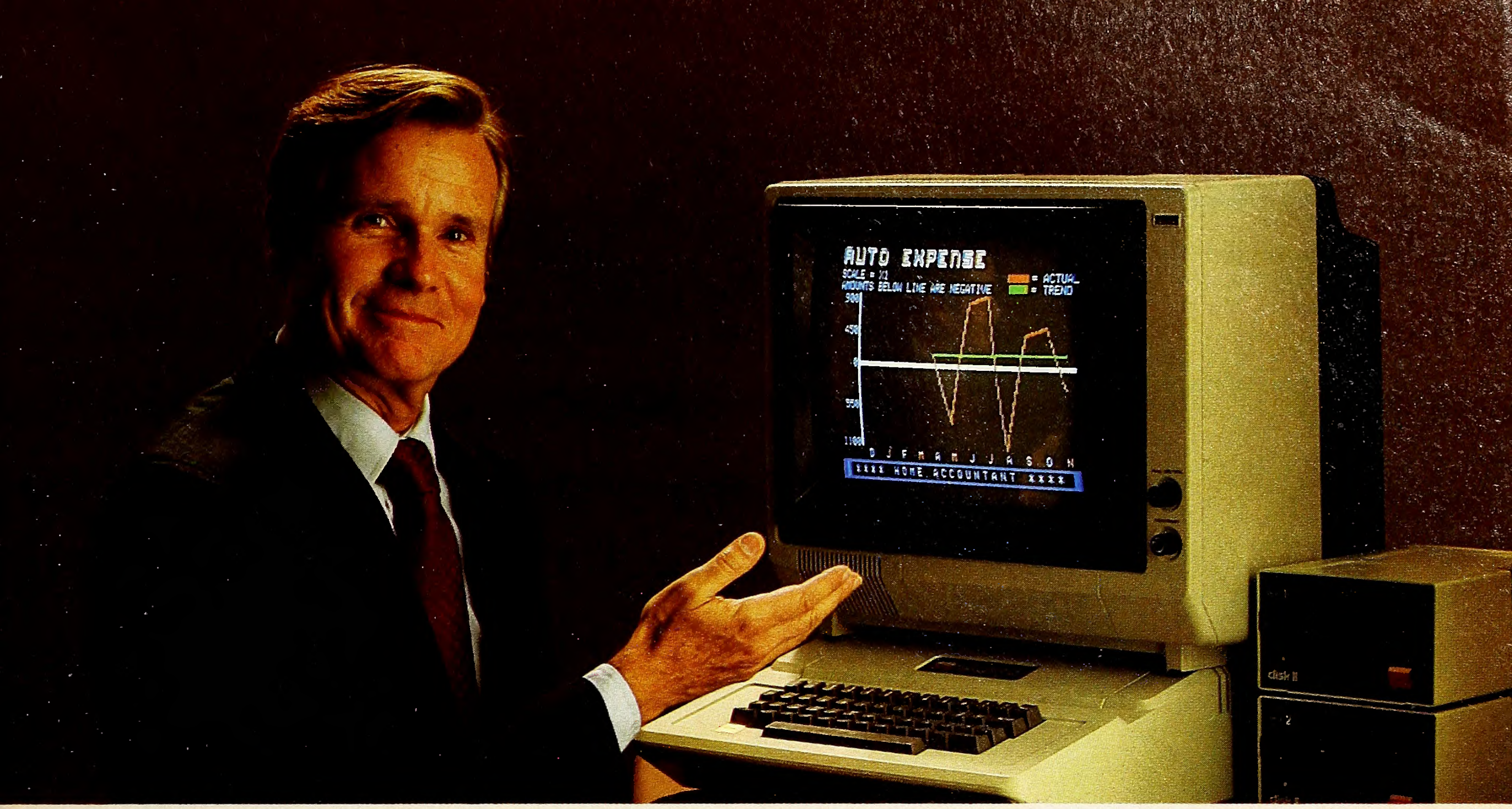
My dealer: ____________________

I counted this many turkeys: ____________________

If I counted correctly, or if I make the most sandwiches, I want: ____________________

with all the trimmings.

Send this entry form or a facsimile to Softalk Gobblers, Box 60, North Hollywood, CA 91603, by December 15, 1982 (the return address was going to be Softalk Turkeys, but then anyone here might have opened it up).




# CONTINENTAL'S HOME ACCOUNTANT IS NUMBER ONE— AND CLIMBING.

For the past several months, Softalk magazine has rated Continental Software's Home Accountant™ No. 1 in its "Home 10" best-seller list.

A lot of programs would have "peaked" by now. But with over 10,000 copies in use, Home Accountant just keeps getting stronger.

There's a reason for this spectacular success—and it's not just the low suggested price of $74.95.

The fact is, Home Accountant is one of those rare programs that virtually everybody can profit from using. It's powerful enough to handle even the most complicated family budget—yet it's so easy to use that one quick trip through the manual may be all you'll ever need.

With Home Accountant you can track up to 100 budget categories, 5 different checking accounts, and all the credit cards you can carry. Just press a few keys and watch the program print your checks, net worth and other financial statements. And when you see the full-color graphs of actual vs. budgeted expenses, trend line analyses, etc., you'll know you bought the best.

See your Apple dealer soon for a demonstration. And start watching your fortunes climb with Home Accountant*

**Continental**
Software

For your nearest dealer, contact us at: Continental Software, 11223 S. Hindry, LA, CA 90045 213-417-8031. Apple is a registered trademark of Apple Computers Inc. *Requires 48k and 1 disk drive. Printer optional.

# Contest Winners

**An Ounce of English Gold.** Eureka! That's what the old-timers yelled when they discovered gold in California more than one hundred thirty years ago. It's also what Jack English of Piscataway, New Jersey, will be thinking all the way to his computer store to pick up his prize for having won Part 4 of the Oracle '82 contest, predicting the closing price of gold on the New York market on September 7, the day after Labor Day.

And this was no small task; remember that all contestants were making their predictions more than eight months before the day. Predictions ranged from $159.46 to $1,372.63. Investors from Beverly Hills were clawing at *Softalk*'s door to get the names of those whose predictions came close. However, swinging joy-


That's Carl "Apple Trivia" Webb receiving his Disk II drive from Marty Potashnick of Computerland of Carlsbad, California. Carl later bought an Apple II Plus to go with his new drive.

sticks and hurling disk drives at them seemed to keep them at bay.

English's prediction of $482 was just a slim dollar above the actual closing price of $481. Only two others came that close, so English dragged them into the Random Number Generator with him. What happened inside would have made *The Texas Chainsaw Massacre* look like *Mister Roger's Neighborhood* and is unprintable here. Anyway, English emerged unscathed and was last seen clearing way for some TG paddles and a Select-a-Port from TG paddles and a Select-a-Port from TG Products.

The dust finally cleared, and one of the victims of the wrath of English (and the RNG) turned out to be Charles S. Lewis of Richmond, Virginia. *Softalk* contest fans will recognize Lewis immediately—he's the winner of June's Apples in History contest and the husband of Oracle '82 Part 1 winner Elizabeth Lewis (don't go away; C. Lewis will appear again in a few paragraphs).

The other RNG casualty was Daniel Tobias from Poughkeepsie, New York. Coincidentally, both Tobias and Lewis were a dollar under the closing price, while winner English was a dollar above. Perhaps the Random Number Generator knows . . . nawww.

**Close, but No Cigar.** Near-misses were Dale Culler (Ames, IA) with $483, Chris Grant (La Mesa, CA), who bought at $485, Rick Jones (Spokane, WA), who sold at $486, and Gary Geniesse (Osprey, FL), who tied with Kevin Park (Bloomington, MN) with predictions of $475. As usual, runners-up win a set of personalized *Softalk* InvisiTabs; Tobias and Lewis win a gold-plated set.

In the race for the grand prize of a new disk drive, field positions are always shifting. Charles Lewis (told you so) is still holding a commanding lead over everyone with an amazing score of −7.6 points. Hot on his tail are Tobias at −14.4 points, English with −20.7 points, and Jones, hanging on with −21.9 points.

Part 4 of the Oracle really spread the field out; the last place entry currently has −909.73 points. People who were neck and neck after Part 2 are now scattered throughout. But this is the *Softalk* Oracle contest, and anything can happen.

**Picking Kumquats.** This is absolutely the last time we ever run one of these creativity contests—at least until next year. These things are so hard to judge, and the contest staff can never agree on a winner. Here's how it worked this time:

The contest meister read all entries and selected the best twenty or so, based on originality, cleverness, entertainment value, and general niceness (neatness should have been a criterion, but if it were, lots of entries would have been disqualified. When we said double-spaced, we meant it).

Next, the entire staff selected their favorite five finalists in order of preference. Six points were given for each first place vote, four points for second, three points for third, and so on. Simple, right? Wrong. Even with such a point scheme, we still had tied scores. And we all know how contest ties are broken around here—the all-powerful Random Number Generator. When we approached the RNG, it told us it would have none of that. RNG said that breaking ties in creativity contests was not in its job description, and that it wanted a bonus and two months' extra vacation time for deciding the winner in this one.

Upon hearing this, the contest meister set the RNG on fire and began gouging it with pointed sticks. One hour and two citations from the North Hollywood Fire Department later, the contest staff gathered in the *Softalk* Kumquatorium where each member presented reasons why his favorite entry should win. Everyone was limited to ten minutes at the podium. At the end, whoever had the fewest tomatoes and overripe fruits thrown at him won.

**Winner.** Congratulations are in order to Bernie Mulaskey of Fairfax, California, because when the contest staffer who presented Mulaskey's case emerged from the Kumquatorium, he had only six tomatoes, three plums, and an eggplant on his coveralls.

This was the first *Softalk* contest ever entered by Mulaskey, a former inventor for Chevron Research. "I usually don't have time for these things, but when I saw the word 'creative,' I just couldn't resist."

Mulaskey has more than forty patents under his belt, mostly in the area of chemical catalysis for the oil industry. He uses his Apple at home and another one at work, both with CP/M systems. However, Mulaskey won't be choosing any CP/M software as his prize from Computerland of Marin, his local store. "I'm going to leave choosing the prizes to my daughter," he explains. "She rarely gets a chance to play with the Apple, so I'll let her choose her favorite games."

Here's Mulaskey's blue ribbon entry. For a unique approach, he took a typical inventor's angle—get the final product and work backward.

> Woz and Jobs really needed a name for their new company. Woz, in his usual left-brained manner, wrote down the first bigger name than IBM he could think of. "ABCDE will be our new name," he said.
>
> "That's not a name," replied Jobs. "At least put 'IBM' in the middle. Our new name is AIBME."
>
> Finally, using his right brain, Woz said, "That's too obvious. Let's change IBM to HAL like they did in the movie, *2001*. Our new name is AHALE. It almost sounds Hawaiian."
>
> They both liked this name, and the new company was born as AHALE Computer. In those days it was necessary to use the ASCII character code to enter their new name into the primitive AHALE computer. So, they input 65-72-65-76-69 in ASCII to spell their new

# SPY'S DEMISE

Arcade Action by Alan Zeldin

Somewhere on each floor of the Soviet diplomatic mission in Pyongyang are the nine parts of an encoded message. Your future is assured if you can just find those pieces and put them together, and then solve the puzzle. But to do so you must avoid the embassy guards who make frequent rounds at unscheduled intervals. They don't ask questions first, either.

# PIE-MAN™

by Eagle Berns and Michael Kosaka

You got a late start looking for that summer job, and all you could find was a baker apprentice position at the Automated Bakery Company. Simple enough, since the pies are made by machine . . . all you have to do is add topping and put the pies away when they come out on the conveyor belt. Shouldn't be too difficult of a summer, you think to yourself . . .

# TRANSYLVANIA

A High Resolution Graphic Adventure

Crafted by Antonio Antiochia

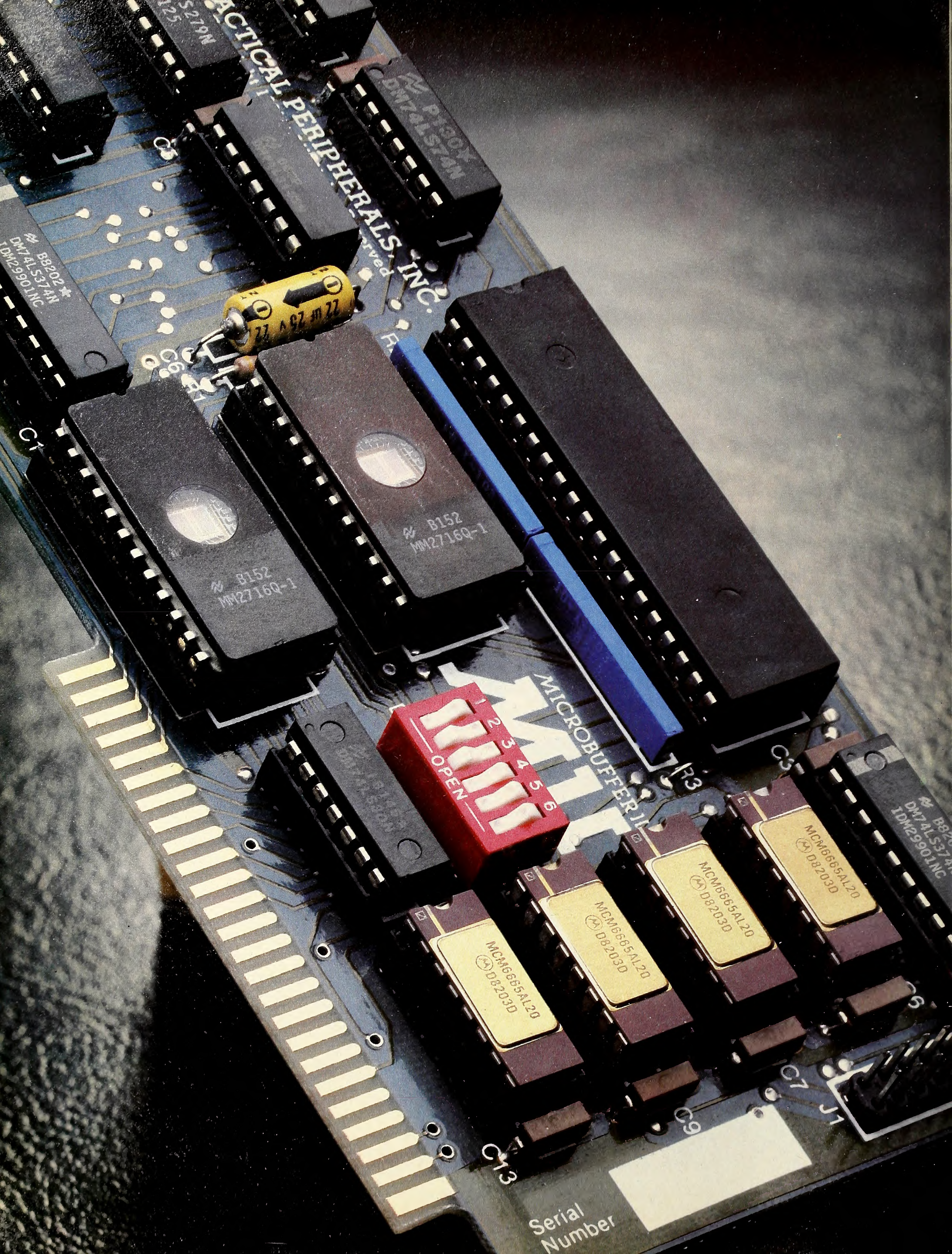
Transport yourself to the dark forests of Transylvania, where mystery lurks behind every towering tree, and venture to rescue a damsel in distress. Transylvania uses over one hundred colors and the finest graphics ever seen in a high resolution adventure to present a true challenge and hours of enjoyment to all adventurers.

Above games now available for the Apple computer. Arcade games work with keyboard, joystick, or Atari joystick. Graphics for all above created with the aid of The Graphics Magician.

## penguin software

the graphics people

830 4th Avenue    Geneva, Illinois 60134    (312) 232-1984



ACTICAL PERIPHERALS, INC.
MICROBUFFER
OPEN
B152
MM2716Q-1
MCM6665AL20
D8203D
Serial Number

# MICROBUFFER WILL SPEED UP ANY PROGRAM THAT REQUIRES PRINTING.

### MICROBUFFER ALLOWS YOU TO PRINT AND PROCESS SIMULTANEOUSLY.

Now you don't have to wait for the printer to finish before you can use your computer again.

### YOU CAN DUMP PRINTING DATA DIRECTLY TO MICROBUFFER.

Unlike your printer, Microbuffer accepts data as fast as your computer can send it. So there's never a bottleneck.

Microbuffer first stores the data in its own memory buffer and then takes control of your printer. This frees the computer for more productive functions.

Additional output may be dumped to the buffer at any time and it will be printed in turn.

### THERE IS A MICROBUFFER FOR ANY PRINTER/COMPUTER COMBINATION.

Microbuffers are available in Centronics-compatible parallel or RS-232C serial versions.

FOR APPLE II COMPUTERS, Microbuffer II features on-board firmware for text formatting and advanced graphics dump routines. Both serial and parallel versions have very low power consumption. Special functions include Basic listing formatter, self-test, buffer zap, and transparent and maintain modes. The 16K model is priced at $259 and the 32K, at $299.

FOR EPSON PRINTERS, Microbuffer is $159 in either an 8K serial or a 16K parallel version. The serial buffer supports both hardware handshaking and XON-XOFF software handshaking at baud rates up to 19,200. Both Epson interfaces are compatible with all Epson commands including GRAFTRAX-80 and GRAFTRAX-80+.

ALL OTHER PRINTER/COMPUTER COMBINATIONS are served by the in-line, stand-alone Microbuffers. Both serial and parallel versions are expandable up to 256K. The serial stand-alone will support different input and output baud rates and handshake protocol. The 32K model starts at $299, 64K for $349. 64K add-ons for up to a total of 256K are just $179.

When you think of how much time Microbuffer will save, can you afford to *not* have one?

### SIMPLE TO INSTALL.

Microbuffer II is slot-independent. It will fit directly inside the Apple II in any slot except zero.

Microbuffer for your Epson mounts easily in the existing auxiliary slot directly inside the printer.

The stand-alone Microbuffer is installed in-line between virtually any printer and computer.

### MICROBUFFER FROM PRACTICAL PERIPHERALS.

Practical Peripherals is dedicated to establishing new industry standards for product performance.

The un-retouched photo at left has been enlarged to demostrate Microbuffer's exact workmanship and precise attention to detail. Specifications demand that each board undergo 36 seperate tests and inspections before it can leave the factory.

Ask your dealer for a demonstration of the most practical, most successful new product of the year — Microbuffer.

PRACTICAL PERIPHERALS, INC.
31245 LA BAYA DRIVE
WESTLAKE VILLAGE, CA 91362
(213) 991-8200

See Us At
COMDEX™
**Nov. 29-Dec. 2, 1982**
Las Vegas Convention Center
Las Vegas, Nevada

name, AHALE.

"Wait a minute!" shouted Jobs. "I can use my right brain, too. Let's put the name of our 6502 chip into AHALE. That would be really clever. We should name the company 65 65 02 76 69 in ASCII."

"Dumb again, Jobs," Woz protested. "Anyone knows that would spell AA(ctrl-B)LE. We can't even print the control character. But we know that our 6502 chip is twice as good as two Z-80 chips or any 8080, so let's use 80 80 instead of 65 02."

"Great idea again, Woz," agreed an enthusiastic Jobs. "The new company will now be known as 65 80 80 76 69, translated as APPLE."

And so to this day, what might have been AHALE Computer is called "Apple" because of the Great Wizard of Woz.

Good going, Bernie. A real kick in the ASCII.

**Place and Show.** Yes, it was an ugly scene amidst flying tomatoes, and everyone agreed that all the finalists should have won. But remember, our contestants don't like to share prizes. Here are the other finalists in no particular order.

Andy Ihnatko of Westwood, Massachusetts, was our man in the field covering the Great Penguin War which pitted the King's United Military Quarry for Upper Antarctic Terrestrials (KUMQUAT) against a scrappy and scrawny Agency for Penguin Parity of Landed Edibles (APPLE). Ihnatko defined the war as a great big bicker over which group of penguins had rights to a limited supply of herring, the Antarctic equivalent of Twinkies.

APPLE, it seems, was fighting for personal liberty, while KUMQUAT was fighting for sheer gluttony. Flipper-to-flipper combat ensued and, according to Ihnatko, the death toll rose to forty penguins within a month. The following is a firsthand account of how the altercation was resolved:

Pelnius Penguin, the APPLE colony's genius, was putting the finishing touches on his latest invention. The machine attracted all the other APPLEs because of the lovely pictures it drew and the pretty sounds it made. Since old Pelnius was an APPLE, and therefore kindhearted, he let KUMQUAT play with the invention also. Soon, the entire colony was using the machine and playing games that Pelnius had written for it, and no one had any time to fight.

Ihnatko tells us that the war was settled by the leaders of each side playing *Flight Simulator*, *Time Zone*, and *Sabotage*. The APPLEs won, and Pelnius named his invention after the ragtag band of diehards.

Finalist Bruno Lanvin from Brooklyn, New York, offers the *French Connection,* or *Explication Francaise*:

P ourquoi pomme? Why Apple? Who knows that?
Qui sait c A?
O range would be too much;
orange en dirait tro P
M ais, poire ferait un bide;
but pear would be a flo P
M elon would be fatal.
Melon serait fata L
E t, donc, pourquoi pas pomme?
So, well, why not Appl E?

Tres bien, Bruno. Vous avez gagne des InvisiTabs.

During all the tomato flinging, we found ourselves a poet in Marsha Hague, who lives in Rochester, Minnesota. Not only did her entry possess more esthetic quality than just a few rhyming words, but it was the only Woz-'n'-Jobs story that took place before the Apple's creation. And here it is:

On lovely Erin long ago, of times
Which only legends tell, a Siren cast
her charming spell. With voice like silv'ry chimes
she sang, and glis'ning dew-dropped shamrocks cried
while unicorns that frolicked stopped and sighed . . .
A gentle rain began to fall, aglow
with scintillating shards of sunlit hues
that spawned a rainbow; most intense its blues!
Then two enchanted strangers found their way
across a magic threshold's mist of time.
At rainbow's end they found a pot sublime!
'Twas filled with apples! Knowing smiles, the two
exchanged. Their search was done and home they flew
while shouting, "Here we'll build a plant someday!"

The *Softalk* Grammy goes to Bruce Hahne of Iowa City, Iowa, who sent in not only the words to his entry, *A Star Is Born* or *What to Name the Baby*, but also enclosed full sheet music with piano accompaniment. The contest staff had a good time singing in an unintended sixteen-part harmony (there was only one part); if you'd like to sing, too, it goes to the tune of *The Ship Titanic*. Hahne points out that the *Titanic* in no way reflects the Apple or its quality. Here's part of it:

Oh, they built a small computer with a 6502.
They knew they had a system that the people would love, too.
But they scratched their heads in vain when they tried to think up a name
A star was born when Apple got its name.

Chorus:
Oh, it was new,
and it was good
and its chips were working like they should (like they should).
They had worked 'til it was done, and now finally it would run.
A star was born when Apple got its name!

They argued over "Lemon," considered "Cherry," too.
Said Steve Jobs, "We'll decide it before this day is through—
for computers without names are like a micro without games."
A star was born when Apple got its name.
(Chorus)

At ten o'clock there entered Jobs's great aunt of eighty-three
sipping cider through a straw as the case would happen to be,
but while coming through the door, she spilled her juice on the motherboard.
A star was born when Apple got its name.
(Chorus)

"Don't mind that spill!" cried Wozniak, "I swear it must be fate!
We'll call it 'Apple,' " he decided, but then his aunt hollered, "Wait!
'Apple' will not win, for it caused mankind's first sin!"
A star was born when Apple got its name.
(Chorus)

Said his aunt, "I wouldn't use it; it's sure to be your bane."
"Never mind," smiled Wozniak, " 'Apple' it remains."
"I'm sure that fate won't lie—it'll be the apple of our eye!"
A star was born when Apple got its name.
(Chorus)

Very few tomatoes were flying when the contest staff reviewed the entry sent in by Shawn R. Smith of Greendale, Wisconsin:

A few years back, two men were put on assignment to build a means of transportation for veteran cavalrymen. Their idea was to build a twenty-four-foot-tall horse with a large passenger area in the torso.

After a few weeks of hard labor, they finally finished it. They named it Jake. Jake was a computerized horse, even though he acted as if he were alive. He also had a lot of extras such as AM-FM radio and air conditioning.

When he was unveiled, the cavalrymen cheered, hollered, and wanted a ride. Jake was more than delighted.

After a few days of transporting peo-

GOTO 272

We got a bizarre phone call the other day. Remember last month's "Contest That's Out of This World"? Apparently only a handful of people came close to the right answer to the game part of the contest by the deadline and the Beznardians, offworld authors of a game distributed by Southwestern Data Systems, were incensed. "Who are these humans, who can't understand simple Gluebedopey? We demand that you continue this contest until someone gets it right. If not, we will turn everyone in Silicon Gulch into semi-sentient gnats." What could we do? Certainly not doom all those high-tech residents to a life of semicoherent buzzing in the wind and fog. So we decided to make the game part two separate contests. The first one closes on the original date of October 31. The second closes on November 30. The prize for both contests is $100 worth of software or hardware from your local dealer.


# To Do It Right.

## To make a Winchester disk for just Apple®II.

That's how we set out to design our hard disk for the Apple®II. To understand the needs of serious users and programmers, and to correct the errors of our predecessors.

You want user-friendliness. So we wrote hard disk versions of Apple DOS, CP/M® and Pascal that are highly user-friendly and loaded with useful features.

- ☐ Auto-boot hard disk
- ☐ Menu-driven utilities
- ☐ Single-keystroke program execution in DOS
- ☐ Disk search with wild cards, and many more

You want flexibility. So we made every feature variable.

- ☐ From 1 to 16 operating systems on each disk
- ☐ Operating system spaces grow as needed
- ☐ Slot independence
- ☐ Variable size volumes (to 400K in DOS)
- ☐ Mountable and unmountable volumes (even in CP/M)
- ☐ Custom DOS allowed

You want reliability. We use Seagate drives only. And Corona's unique data protection technology.

- ☐ DataGuard™ 32-bit error correction code
- ☐ FailSafe™ read-after-write and automatic bad-sector reallocation
- ☐ 2-level impact-protection packaging

You want low-cost backup. So we wrote backup utilities that make floppy backup convenient.

- ☐ File compacting to reduce the number of floppies
- ☐ Volume selective backup in all operating systems
- ☐ Date selective backup in Pascal
- ☐ Automatic diskette sequencing to ease floppy handling

You want compatibility. We maximize compatibility with existing software and peripheral cards.

- ☐ 9K interface card leaves main memory untouched
- ☐ Interrupts are allowed
- ☐ Boot protected floppies from slot 6 without removing hard disk
- ☐ Automatic slot/drive to hard disk remapping

And you want support. We do that right too.

- ☐ Hardware depot service in every region
- ☐ Programmer's guides at each level
- ☐ Software theft-protection on the hard disk

Compare the features that matter to you. And visit our local dealer or distributor for a demonstration. You'll see the difference specialization makes.

---

**Corona Starfire™— The Winchester Disk for Apple II**

**$2495 / 5 MB     $2995 / 10 MB**

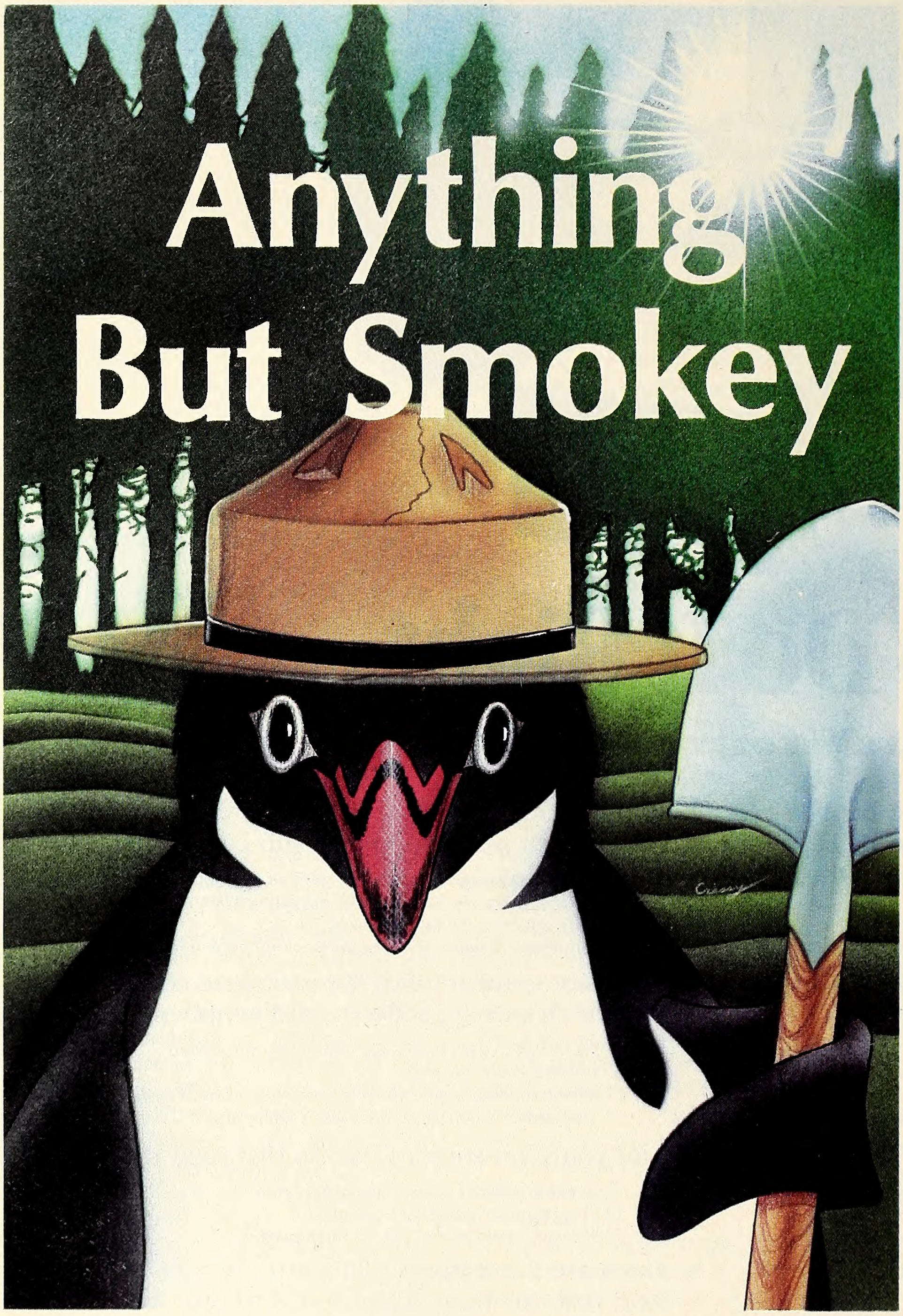
(All software included. Pascal not needed for CP/M.)

**corona™**

*The Third Generation Microcomputer Company*

31324 Via Colinas   Section 110   Westlake Village   CA 91361   (213) 706-1505

# Anything But Smokey



**"Remember, only you can prevent software piracy."**

For more than thirty years, a real bear named Smokey reigned as the mascot of the United States Forest Service and as the conscience of campers all over the country. He was well qualified to speak against forest fires: he was a victim of one as a cub, which is how the service found him. The original Smokey is gone, but his image remains to remind us about the disasters we can help prevent.

Just such a mascot, just such a lovable conscience, seemed the ideal reminder to us as computerists to help keep the fire of program piracy under control.

It seemed only natural to the people at Penguin Software, concerned as they have been with the issue of piracy and innovative as they've been in dealing with it, that their already lovable company pet, the penguin, should carry the standard for its prevention.

**What's in a Name?** Such is the genesis of the software piracy penguin. All he lacks now is—and here's where you come in—a name. Obviously, it can't be Smokey.

What would you name the friendly poster penguin?

Send as many choices as you like for a name (male or female—darned if we can tell), but each entry must be on a separate card or piece of paper, along with your own name, address, and phone number.

All entries will be judged by the staff of Penguin Software, who's also providing the prizes. If you'd like to tell why you chose the name you did, you may well influence the judges. On the other hand, if your name is perfect, it will win even if you don't explain it.

**Why Bother.** The thinker-upper of the winning name will get the very first "??? the Penguin" T-shirt produced, plus $100 worth of software from any *Softalk* advertiser. Ten runners-up will receive the second through tenth "??? the Penguin" T-shirts.

Mail your entry by December 15, 1982, to Softalk Bear, Box 60, North Hollywood, CA 91603. And remember, No-Name Penguin says, "Only you can prevent contest failure."

*This contest is sponsored and will be judged by Penguin Software, Geneva, Illinois.*

At Spinnaker Software, we make educational games that are actually fun.

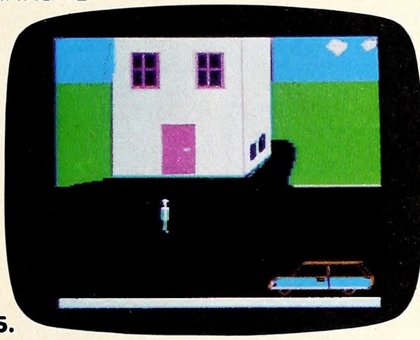
Because they're fun, your children will use them. Instead of letting them collect dust in the basement.

And because your children use them, they'll be learning. And after all, isn't that what educational games are all about?

**Our games are educational, because you can't kid parents.**

As a parent, you're probably very concerned with how much time your kids spend playing mindless video games.

Sure, they're fun. But they don't do much more than develop reflexes and hand-eye coordination. Spinnaker games are different.



All our games have true educational value. They help develop a child's learning skills. And that's something your kids can take with them wherever they go.

**Our games are fun, because you can't kid kids.**

Kids like Spinnaker games for the same reasons they like roller coasters, going to the beach and ice cream sundaes.

They're fun. Lots of fun. So much fun your kids will probably forget they're learning.

Our games make the computer screen come to life. With colorful graphics, animation and sound.

And they're easy to use. In fact, a lot of our games are easy enough for kids who've never even used a computer before.

**How do we make our games both educational and fun?**

We're glad you asked.

Educators and game programmers write our software.

Educators, because they've been in the classroom and know how children learn. And what it takes to keep their interest.

Game programmers, because they know how to have fun with computers. These programmers give our games the high resolution graphics, animation and sound that make them so entertaining.

And right now, we're introducing four new games that can be played on the most popular computers, Apple,® Atari,® and IBM.®

First, there's FACEMAKER. It's for young computer users, kids ages 4-8. FACEMAKER helps children improve memory and concentration and provides familiarity with the computer.

Another game for young users is STORY MACHINE. This game lets children ages 5-9 write their own stories and see them acted out on the screen. STORY MACHINE helps children learn to write correctly and acquaints them with the keyboard.

Our SNOOPER TROOPS™ detective series gives your child mysteries to solve. As a Snooper Trooper, your child will have to do some daring detective work, including crawling through dark houses and talking to mysterious agents.

Designed for kids ages 10 and older, SNOOPER TROOPS helps children learn to take notes, draw maps, classify information, and develops vocabulary and reasoning skills.

All four games are available in stores today.

With Spinnaker products, you can rest easy knowing your children are spending their time wisely.

So ask your retailer about the growing line of Spinnaker games.

Because one of the smartest things parents can do is help their children learn.

SPINNAKER™

We make learning fun.

# FASTALK

Fastalk is your quick guide to popular, specialized, or classic software. Programs appearing in Fastalk must meet one or more of the following criteria: (1) equal or surpass in sales the least-selling program to appear on any of the current bestseller lists; (2) relate to a specialized subject area and be in general distribution (more specialized packages and areas will be included as Fastalk matures); (3) be new and of professional quality (such programs will be carried for one month only—after that, they must meet other criteria for inclusion); (4) stand out as extraordinary.

Designation as a classic is noted by a bullet preceding a program's title.

Where opinion is expressed, *Softalk* has seen the software in question; the date of *Softalk*'s review, if any, is given at the end of the item.

*Softalk* may arbitrarily omit any package from Fastalk, whether or not it meets the foregoing criteria.

## Adventure

● **Adventure.** Crowther, Woods. The original text adventure, created on mainframe, contributed to by many over a long time. Very logical within fantasy framework, excellent puzzles, maps; complex, convoluted, and great. Several publishers: Microsoft, 10700 Northup Wy., Bellevue, WA 98004. $28.95. Apple, 10260 Bandley Dr., Cupertino, CA 95014. $35. Frontier Computing, Box 402, 666 N. Main, Logan, UT 84321. $10.

**Ali Baba and the Forty Thieves.** Smith. Fanciful Arabian Nights role-playing game with a sense of humor. Fresh, fast action, challenging options, and secrets that are a joy to discover. Quality, 6660 Reseda Blvd., Ste. 105, Reseda, CA 91335. $32.95 *7/81.*

**Cyborg.** Berlyn. Text adventure with brief action skill game hidden in plot. As a futuristic cyborg, you're lost in a strange forest, desperately needing food and power. In its realism and use of true plot, it represents one of the most significant advances in adventuring since the original *Adventure*. Sentient, Box 4929, Aspen, CO 81612. $32.95. *11/81.*

**Deadline.** Blank, Lebling. Episode one in a projected series of murder mysteries by the authors of *Zork*. Interrogate, accuse, make transcripts. Includes inspector's casebook, lab report. Infocom, 55 Wheeler St., Cambridge, MA 02138. $49.95. *8/82.*

**Escape from Rungistan.** Blauschild. A vacation with a vengeance. Get out of jail, battle snakes, bears, and cannibals; acquire skills to get your money refunded. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95. *8/82.*

● **Hi-Res Adventure #1: Mystery House.** Williams. Whodunit in a Victorian mansion. First adventure with pictures. Vocabulary of more than 300 words. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $24.95.

**Hi-Res Adventure #2: The Wizard and the Princess.** Williams, Williams. Attempt to rescue princess from vengeful wizard. Features 250 illustrations in full color. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $32.95. *11/80.*

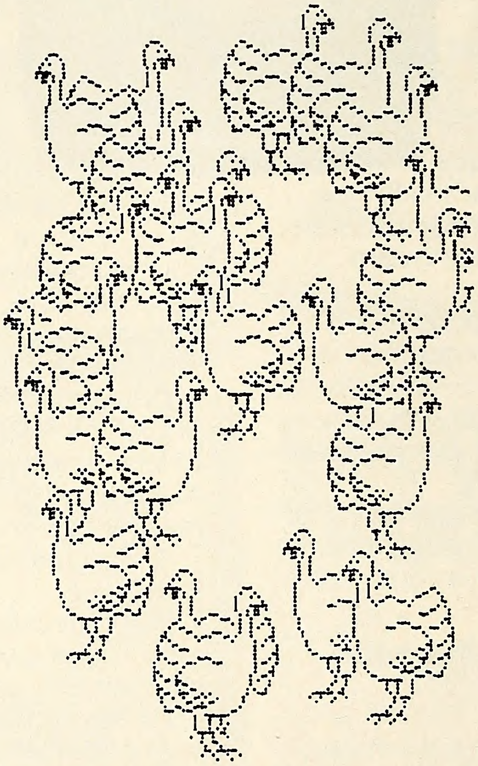
**Hi-Res Adventure #3: Cranston Manor.** DeWitz, Williams. More full-color adventuring involving the redistribution of wealth. Long on great riddles, short on plot. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $34.95. *9/81.*

**Hi-Res Adventure #4: Ulysses and the Golden Fleece.** Davis, Williams. Re-creation of the Greek legend, featuring graphics advances and ability to communicate with the characters. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $34.95. *12/81.*

**Kabul Spy.** Wilson. Cold War espionage adventure in which you must slip into Afghanistan to rescue a physicist before the commies make him talk. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $34.95

**Mask of the Sun.** A unique animated graphic quest with unusual full parsing. See everywhere you can go as you travel, watch things transform. A professional-looking graphics breakthrough with nice puzzles. Ultrasoft, 24001 S.E. 103rd St., Issaquah, WA 98027. $39.95.

● **The Prisoner.** Mullich. Superb TV series captured in computer game. Escape from an island requires player to solve logical puzzles, overcome obstacles, and answer riddles. Excellent computer fare; nothing else like it. Edu-Ware, Box 22222, Agoura,CA 91301. $29.95. *3/81.*



**Prisoner II.** Mullich. Totally relandscaped version of original game: hi-res graphics added, puzzles reworked, obstacles expanded. Sophisticated and difficult exercise in intimidation with elements of satire. Edu-Ware, Box 22222, Agoura, CA 91301. $32.95. *10/82.*

**Queen of Phobos.** Hi-res treasure hunt. Outwit four opponents on derelict ship in space. Looters after your cookies, too. Phoenix, 64 Lake Zurich Dr., Lake Zurich, IL 60047. $34.95.

● **S.A.G.A. Series.** Adams. Scott Adams's prototypical adventures—twelve in all—spruced up with 100-color graphics and Votrax vocals. Fun, not always logical, very story-oriented series. First to make chance a significant element of play (you can get killed a lot). Each adventure has its own theme; you do a lot of exotic traveling. They map small but score big on imagination. Adventure Intl., Box 3435, Longwood, FL 32750. $29.95 each.

**Starcross.** Science fiction prose adventure that comes wrapped in a flying saucer. In the year 2186, your mission to harness a black hole takes some unexpected turns. Likeable, engaging. Infocom, 55 Wheeler St., Cambridge, MA 02138. $49.95.

**Swordthrust Series.** Set of adventures, seven so far, that integrate fantasy role playing. Create one character, make new friends in each adventure, battle monsters and achieve goals together. Good stories, fun to map. Vocabulary no mystery but puzzles are. Single character goes through all. CE Software, 801 73rd St., Des Moines, IA 50312. Number 1 prerequisite for rest. Each adventure, $29.95. *8/82.*

**Time Zone.** Williams, Williams. "Microepic" hi-res adventure featuring ten periods from past and future history all over world and universe on eight double-sided disks. Good puzzles, many dangers. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $99.95. *1/82.*

**Transylvania.** Antiochia. Best graphics ever in a hi-res adventure. Excellent puzzles and logic — no tricks. Enjoyable. Penguin, 830 4th Ave., Geneva, IL 60134. $34.95. *10/82.*

**Zork.** Lebling, Blank. Part one of mainframe adventure; understands complete compound sentences and questions. Simultaneous manipulation of objects. Text. Infocom, 55 Wheeler St., Cambridge, MA 02138. $39.95. *6/81.*

**Zork II.** Lebling, Blank. *Zork* comes into its own in sequence. Great text adventure technique and communication. Infocom, 55 Wheeler St., Cambridge, MA 02138. $39.95. *3/82.*

**Zork III.** Lebling, Blank. Text lives! A masterpiece of logic and a grand adventure to revel in. Hard, logical puzzle with unique point system. Infocom, 55 Wheeler St., Cambridge, MA 02138. $39.95. *9/82.*

## Business

**Accounting Plus II.** Software Dimensions. Integrated package: general ledger, accounts receivable and payable, and inventory-purchasing modules. Basic and machine language. Menu-driven; prompting. Systems Plus, 1120 San Antonio, Palo Alto, CA 94303. $1,250.

**Apple Plot.** Converts numerical data into graphs; stores on hi-res page or prints out. *VisiCalc* interface. Apple, 10260 Bandley Dr., Cupertino, CA 95014. $70.

**Asset Manager.** Calculates depreciation using current balance; chooses depreciation representing greatest savings. Handles up to 999 assets. Micro Lab, 2310 Skokie Valley Rd., Highland Park, IL 60035. $200.

**BPI System.** Popular five-module business package; programs also available separately. Includes general ledger (a bestseller), accounts receivable, accounts payable, payroll, inventory control, and job costing. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $395 each. Job costing: $595.

**Business Plus.** Interactive package for service-type companies. With full-reporting general ledger (takes up to 250 items), accounts receivable, and accounts payable. Does two-year bar graphs. Advanced Operating Systems, 450 St. John Rd., Ste. 792, Michigan City, IN 46360. $399.

**Computer Programmed Accountant.** Five-module package: general ledger (very popular), accounts receivable, accounts payable, payroll, and property


MONSTER MASH
R.I.P.
R.I.P.
R.I.P.
R.I.P.
R.I.P.
ware

# the game control you've been waiting for

# TRAK BALL

$64.95

TG Products, the oldest and largest manufacturer of accessories for personal computers, brings you fantastic control and versatility with the new TRAK BALL control. You will be amazed at how accurately, quickly and easily you can position screen objects in advanced arcade-level games.

TRAK BALL is designed for years of reliable, active play and is produced of the highest quality components. Models are available now for your Atari*, Apple II*, and IBM* personal computers. Ask for them at your dealer or order direct from TG Products, 1104 Summit Ave., Suite 110, Plano, Texas 75074. Telephone (214) 424-8568.

*Atari, Apple and IBM are registered trademarks of Atari Inc., Apple Computers, Inc. and International Business Machines Corporation, respectively.

TG PRODUCTS

Allow two weeks for shipment.
All mail orders add $2.00 for postage and insurance.
Texas residents add 5% sales tax.

management. All other modules post automatically to general ledger. Continental, 11223 S. Hindry Ave., Los Angeles, CA 90045. $1,495. Separate modules: $250 each, except property management: $495.

**Creative Financing.** Evaluates loans and investments, provides R-O-I projections, payment tables, and objective decisions. Howard Software, 8008 Girard Ave., Ste. 310, La Jolla, CA 92037. $195.

**Datadex.** General-purpose database manager able to perform specific applications. File generation and report utilities allow definition of file structure and appearance of reports. Information Unlimited, 281 Arlington Ave., Berkeley, CA 94707. $150. *9/81.*

**The Data Factory.** Passauer. Database management system allows listing files, getting file statistics, selecting another file, transferring records to new database, and adding fields to update forms. Disk swapping required; excellent product overall. Several compatible products available. Micro Lab, 2310 Skokie Valley Rd., Highland Park, IL 60035. $150. *8/81.*

**Data Perfect.** Assembly language database companion to *Letter Perfect*; compatible with lower case in 40-column, most 80-column boards. Lay out, revise own screen, record design. Excellent built-in editor; ability to be edited by word processor. Searches, sorts, generates reports. LJK, Box 10827, St. Louis, MO 63129. $99.95.

**Data Reporter.** Allows plotting of data in various charts and graphs; stores data segmented by up to thirty-five fields. Machine language search and sort. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $220.

**dBase II.** Speedy relational database management system. Requires SoftCard. Ashton-Tate, 9929 Jefferson Blvd., Culver City, CA 90230. $700.

**DB Master.** Comprehensive database management system with password protection, extensive report creation options. Up to 1,020 characters per record. Stoneware, 50 Belvedere St., San Rafael, CA 94901. $229. *10/81.*

**DB Master Utility Pak I.** Compatible with version III. Translates *DB* files to Apple text, restructures existing files, replicates and merges, and recovers crashed files. Stoneware, 50 Belvedere St., San Rafael, CA 94901. $99.

**DB Master Utility Pak II.** Accessory disk with label printer, global editor, file merge, reblocker, and forms printer. Stoneware, 50 Belvedere St., San Rafael, CA 94901. $99.

**Desktop Planner.** Models and analyzes budgets, profits and losses, sales forecasts, cash flow; "what if?" calculations. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $250.

**Dow Jones News and Quotes Reporter.** With modem, checks latest financial news and stock quotes for more than 6,000 securities from local Dow Jones data bank. Apple, 10260 Bandley Dr., Cupertino, CA 95014. $95. *2/82.*

**Executive Briefing System.** Nifty business graphics package for preparing color slides, graphs, and charts. Lotus, 55 Wheeler St., Cambridge, MA 02138. $199.

**1st Class Mail.** Schoenburg, Pollack. Fantastically user-friendly program for specialized database applications. Twelve fields, ability to sort and filter on any field or combination. Continental, 11223 S. Hindry Ave., Los Angeles, CA 90045. $74.95. *6/82.*

**General Manager.** Database program that allows economic projections, search and select options, and screen formatting for data entry. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $99.95.

**Information Master.** Database management program that can keep records sorted in five separate orders simultaneously. High Technology, Box 14665, Oklahoma City, OK 73113. $150.

**Infotory.** Complete purchase order and inventory system for under 9,999 items of one type. Prints receiving, sales, purchase orders; audit trails available. SSR, 1600 Lyell Ave., Rochester, NY 14606. $295.

**Inventory Manager.** Inventory-only data management system. Thirteen categories; allows for retail-only sales. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $149.95.

**List Handler.** List-lover's delight. Prints lists, labels, and letters. Handles up to 3,000 records per disk and eight disk drives. Takes requests. Silicon Valley Systems, 1625 El Camino Real, Ste. 4, Belmont, CA 94002. $89.95.

**MicroFinesse.** Pascal-based spreadsheet from England. Handles models of up to 5,000 cells, makes automatic what-if? calculations. Easy to use. Osborne/McGraw-Hill, 630 Bancroft Wy., Berkeley, CA 94710. $495. *7/82.*

**Paymaster.** Payroll package that handles up to 100 employees. Accesses any data elements, keeps checks on file; variable deductions, fill-in-the-blanks tax tables. Masterworks, 1823 W. Lomita Blvd., Lomita, CA 90717. $275.

**Personal Filing System.** Page. User controls data in totally unstructured database. Up to thirty-two pages (screens) of information in each record. Software Publishing, 1901 Landings Dr., Mountain View, CA 94043. $95. *10/80.*

**PFS:Graph.** Chin, Hill. Works alone or interfaces with *PFS* databases and *VisiCalc* files. Produces bar, line, and pie charts merging data from several sources. Software Publishing, 1901 Landings Dr., Mountain View, CA 94043. $125.

**PFS:Report.** Page. Powerful report generator designed for use with *PFS*. Sorts, calculates, totals, formats, prints presentation-quality columnar reports. Software Publishing, 1901 Landings Dr., Mountain View, CA 94043. $95. *10/81.*

**Systems II EX.** Fully integrated, eleven-module business accounting package. Sorts and updates accounts: general ledger, payroll, inventory. Optional modules. Westware, 2455 S.W. 4th St., Ontario, OR 97914. $1,495.

**VC-Manager.** Chapman. *VisiCalc* utility enabling performance of arithmetic operations on up to fifteen models at once and addition of one model to another. Micro Decision Systems, Box 1392, Pittsburgh, PA 15219. $65.

**VersaForm.** Business forms generator for invoicing, mailing lists, sales analysis, inventory. Hard disk compatible. Applied Software Technology, 15985 Greenwood Rd., Monte Sereno, CA 95030. $389.

**VersaPlot.** Graph and chart maker. Combines file handling, data editing, and graph plotting in a fast, easy-to-use operating system. SpectraSoft, 350 Lantana, Ste. 775, Camarillo, CA 93010. $99.50.

• **VisiCalc.** Bricklin, Frankston. Electronic worksheet for any problem involving numbers, rows, and columns. No programming necessary. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $250. *10/80.*

**VisiCalc Formatting Aids.** Four programs any *VisiCalc* user would welcome: label splitter, formula reader, print-file reader, and variable-width reader. Data Security Concepts, Box 31044, Des Peres, MO 63131. $44.95. *9/82.*

**VisiFile.** Creative Computer, Jameson, Herman. Database management system for organization and retrieval of information, allowing sort and modification of records. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $250.

**VisiSchedule.** Critical path PERT schedule planner. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $300.

**VisiTran.** Use to create Basic exec files to transfer variables to *VisiCalc*. Requires some Applesoft programming. ADC Associates, 960 San Antonio Rd., Palo Alto, CA 94303. $99. *8/82.*

**VisiTrend/VisiPlot.** Kapor. Combines *VisiPlot* graphics with time-series manipulation, trend forecasting, and descriptive statistics. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $259.95. *7/81.*

# Communications

**ASCII Express II.** Blue. Modem software provides automatic redial, individual macro files, and improved file transfer capabilities. Sends any DOS file; uploads one character or one line at a time. Included utilities convert Integer Basic, Applesoft, or binary programs into text files. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $79.95. *9/81.*

**ASCII Express: The Professional.** Greatly improved version of the original. Supports multiplicity of hardware and prints simultaneously. Southwestern Data, Box 582, Santee, CA 92071. $149.95.

**Data Capture 4.0.** Copiable, modifiable smart terminal program; compatible with Apple III and most lower-case adapters. Southeastern Software, 6414 Derbyshire Dr., New Orleans, LA 70126. $65.

**Hello Central.** Menu-driven modem software. Upload-download, send-capture, save, retrieve, edit and manipulate files and programs. Advanced Operating Systems, 450 St. John Rd., Ste. 792, Michigan City, IN 46360. $99.

**Micro/Courier.** Electronic mail program. Provides file transfer of any DOS 3.3 file (correspondence, *VisiCalc*, charts) automatically and unattended. Built-in text editor; maintains 100 mailboxes; permits optional clock and calendar scheduling. Microcom, 1400A Providence Hwy., Norwood, MA 02062. $250.

**Micro/Terminal.** Access any in-house or remote database, set up and log only once. Built-in editor or edit off-line. Microcom, 1400A Providence Hwy., Norwood, MA 02062. $84.95.

**P-Term: The Professional.** Supports all Pascal-compatible interfaces, asynchronous serial cards, Many people think just because a computer does difficult things, it must be difficult to operate. Not so. At least not with PFS™ and PFS: REPORT.

PFS and PFS: REPORT let you concentrate on doing your work, not on working your computer. There's no complicated programming or computer language to learn. Using the PFS Series is so straightforward, you can learn it in as little as 20 minutes, and that's something both experts and beginners can appreciate.

**Design your own management system, simply.**

PFS lets you arrange your information in "forms" you design yourself. Once you've created your form—with spaces for all the necessary data—you simply fill in the blanks.

If you don't like the original form, PFS lets you change *just* the form, without having to totally re-enter the information on it.

To use your forms, just call up the information you want in a variety of ways such as "less than," "more than," "equal to," "between" or "not." You can even find specific characters, words, names or numbers. And PFS can also sort labels, change its 40-column format to accommodate larger printouts, and tell you how much room you have left on your disk.

**Get more out of what you put in.**

PFS: REPORT sorts, calculates, totals, formats and prints *presentation-quality* reports with up to 9 columns, in alpha or numeric order, with automatic entering and justification. With your information in this format, you can analyze it quickly to help spot trends, plan and make decisions.

**A whole family of software for Apple® II and III.**

PFS consistently ranks among the 5 top-selling business software packages for Apples. Because you can use it for almost anything—from chemical analyses to keeping track of your wine collection, from balancing your checkbook to computing your payroll. And there are more programs on the way.

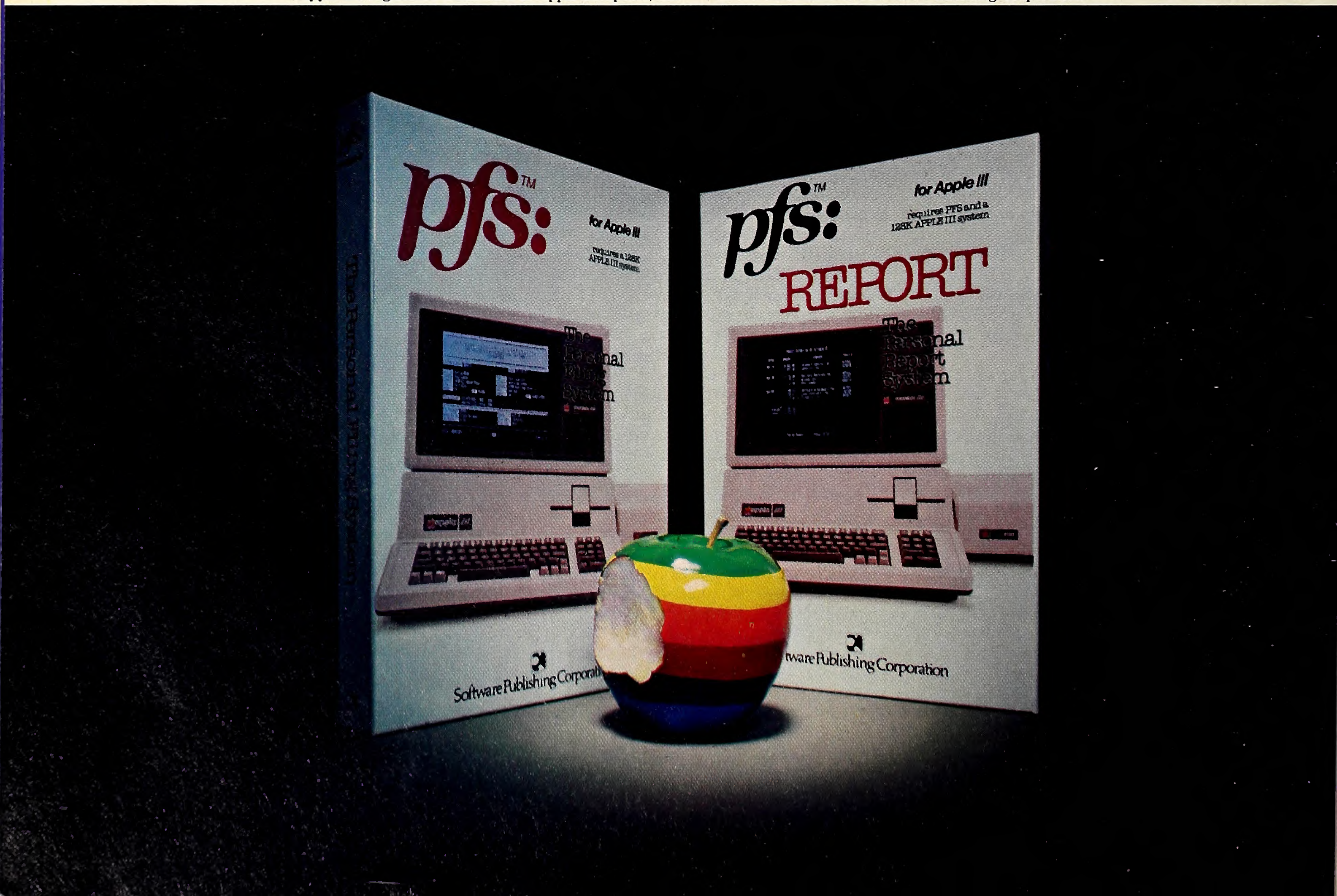
For a free, hands-on demonstration of PFS and PFS: REPORT, stop by your local computer store. If they don't have it yet, tell them to call Software Publishing Corporation at (415) 962-8910. Or write 1901 Landings Dr., Mountain View, CA 94043.

You'll be amazed at how much you can bite off.

# Hardworking software for Apples that isn't a lot of work.

**pfs: Software Series**

®Apple is a registered trademark of Apple Computer, Inc. ™PFS is a trademark of Software Publishing Corporation.



Apple-compatible modems, and baud rates up to 2,400. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $129.95.

**Super Smart.** Terminal emulation package to capture, create, edit, print, and save data. Utilizes full capabilities of Hayes Micromodem II; supports full ASCII. Softspoken, Box 7000-863, Redondo Beach, CA 90277. $60.

**Transend I, II, and III.** Intelligent terminal software with multiple hardware compatibility. Advanced, easy to use. The *I* sends text only; menu driven, limited editor. The *II* sends text and files like *VisiCalc*; verifies transmission. The *III* does both and handles electronic mail with auto-redial, clock calendar, and password protection. Upgrade for only $20; all three get an A+ for error handling. SSM, 2190 Paragon Dr., San Jose, CA 95131. $89, $149, $275. *9/82.*

**VisiTerm.** Well-planned, comprehensive. Hi-res sixty-character display; wide range of protocols for sending text. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $129. *9/81.*

**Z-Term.** Blue. Flexible, customizable communications software written specifically for the CP/M Apple. A quality package. Southwestern Data, Box 582, Santee, CA 92071. $99.95. *5/81.*

**Z-Term: The Professional.** More than an update. Compatible with a great variety of modems, interface cards, and screen modes. Simple file transfer with integrity. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $149.95.

## Fantasy

**Apventure to Atlantis.** Clardy. The sequel and worthy successor to *Odyssey*. Many refinements including recruitable entourage of wizards with individual attributes. Included cheat sheet is invaluable. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $40. *6/82.*

• **Beneath Apple Manor.** Worth. The original dungeon game for the Apple, created in 1978. Even in lo-res, it still stands up. Quality, 6660 Reseda Blvd., Ste. 105, Reseda, CA 91335. $19.95.

**Curse of Ra.** Expansion module to (and requires) *Temple of Apshai*. Find the magic treasure guarded by the demons of Ra; overcome the curse. Epyx/Automated Simulations, 1043 Kiel Ct., Sunnyvale, CA 94086. $19.95.

**Danger in Drindisti.** Expansion module to (and requires) *Hellfire Warrior*. Find the pattern to the glass wizard's maze; steal his magical staff. Epyx/Automated Simulations, 1043 Kiel Ct., Sunnyvale, CA 94086. $19.95.

**Knight of Diamonds.** Second scenario of *Wizardry*, requiring thirteenth-level characters from the original. Individual quests on each of six dungeon levels. Great. Sir-tech, 6 Main St., Ogdensbury, NY 13669. $34.95. *7/82.*

• **Odyssey: The Compleat Apventure.** Clardy. Fantasy adventure far beyond one place and one setting. Castles, catacombs, an ocean voyage, and the orb of power. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $30. *10/80.*

**Taipan!** Canfil. Roam the China Seas as an opium smuggler in this exotic fantasy with a challenging hi-res pirate sequence. Avalanche, 2460 Embarcadero Way, Palo Alto, CA 94303. $39.95. *10/82.*

• **Temple of Apshai.** Lead title in Dunjonquest series, winner 1981 Academy of Adventure Gaming Arts and Design "Computer Game of the Year" award. Epyx/Automated Simulations, 1043 Kiel Ct., Sunnyvale, CA 94086. $39.95

**Ultima.** British. Hi-res color adventure, progressing from Middle Ages to beyond the space age. A masterpiece. California Pacific, 1615 5th St., Davis, CA 95616. $39.95. *6/81.*

**Ultima II.** British. Faster play in a bigger universe with a time-travel option. Typically British look and feel. Events are much more interdependent; larger realm of fantasy with more transactions available. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $59.95.

**Upper Reaches of Apshai.** The next four levels (and requires) *Temple of Apshai*. Discover the secret of the monastery, battle giant tomatoes and killer chickens. Epyx/Automated Simulations, 1043 Kiel Ct., Sunnyvale, CA 94086. $19.95.

• **Wilderness Campaign.** Clardy. First fantasy game to leave the dungeon for the great outdoors; first in hi-res; first to bargain with merchants; and more. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $17.50.

**Wizardry.** Greenberg, Woodhead. Ultimate role-playing fantasy; ten-level maze in hi-res. Generate twenty characters, six at a time on expeditions. Gripping game; superbly produced. Sir-tech, 6 Main St., Ogdensburg, NY 13669. $49.95. *8/81.*

## Graphics

**Accu-Shapes.** Generates Apple shape tables. Uses lo-res to shape and edit, displays in hi-res. Accent, 3750 Wright Pl., Palo Alto, CA 94306. $49.95.

**Alpha Plot.** Kersey, Cassidy. Hi-res graphics and text utility with optional xdraw cursor and proportional spacing. Beagle Bros, 4315 Sierra Vista, San Diego, CA 92103. $39.50.

**The Animator.** Creates elaborate animated titles that you can add to your own programs. Machine language run. BalbeSoftware Systems, #6 White Plains, St. Louis, MO 63017. $49.95. *10/82.*

**Apple World.** Projects and rotates 3-D color images on screen in true perspective, drawing up to 65,000 points per side. Includes screen-oriented text editor for image formation. United Software of America, 750 3rd Ave., New York, NY 10017. $59.95.

**The Arcade Machine.** Jochumson, Carlston. Step-by-step arcade game designer—shapes, scoring, sound, and titles. Begin with variations on five games included, then on to your own. Broderbund, 1938 4th St., San Rafael, CA 94901. $59.95.

**Ceemac.** Boering. Visual composition language. Compose-execute-compose swapping by single key commands. Interpreter released as *Fire Organ.* Vagabondo Enterprises, 1300 E. Algonquin, Ste. 36, Schaumburg, IL 60195. $75.

**The Complete Graphics System II.** Pelczarski. A wealth of graphics tools at a reasonable price. Make 2-D drawings with game paddles, add text in destructive, nondestructive, or reverse modes, create 3-D figures with a panel module, and shape tables with a shape module. Manual features complete outline of command structure. Penguin, 830 4th Ave., Geneva, IL 60134. $69.95; Apple Graphics Tablet version, $119.95. *7/81.*

**Game Animation Package.** Bredon, Kampschafer, Clardy, Conley. Arcade game utility with two programs: one creates pictures for hi-res color adventure games; the other uses bit-map graphics to create title pages. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $49.95.

**GPS.** Versatile graphics program. Creates, manipulates, and edits images like a word processor. Easy to use; in standard and professional formats. Stoneware, 50 Belvedere St., San Rafael, CA 94901. $59.95, $99.99.

**GraForth.** Lutus. A graphics language rewritten for maximum speed. Plotting, line, text display, character image, and high speed 3-D graphics, with variety of colors and drawing options. Includes music synthesizer. Insoft, 10175 S.W. Barbur Blvd., Ste. 202-B, Portland, OR 97219. $75. *8/82.*

**Graphics A2-3D1.** High-speed 3-D animation package to guide beginner through scene creation, storage, retrieval, movement, and advanced applications. SubLogic, 713 Edgebrook Dr., Champaign, IL 61820. $59.95.

**The Graphics Magician.** Jochumson, Lubar, Pelczarski. Outstanding animation package consisting of a picture editor and shape table extender designed to allow programmers to design and store graphics files. Comes with utility program to transfer binary files. Penguin, 830 4th Ave., Geneva, IL 60134. $59.95; Apple Graphics Tablet version, $69.95. *5/82.*

**LPS II.** Superb hi-res graphics drawing system with light pen. Draw freehand or use circles and lines to create geometric shapes. Fill routine with colors and patterns; fun animation demo; programmable Pentrak driver. Gibson, 406 Orange Blossom, Irvine, CA 92714. $349.

**The Poor Man's Graphics Tablet.** Easy graphics utility that traces transparencies from the screen, drafts, edits shapes, and assembles scenes. Over fifty-nine textures and even more colors; single-key commands. Rainbow, 19517 Business Center Dr., Northridge, CA 91324. $49.95.

**Special Effects.** Pelczarski. Artist's graphic package for creating and enhancing computer graphics. With 108 colors and 96 brushes, magnification and editing point-by-point. Reverse colors, create mirror images, move images around. Penguin, 830 4th Ave., Geneva, IL 60134. $39.95.

**Zoom Grafix.** Holle. Graphics printing utility allows display of picture on screen prior to print; prints out selected portion at any size. Phoenix, 64 Lake Zurich Dr., Lake Zurich, IL 60047. $39.95. *2/82.*



## Home-Arcade

**ABM.** Atomic war high jinks. Defend the East Coast from Russian nuke attack. Incoming warheads can do splits. Muse, 330 N. Charles St., Baltimore, MD 21201. $25.

**Alien Ambush.** Basic shoot-'em-up with a difference: targets split when you hit them and you're in double trouble. Micro D, 17406 Mt. Cliffwood Circle, Fountain Valley, CA 92708. $24.95.

● **Alien Rain (Apple Galaxian).** Suzuki. Monsters in this home-arcade classic seem to take it personally when you gun down one of their kind. Broderbund, 1938 4th St., San Rafael, CA 94901. $24.95. *2/81.*

**Apple Panic.** Serki. Rid a five-story building of crawling Apples and butterflies by running up and down connecting ladders, digging traps in floors, then covering critters over before they devour you. Extremely addictive, excellent hi-res play. Broderbund, 1938 4th St., San Rafael, CA 94901. $29.95. *9/81.*

**Bandits.** Ngo. Fight off waves of multiple menaces intent on killing you and stealing your supplies. Delirious nonstop action, animated to the hilt. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $34.95.

**Beer Run.** Turmell. Artesian's delight. Catch falling cans on your way up one building, hop the blimp, and work your way down another. Sirius, 10362 Rockingham Dr., Sacramento, CA 95827. $29.95. *1/82.*

**Bug Attack.** Nitchals. Sing along with dagger-wielding ants, blue worms, swarming med-flies, a millipede, the 1812 Overture, lots of bright colors, terrific hi-res animation, and bouncy style. Cavalier, Box 2032, Del Mar, CA 92014. $29.95. *11/81.*

**Bug Battle.** Garden-variety shoot-'em-up that requires careful weeding and no fear of spiders. United Software of America, 750 3rd Ave., New York, NY 10017. $22.50.

**Cannonball Blitz.** Lubeck. In the cold light of *dawn,* you must find the *key* to victory, no matter how in*congr*uous. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $34.95. *7/82.*

**Choplifter.** Gorlin. Fly your chopper into the Bungeling Empire to rescue the sixty-four hostages, avoiding interceptor jets, homing mines, and tanks. Challenging, realistic, and playful. Broderbund, 1938 4th St., San Rafael, CA 94901. $34.95. *7/82.*

**County Fair.** Illowsky. Shooting gallery with hungry ducks and multiplying rabbits. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $29.95.

**Crazy Mazey.** Skill and strategy needed in this auto-chase maze game. Not flashy but enduring fun; jazzy sound. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $29.95. *10/82.*

**Crisis Mountain.** Schroeder. Run, crawl, walk, and leap through mountain maze fraught with rolling rocks, geysers, and chasms; collect nuclear devices. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $34.95. *10/82.*

**Crossfire.** Sullivan. Aliens come at you from three directions on a grid laid out like city blocks. Each alien has four lives and metamorphoses into its next one when shot. Strategy and intense concentration required. Superb, smooth animation of a dozen pieces simultaneously. One of the great ones. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $29.95. *1/82.*

**David's Midnight Magic.** Snider. Pinball challenger to *Raster Blaster.* Excellent hi-res graphics and animation. Provision for earning extra balls. Broderbund, 1938 4th St., San Rafael, CA 94901. $34.95. *2/82.*

**The Eliminator.** Anderson. Pit your hi-res space fighter against numerous adversaries. Plenty of action. Adventure Intl., Box 3435, Longwood, FL 32750. $29.95. *7/82.*

● **Epoch.** Miller. Superbly stylized animation enhances this filmic shoot-'em-up. Tremendous sense of being in space; neat classical music and dramatic time warp sequence. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $34.95. *10/81.*

**Falcons.** Varsanyi, Ball. A hypnotically good shoot-'em-up with several levels of difficulty. Piccadilly, 89 Summit Ave., Summit, NJ 07901. $29.95. *10/81.*

**Fly Wars.** Trap fly fighters in your web, score with exploding cocoons. Beware the beetle and bug spray. Simple, addicting. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95. *7/82.*

**Gold Rush.** Berlyn, Wilker. Transport the gold from the train through the forest to waiting hoppers, avoiding bears, Indians, bandits, and random troublemakers. Sentient, Box 4929, Aspen, CO 81612. $34.95. *6/82.*

**Gorgon.** Nasir. Fly over planet shooting and dodging invaders and saving kidnapped inhabitants. Outstanding hi-res graphics, challenging refueling sequence—if you can get that far. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $39.95. *8/81.*

**Guardian.** Tom & Jerry. Blast your way out of six maze levels surrounded by hostile aliens. Fast and tricky. Continental, 11223 S. Hindry Ave., Los Angeles, CA 90045. $29.95. *7/82.*

**High Orbit.** Merret. Use lasers to lift modules into place to construct floating space station. Watch out—it's a crowded cosmos. Gebelli, 1787 Tribute Rd., Ste. G, Sacramento, CA 95815. $29.95.

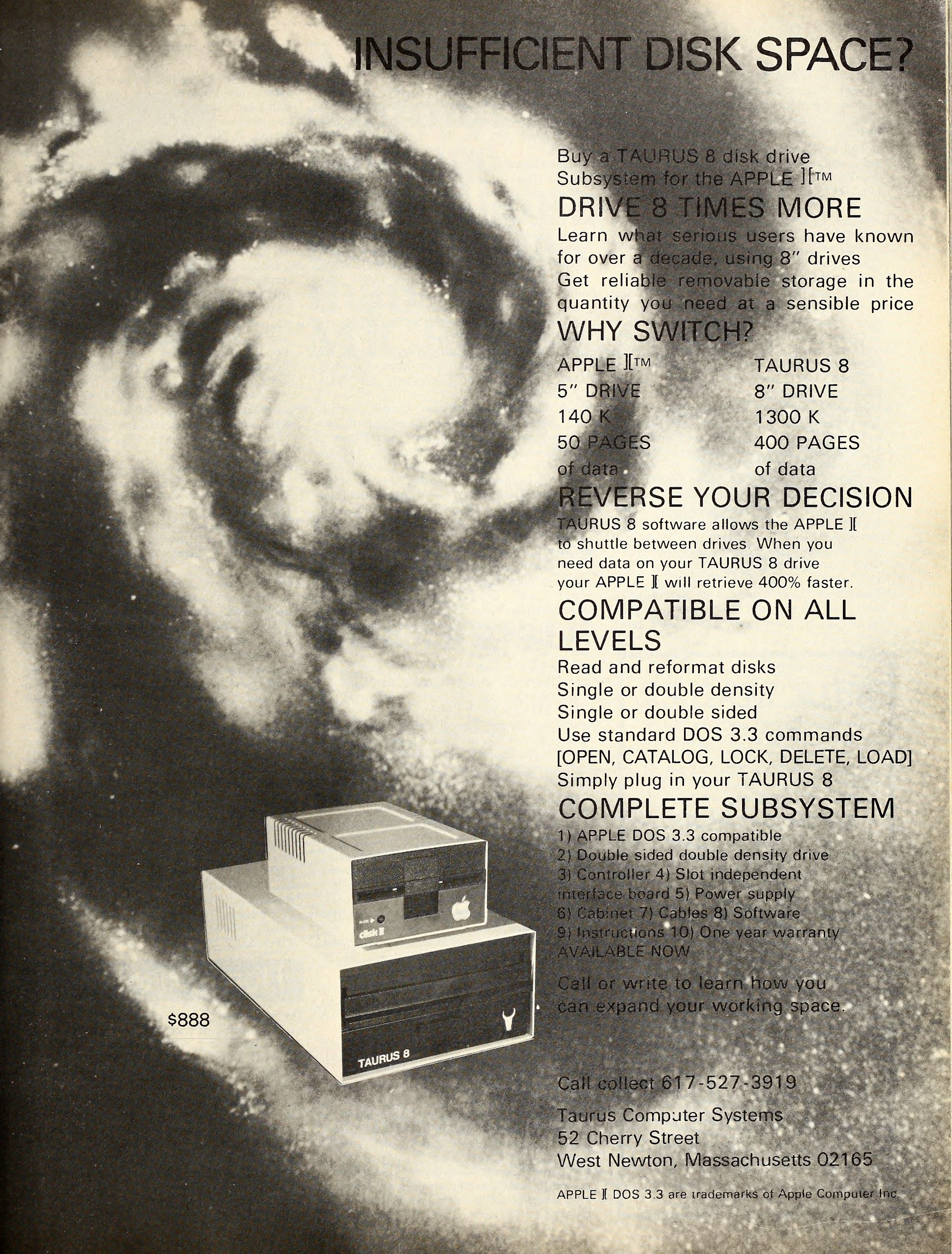
**Horizon V.** Nasir. Okay follow-up to *Gorgon* with superb animation, though not much challenge. Gebelli, 1787 Tribute Rd., Ste. G, Sacramento, CA 95815. $34.95.

**Human Fly.** Bagley. Good crude fun. Climb the C.P.U. building, avoiding apes, nasty birds, and slamming windows. Promises many excruciating falls. C.P.U., 9710 24th Ave. S.E., Everett, WA 98204. $29.95.

**Hungry Boy.** Nakan. Eat-the-dots, big ones and little ones. Four ghosts chase you through a maze—when their colors change, you can chase them. Astar Intl., 5675 Francis Ave., Chino, CA 91710. $24.95.

**Jawbreaker.** Lubeck. Candy store–oriented eat-the-dots game with automatically escalated skill levels. A courtroom favorite. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $29.95.

**Kamikaze.** A rain of planes falls mainly in this game — an aerial version of *Depth Charge.* Hayden, 600


INSUFFICIENT DISK SPACE?
Buy a TAURUS 8 disk drive
Subsystem for the APPLE ][™
DRIVE 8 TIMES MORE
Learn what serious users have known for over a decade, using 8" drives
Get reliable removable storage in the quantity you need at a sensible price
WHY SWITCH?
APPLE ][™ | TAURUS 8
5" DRIVE | 8" DRIVE
140 K | 1300 K
50 PAGES of data | 400 PAGES of data
REVERSE YOUR DECISION
TAURUS 8 software allows the APPLE ][ to shuttle between drives. When you need data on your TAURUS 8 drive your APPLE ][ will retrieve 400% faster.
COMPATIBLE ON ALL LEVELS
Read and reformat disks
Single or double density
Single or double sided
Use standard DOS 3.3 commands
[OPEN, CATALOG, LOCK, DELETE, LOAD]
Simply plug in your TAURUS 8
COMPLETE SUBSYSTEM
1) APPLE DOS 3.3 compatible
2) Double sided double density drive
3) Controller 4) Slot independent interface board 5) Power supply
6) Cabinet 7) Cables 8) Software
9) Instructions 10) One year warranty
AVAILABLE NOW
Call or write to learn how you can expand your working space.
$888
disk ][
TAURUS 8
Call collect 617-527-3919
Taurus Computer Systems
52 Cherry Street
West Newton, Massachusetts 02165
APPLE ][ DOS 3.3 are trademarks of Apple Computer Inc.

Suffolk St., Lowell, MA 01854. $34.95. *10/82.*

**Labyrinth.** Schram. Save your comrades amid *Crossfire*-style foes in a constantly shifting maze pattern. Challenging, excellent, lasting fun. Broderbund, 1938 4th St., San Rafael, CA 94901. $29.95. *6/82.*

**Laf Pak by Chuckles.** Beuche. Four-game variety disk; a real bargain. *Creepy Corridors* (the best), *Apple Zap*, *Space Race*, and *Mine Sweep.* Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $34.95. *10/82.*

**Lemmings.** Thompson. Round up mass-reproducing rodents, detaining nonbreeding pairs, before they migrate into the sea. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95. *6/82.*

**Marauder.** Weigandt, Hammond. Double duty: bust through force field as a rocket, then switch to man in a maze. Nine mazes with fifteen levels of difficulty. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $34.95.

**Mars Cars.** Fun little maze game — burn the levels as you play. Some easy; some hard. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $29.95. *10/82.*

• **Meteoroids (Asteroids) in Space.** Wallace. Making little asteroids out of big ones, plus occasional hostile alien ships. Hyperspace, autobrake, autofire. Quality Software, 6660 Reseda Blvd., Ste. 105, Reseda, CA 91335. $19.95.

**Microwave.** Zimmermann, Nitchals. Brightly colored, highly addictive maze game featuring continuous Looney Tunes musical accompaniment. Cavalier, Box 2032, Del Mar, CA 92014. $34.95. *5/82.*

**Minotaur.** Miller. Incorporates adventure elements and thirty-two four-level mazes. Surprises. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $34.95. *5/82.*

**Mouskattack.** Lay pipe through the maze, avoiding mice. Alas, cats and traps won't save you from Super Mouse. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $34.95.



**Nightmare Gallery.** Aldrich, Clardy. High-moon shoot-'em-up. Fast action with ghosts, mummies, and menacing rows of tombstones. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $34.95.

**Olympic Decathlon.** Smith. Ten standard decathlon events. Hi-res animated athletes, muscle-stirring music; you provide the sweat. Microsoft, 10700 Northup Wy., Bellevue, WA 98004. $29.95. *6/81.*

**Pest Patrol.** Allen. Where have all the flowers gone? Frenzied new bug game with hopping spiders, killer butterflies, and shielding snails—all with dive-bombing capabilities. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold. CA 93614. $34.95. *10/82.*

**Phaser Fire.** Salt City. Space shoot-'em-up. Defend vortex from swooping rockets and space junk. Gebelli, 1771 Tribute Rd., Ste. A, Sacramento, CA 95815. $29.95.

**Pig Pen.** TMQ. Latest wrinkle in drop-the-dots, featuring hi-res swine and instant hams. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $29.95. *9/82.*

**Pinball A2-PB1: Night Mission.** Artwick. Fantastically realistic and competitive ten-mode pinball simulation, allowing user modification and definition of virtually every aspect of play. SubLogic, 713 Edgebrook Dr., Champaign, IL 61820. $29.95. *5/82.*

• **Pool 1.5.** Hoffman, St. Germain, Morock. Makes most shots you could on a real table, with the advantages of instant replay and slow motion. Four different games. IDSI, Box 1658, Las Cruces, NM 88004. $34.95. *6/81.*

**Quadrant 6112.** Hold your space alone against a fleet of rebel invaders popping through two blue squares. Sensible, 6619 Perham Dr., W. Bloomfield, MI 48033. $34.95.

• **Raster Blaster.** Budge. Pinball game as good as real ones. *Softalk* readers' Most Popular Program of 1981. BudgeCo, 428 Pala Ave., Piedmont, CA 94611. $29.95. *5/81.*

**Rear Guard.** Five-level rocket run over scrolling terrain. The twist: you dog the aliens, they don't dog you. Adventure Intl., Box 3435, Longwood, FL 32750. $29.95. *8/82.*

**Russki Duck.** Knopp, Merrell. Recover stolen missile plans hidden in fake duck while dispatching enemy agents. Fairly easy. Gebelli, 1778 Tribute Rd., Ste. G, Sacramento, CA 95815. $34.95.

**Sea Fox.** Good sub versus convoy home-arcader. Variety of vessels, bouncing torpedoes, refueling dolphins (food not included), and intelligent depth charges. Broderbund, 1938 4th St., San Rafael, CA 94901. $29.95.

**Serpentine.** Hypnotic snake-chase maze game. Clean action, thrills, hairy escapes. Recommended. Broderbund, 1938 4th St., San Rafael, CA 94901. $34.95. *10/82.*

**Sheila.** Fitzgerald. Highly adventure-flavored, five-level, real-time maze game with weapons, commands, and spells—acquired with increasing point totals. H.A.L. Labs, 4074 Midland Rd., Ste. 23, Riverside, CA 92505. $23. *7/82.*

**Snack Attack.** Illowsky. A three-maze eat-'em-up; starts at any of five speed levels. Nonfattening. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $29.95. *1/82.*

**Snake Byte.** Arcade action featuring fruit and serpents. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95.

**The Snapper.** Different. Eat the *blots* while the whirlers slowly consume the maze. Takes strategy and quick thinking on slippery speedways, avoiding the ever-tossing gamma sticks. Silicon Valley Systems, 1625 El Camino Real, Ste. 4, Belmont, CA 94002. $32.95.

• **Sneakers.** Turmell. Many-layered shoot-'em-up, one of the best. Stomping sneakers and swarm of other creatures add to the fun. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95. *9/81.*

**Spy's Demise.** Be the first on your block to run a maze of pile-driving elevators. Fast, frustrating fun with a game show bonus: complete puzzle after all nine levels and win a prize. Penguin, 830 4th Ave., Geneva, IL 60134. $29.95.

**Star Blaster.** Mines, fireballs, space tunnels, general obstructions and unfriendlies waylay your starship. Piccadilly, 89 Summit Ave., Summit, NJ 07901. $29.95. *8/82.*

**Star Blazer.** Suzuki. Bomb-run game with five levels, minutely exact animation, and style to burn. A joy. Broderbund, 1938 4th St., San Rafael, CA 94901. $31.95. *4/82.*

**Succession.** Eisnaugle. Real-time maze game. Get the creatures in numbered order as the chaser nips at your heels. Piccadilly, 89 Summit Ave., Summit, NJ 07901. $29.95. *10/82.*

• **Super Invader.** Hata. The daddy of home-arcades. Still good hi-res, still a challenge. *Softalk* readers' Popular Program of 1978–80. Astar Intl., through California Pacific, 1615 5th St., Davis, CA 95616, and Creative Computing, 39 E. Hanover Ave., Morris Plains, NJ 07950. $19.95.

**Swashbuckler.** Stephenson. Hi-res swordfighting with realistic pirates, snakes, rats, and other scum. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $34.95. *8/82.*

**Teleport.** Abbot. Need a job? Learn to stun and bag aliens in your spare time. Maze game with lots of action. Cavalier, Box 2032, Del Mar, CA 92014. $29.95. *10/82.*

**Tharolian Tunnels.** Nelsen. Shoot-'em-up with several stages of play; on par with *Falcons.* Software Farm, 3901 S. Elkhart St., Aurora, CO 80014. $29.95. *10/82.*

**Threshold.** Schwader, Williams. Another shoot-'em-up. Hi-res graphics, animation, and accurate collisions. Targets include everything from flying maple trees to Volkswagen Bugs. Frustratingly small fuel supply. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $39.95. *12/81.*

**Track Attack.** Jochumson. Three-level train robbery chase game requiring considerable dexterity. Broderbund, 1938 4th St., San Rafael, CA 94901. $29.95. *4/82.*

**Tunnel Terror.** Popejoy. Noisy arcade fun. Shoot down tunnel at escaping aliens, get them before they get you. Adventure Intl., Box 3435, Longwood, FL 32750. $29.95. *10/82.*

**Twerps.** Thompson. Plot, elaborate animation, and sound link aspects of several different game styles together. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95.

**Wayout.** Exciting 3-D maze that moves in perspective as you play. Map displayed at all times. Lots of angles and Cleptangles. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $39.95. *10/82.*

**Zenith.** Nasir. Similar to *Horizon V*; 3-D scrolling over planetoid. Build city while fighting off aliens. Gebelli, 1771 Tribute Rd., Ste. A, Sacramento, CA 95815. $34.95. *8/82.*

## Home/Hobby

**The Accountant.** Forman. Double-entry finance system features seven integrated files and a set of automatic transactions. Decision Support, 1438 Ironwood Dr., McLean, VA 22101. $129.95. *1/82.*

**A.L.D.S.** Assembly language development system. Writes Z-80, 80-80, and 6502 op-codes; defines macros; transfers 6502 to DOS. Microsoft, 10700 Northup Wy., Bellevue, WA 98004. $125.

**Apple Aide.** Programmer's utility for Basic or machine language that has disk editor, disk mapping, and how-tos on writing and editing. Advanced Operating Systems, 450 St. John Rd., Michigan City, IN 46360. $49.95.

**Apple-Cillin.** Hardware diagnostic tests for all RAM and ROM, plug-in cards, cp registers, disks; nine

video test patterns. XPS, 323 York Rd., Carlisle, PA 17013. $49.95.

**Apple Logo.** Papert. Custom version (by its inventor) of MIT-developed turtle graphics language. First-rate educational tool with graphics, mathematical, even games use. Hefty documentation. Apple, 10260 Bandley Dr., Cupertino, CA 95014. $175.

**Apple Mechanic.** Kersey. Multiple utility disk with shape editor, custom typefonts, byte rewriter, and tricks to facilitate music, text, and hi-res generation. Beagle Bros, 4315 Sierra Vista, San Diego, CA 92103. $29.50.

**Apple Spice.** Kosak, Fox. Powerful Applesoft expansion utility using *&* and *usr* functions. Easily incorporated programming routines. Adventure Intl., Box 3435, Longwood, FL 32750. $29.95. *5/82.*

**Audex.** Collection of utilities to create, edit, and play back your own sounds for your own programs; in Basic and assembly language. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95.

**Bag of Tricks.** Worth, Lechner. Four utility programs for dumping and examining a raw track, sector editing, reformatting tracks, and repairing damaged disk catalogs. Quality Software, 6660 Reseda Blvd., Ste. 105, Reseda, CA 91335. $39.95.

**Busywork.** Basic programs and routines for developing new business programs. Use as a start-up, add your own program codes as you go. Datum Consultants, 1641 State St., Box 238, DeKalb, IL 60115. $39.95.

**Chequemate.** Home finance package that handles checks, charge cards, cash control, automatic tellers, and more. Reports to screen or printer. A bargain. Masterworks, 25834 Narbonne Ave., Lomita, CA 90717. $39.95.

**C.O.R.P.** Program generator. Answer questions in English to design Basic programs that run without *C.O.R.P.* Dynatech, 7847 Caldwell Ave., Niles, IL 60648. $250.

**Crossword Magic.** Crossword puzzle maker. Choose subject, words, and clues; program automatically connects words. Play on screen or make professional-quality print-out. L & S Computerware, 1589 Fraser Dr., Sunnyvale, CA 94087. $49.95.

**Datafax.** Database utilizing unstructured keyword classification system for categorizing and cross-referencing by any method. No programming required; hard disk compatible. Link Systems, 1640 19th St., Santa Monica, CA 90404. $199.

**Disk Library.** Reads your catalogs and organizes your programs in a printed, sorted list. CP/M and Pascal files entered through keyboard. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $49.95.

**Diskovery.** Disk utility. Recover lost files, test drive speed, fix I/O errors, and more. Micro Mantic, 541 N.E. McWilliams Rd., Bremerton, WA 98310. $59.95.

**Disk Recovery.** Utility to recover disk files. Deletes files and rewrites sectors if you can't patch by hand. Sensible, 6619 Perham Dr., W. Bloomfield, MI 48033. $30.

**Disk Scanner.** Looks for and fixes bad tracks, checks for bad sectors, and rebuilds your catalog. Sensible, 6619 Perham Dr., W. Bloomfield, MI 48033. $30.

**DOS Boss.** Kersey, Cassidy. Utility to change, shorten DOS commands; customize catalog. Good ideas and witty presentation. Beagle Bros, 4315 Sierra Vista, San Diego, CA 92103. $24. *10/81.*

**DOS Tool Kit.** Excellent utility package; Apple II assembler-editor system and Applesoft tool kit. Edit, assemble machine language programs; write, edit Basic programs. Simplifies graphics, includes character generator. Apple, 10260 Bandley Dr., Cupertino, CA 95014. $75. *10/81.*

**Double Check.** Hill. Checkbook balancer that handles dozens of accounts. Lists, sorts, prints; has 100 categories; shows seventeen checks per screen. Computer Tax Service, Box 7915, Incline Village, NV 89450. $39.95.

**Electric Duet.** Lutus. Two-voice music without hardware. A bit involved, but superb sound quality. Insoft, 10175 S.W. Barbur Blvd., Ste. 202-B, Portland, OR 97219. $29.95. *7/12.*

**Expediter II.** Einstein, Goodrow. Applesoft compiler translates Basic programs into machine language. Will display or print a running list of source program lines and compiled addresses; compiled program size reduced up to 50 percent. No stop on fatal errors. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $99.95. *9/81.*

**Fast DOS.** Triples access speed; compatible with all DOS-Applesoft programs that access DOS through standard hooks. Wytand P/L, 60 Gollan Ave., Dundas, 2117, Australia. $29.

**File Whiz.** Goss. Quickly learned database management program with six command modes. Files generated are accessible from Basic programs. Fast, easy, and convenient for home use and users. Soft-House, Box 6383, Rochester, MN 55903. $79. *12/81.*

**Financial Management System II.** Home finance management; maintains multiple accounts, generates complete audit reports, and stores unlimited files. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071.

**GPLE.** Enhanced version of the *Program Line Editor.* Edit everything on a line, line by line, or on a range of lines; plus search for strings. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $60.

**Home Accountant.** Schoenburg. Thorough and powerful home finance program. Monitors five checking accounts against a common budget, plus credit cards and cash; one-step record of transfer of funds. Continental, 11223 S. Hindry Ave., Los Angeles, CA 90045. $74.95. *4/82.*

**The Inspector.** Sefton. Fast, flexible utility for examination of disk sectors, directory, and track-sector lists. Salvage blown disks, change data, delete DOS. Omega, 222 S. Riverside Plaza, Chicago, IL 60606. $49.95. *11/81.*

**LISA 2.5.** Hyde. Long-time popular assembler with extended mnemonics and more than thirty op-codes. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $79.95.

**Master Diagnostics.** Twenty-four tests that check your Apple for component malfunctions and suggest replacement. Software Source, 17905 Ventura Blvd., Encino, CA 91316. $69.95.

**MasterType.** Zweig. Learn to type by playing a game; simple and ingenious. Lightning, Box 11725, Palo Alto, CA 94306. $39.95. *4/81.*

**Memory Management System.** Relocates DOS to RAM board in slot 0, frees memory location space for programming, Computer Data Services, Box 696, Amherst, NJ 03031. $49.95.

**Merlin.** Does assembly language programming with a dozen editing commands and twenty-eight pseudo-ops. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $64.95.

**Multi-Disk Catalog III.** Very fast machine language database program for reading and storing file names, types, and sizes. Fast, powerful sort-and-search feature. Sensible, 6619 Perham Dr., W. Bloomfield, MI 48033. $25. *10/81.*

**Personal Finance Manager.** Gold, Software Dimensions. Handles up to 200 entries a month from maximum of 14 separate accounts. Search-sort-edit routine. Apple/Special Delivery, 10260 Bandley Dr., Cupertino, CA 95014. $75. *11/81.*

**Personal Finance Master.** Personal and small business financial system; covers all types of accounts. Spectrum, 142 Carlow, Box 2084, Sunnyvale, CA 94087. $74.95.

• **Program Line Editor.** Program development and modification program with more than eleven editing commands, listing control, lower case, and programmable cursor control. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $40.

**Program Writer/Reporter.** Database code generator that does standalone program writing. Interactive between files and fields within programs. Vital Information, 7899 Mastin Dr., Overland Park, KS 66204. $200.

**Real Estate Analyzer.** Make buy and sell decisions, compare investments, project future sales year-to-year for ten years. File, retrieve, and alter information itemized in tabular form. Howard Software, 8008 Girard Ave., Ste. 310, La Jolla, CA 92037. $195. *7/81.*

**The Routine Machine.** Meyer. Programming tool that extends Applesoft. Library of routines; install your own; no programming knowledge necessary. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $64.95.

**S-C Macro Assembler.** Development tool for programming in assembly—edit, copy, and replace. Full macros and conditional assembly. Source files can be as large as your disk space; language-card version included. S-C Software, Box 280300, Dallas, TX 75228. $80.

**Soft-Step.** Applesoft Basic interactive debugger. Steps through programs, breaks at any point; trace and list functions are improvements over originals. Accent, 3750 Wright Pl., Palo Alto, CA 94306. $49.95. *8/82.*

**Statistics with Daisy.** Statistics analyzer for business, science, and social use. Hypothesis testing, correlations, multiple regression, and variance analysis. Rainbow, 19517 Business Center Dr., Northridge, CA 91324. $79.95. *10/82.*

• **Super Disk Copy III.** Hartley. Easy-to-use menu-driven software library utility; transfers all types of DOS files. Sensible, 6619 Perham Dr., W. Bloomfield, MI 48033. $30. *10/81.*

**SuperPilot.** Courseware authoring language based on Apple and Common Pilot. Twenty-six instructions for creating individually tailored lessons. Apple, 20525 Mariani Ave., Cuptertino, CA 95014. $200.

**TASC.** Peak, Howard. Applesoft compiler. User con-

trols locations of three memory compartments. Microsoft, 10700 Northup Wy., Bellevue, WA 98004. $150. *9/81.*

**The Tool.** Code generator. Programs generated will address up to 80 megabytes (four hard disks). Has customizing features; good productivity tool for programmers. High Technology, Box 14665, Oklahoma City, OK 73113. $395.

**Typing Tutor.** Ainsworth, Baker. Four levels of proficiency; individualized drills created with time response monitoring. Microsoft, 10700 Northup Wy., Bellevue, WA 98004. $24.95.

**Utility City.** Kersey. Twenty-one utilities on one disk. Beagle Bros, 4315 Sierra Vista, San Diego, CA 92103. $29.50.

**VisiDex.** Jennings. Electronic index and file-agenda program for spontaneous or structured information entry. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $199.95.

**Watson.** Dutiful disk utility, requires *The Inspector.* Recovers blown disks, repairs bad data files, searches and scans. Omega, 222 S. Riverside Plaza, Chicago, IL 60606. $49.95.

## Strategy

**Acey Deucey.** Okay version of the betting card game. Computer seems to have all the odds. L & S Computerware, 1589 Fraser Dr., Sunnyvale, CA 94087. $29.95. *10/82.*

**Air Navigation Trainer.** Winograd. Flight simulator with four games and VOR training aid. Can be played for fun or used to learn basic navigation. Space-Time Associates, 20-39 Country Club Dr., Manchester, NH 03102. $40.

**AirSim-1.** Machine language flight simulator in 3-D with six landing fields and optional instrument flying mode. Mind Systems, Box 506, Northampton, MA 01061. $40.

**Bull Run.** Finelli. Stock market simulation game. Trade twelve stocks, purchase options, earn T-bill interest. First Flight, Box 555, Kitty Hawk, NC 27949. $29.95.

**Casino.** Five hi-res games, Vegas-style: blackjack, baccarat, keno, poker, and roulette. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $39.95. *10/82.*

**Castle Wolfenstein.** Warner. First game to fuse successfully best elements of home-arcade and adventure. Escape from Nazi stronghold, finding and taking secret plans. Room layout changes with each new game. Enemy speaks, in German. Muse, 347 N. Charles St., Baltimore, MD 21201. $29.95. *10/81.*

● **Computer Baseball.** Merro, Avery. Remarkable programming feat, simulating individual player abilities from the teams of thirteen famous World Series. Can enter and play teams of your own creation. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $39.95. *9/81.*

**Cosmic Balance.** Design your own ships and create your own space fleet. Tactical space game that's fast and easy to play. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $39.95.

**Cytron Masters.** Bunten. Fast-paced strategy in real time; plays much like football. Rates high in excitement and intelligence. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $39.95. *10/82.*

● **Flight Simulator.** Artwick. Utilizes aerodynamic equations and airfoil characteristics for realistic simulation of take-off, flight, and landing. SubLogic, 713 Edgebrook Dr., Champaign, IL 61820. $33.50.

**Galactic Gladiators.** Reamy. Easy and enjoyable tactical simulation. Suit up and arm your fighters for ten different scenarios. High speed for a strategy game. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $39.95.

**Gin Rummy.** Carpet. Play against computer. Hi-res cards can change position in hand; your entire hand visible. Space bar allows you to change your mind when discarding. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $29.95. *6/82.*

**Guadalcanal Campaign.** Grigsby. Massive land, sea, and air re-creation of the dramatic battle. Fast and exciting; good balance of conflict. Includes 300-turn campaign game. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $59.95. *10/82.*

**Hi-Res Computer Golf.** Aronoff. A masterpiece of skill testing, judgment, strategy, and visual acuity. One of the few computer sports simulations that itself requires athletic dexterity. Avant-Garde, Box 30160, Eugene, OR 97403. $29.95. *2/82.*

● **Microgammon II.** Competition program for learning, practice, and improvement of backgammon skills. Tournament play. Softape, 5547 Satsuma Ave., North Hollywood, CA 91601. $19.95. *2/81.*

**Millionaire.** Executive stock market simulation game played via reports, graphs, options, and volume indicators. Blue Chip, 19537 Wells Dr., Tarzana, CA 91356. $79.95.

**Pursuit of the Graf Spee.** The 1939 engagements of the German pocket battleship off South America. Visibility and sighting system; separate ranges for each gun turret. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $59.95.

**Rendezvous.** Huntress. Space shuttle simulation in 3-D, created by senior scientist at JPL. Orbit earth, match orbit, and dock with space station. Authentic, demanding. Edu-Ware, Box 22222, Agoura, CA 91301. $39.95. *7/82.*

**RobotWar.** Warner. Strategy game with battling robots is teaching device for programming. Muse, 347 N. Charles St., Baltimore, MD 21201. $39.95. *1/81.*

● **Sargon II.** Spracklen, Spracklen. Computer chess game with seven levels of play. Hayden, 50 Essex St., Rochelle Park, NJ 07662. $34.95.

**The Shattered Alliance.** Suit up and arm unicorns, dwarfs, zorks, and centaurs to do battle in seven scenarios, both fantasy and historical. Also called *Chronicles of Osgoth.* Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $59.95. *7/81.*

**Spitfire Simulator.** Air flight simulator — Spitfire in combat with German Aces — with 3-D scenery and moving target aircraft. Mind Systems, Box 506, Northampton, MA 01061. $40.

**Warp Factor.** Space war game featuring twelve starship designs representing five galactic empires, with possible scenarios ranging from skirmishes to galactic war. Extremely challenging. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $39.95. *7/81.*

## Word Processing

**Apple Writer.** The most popular word processing program in town. Type, erase, move words around, save and insert segments from disk, and print out. Easy to use. Apple, 10260 Bandley Dr., Cupertino, CA 95014. $75.

**Apple Writer II.** Lutus, Finstead. Written in word-processing language. Additional editing features and functions menu; continuing features and functions menu; continuous readout of character count and length. Apple/Special Delivery, 10260 Bandley Dr., Cupertino, CA 95014. $150.

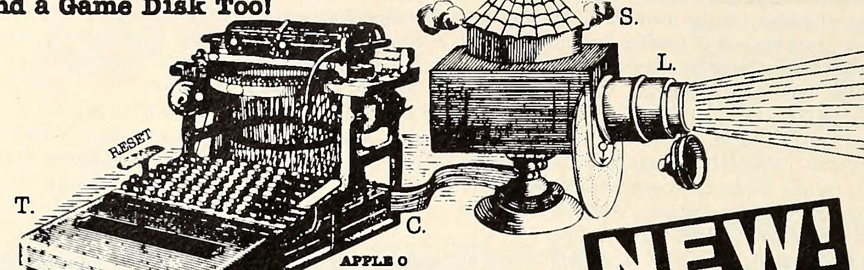
**Apple Writer Extended Features.** Malachowski, Cooper. Enables production of multiple copies of *Apple Writer* files and insertion of variables; converts Applesoft programs to *Apple Writer* and vice versa. Brillig Systems, 10270 Fern Pool Ct., Burke, VA 22015. $34.95. *7/81.*

**EasyWriter.** Word processor; choose 40 or 80 column version. Information Unlimited, 281 Arlington Ave., Berkeley, CA 94707. $99.95.

**Executive Secretary.** Editing, printing, and form let-

# NEW APPLE® UTILITIES FROM BEAGLE BROS

**And a Game Disk Too!**




## NEW!

## Alpha Plot

### Hi-Res Graphics/Text Utility by Bert Kersey & Jack Cassidy

Here are a few of Alpha Plot's useful graphics features. Compare with other graphic utilities at any price—

**HI-RES DRAWING:** Create hi-res pictures and charts with text, on both pages; all Optional Xdraw cursor (see lines before drawing). Mix colors & Reverse (background opposite). Circles, Boxes, Ellipses; filled or outlined. **Compress Hi-Res to 1/3 disk space.** Relocate any portion of an image anywhere on either page. Superimpose too & convert hi-res to lo-res for colorful abstracts!

**HI-RES TEXT:** Beautiful upper/lower case with descenders (no hardware required). Color and reverse characters positionable anywhere (no vtab/htab limitations). Professional-looking **proportional spacing** and adjustable character height and letter spacing. Sideways typing for graphs too!

**$39^{50}$**
- ☐ Unprotected disk (48K min.)
- ☐ Beagle Bros Apple Tip Book #4
- ☐ Peeks & Pokes Chart


## NEW!

## Frame-Up

### Graphics Display Utility by Tom Weishaar

Frame-Up is a very-high-speed Apple "slide projector" utility that lets you create professional-looking displays of intermixed hi-res, lo-res and text pages **on any Apple.** Frame-Up is very easy-to-use and above-all **FAST,** allowing you to load hi-res pictures, for example, in **2½-seconds;** that's five-times faster than normal! Paddles or keyboard are used to change images in forward or reverse order, skipping pages if you want. OR presentations may be left unattended, with **each page individually timed** to appear and remain on the screen from 3 to 99 seconds, as you choose.

Frame-Up includes a sophisticated black and white **text screen editor** that lets you create text "slides" as part of your show. You can even add type "live" on the screen during your presentations. Up to 17 hi-res or 136 lo-res/text pages may be stored per disk. One or two drives are supported. The order and timing of your graphics and text images may be easily (and instantly!) arranged and rearranged. Frame-Up includes a **display module which may be copied** and distributed to your associates so they can run your display, as you designed it, on their Apple or ANY Apple!

Frame-Up is ideal for store displays, presentations to the boss, club programs, trade show booths, product demos, promotions, seminars, conventions, classes, and so on.

**$29^{50}$** Machine language. Unprotected. 48K minimum. Peek/Poke Chart included.

## Apple® Mechanic

### Shape Writer/Byte-Zap Utility by Bert Kersey

Another best-selling multiple-utility disk—Nine useful, listable, copyable and customizable programs—

**SHAPE EDITOR:** Put professional hi-res animation in your programs. Keyboard-draw any shape and let your Apple write a shape table and store it on disk. Design large and small **custom typefaces** too, with special characters. 6 fonts on the disk. LISTable demos show how to use shape tables to animate games, graphic displays, and attractive Charts & Graphs. A valuable time-saving utility/learning tool.

**BYTE ZAP:** A MUST utility. Rewrite any byte on a disk by loading a sector onto the screen for inspection. **Hex/Dec/Ascii** display optional. Examine bytes via cursor control; enter hex, dec or ascii to change. Create illegal filenames, restore deleted files, change greeting program names, repair/protect disks, change DOS, examine program files. Clear illustrated instructions show how disk data is stored and how to access it. Very educational.

**MORE:** A disk PACKED with useful music, text and hi-res tricks **for use in your programs.** A great demo-writer program, useful hi-res utilities and educational, entertaining documentation.

**$29^{50}$**
- ☐ Unprotected disk (48K min.)
- ☐ Beagle Bros Tip Book #5
- ☐ Peeks & Pokes Chart


## NEW!

## Beagle Bag!

### 12 Games on One Big Disk by Bert Kersey

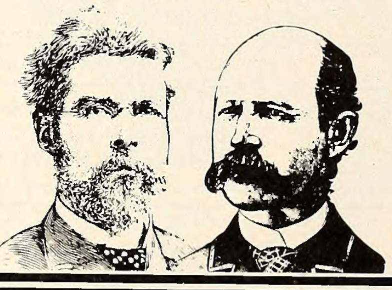
Twelve great games from our classic Beagle Bros collection—TextTrain, Slippery Digits, Wowzo, Magic Pack, Buzzword... Almost all of our "Game Pack" games have been updated and re-released on one jam-packed unprotected disk! **ALSO INCLUDED** is our "Beagle Menu" greeting program (description under "Typefaces" disk on this page).

**Compare Beagle Bag with any 1-game locked-up disk on the market today!**

All 12 games are a blast, the price is right, the instructions are crystal clear, AND the disk is copyable! You can even list the programs to see what makes them tick!

**$29^{50}$** Unprotected. Paddles NOT required. Beagle Menu works with all normal-DOS disks. Includes Peeks/Pokes Chart.

BEAGLE BROS DISKS ARE UNLOCKED AND UNPROTECTED. THIS MEANS EVERY PROGRAM IS INSPECTABLE, CUSTOMIZABLE, IF YOU WANT, AND COPYABLE, GIVING YOU THE MOST FOR YOUR SOFTWARE DOLLARS. DON'T SETTLE FOR LESS.



**Beagle Bros™ MICRO SOFTWARE**

**4315 Sierra Vista, San Diego, Ca 92103**
**714-296-6400**

"APPLE" is a registered trade mark of You-Know-Who.

## NEW!

## Typefaces

### for Apple Mechanic

Here are more hi-res fonts for Apple Mechanic's Xtyper and Hi-Writer programs—26 of them at last count, both large and small, all **proportionally-spaced** and positionable anywhere on either hi-res screen. Most are **full 96-character fonts** many with special graphic characters. Each character (from "!" to "□") of every font (from "Ace" to "Zooloo") is, of course, editable with Apple Mechanic's Font Editor.

**BONUS:** Here's BEAGLE-MENU! A unique greeting program that displays **only the catalog file names you want** on the screen (for example, only locked-Applesoft files, or only Binary files) for one-key cursor selection. Just hit Return to Run, Brun or Exec the program at the cursor. Many other features— Space-on-Disk, Load/Bload option, forward and backward catalog "scrolling" for easy file location, and optional sector-number elimination. PLUS the ability to **swap file names** in your catalog!

**$20^{00}$** Unprotected. Beagle Bros' Apple Mechanic disk is required to utilize the type fonts. Beagle-Menu works with all normal-DOS 3.3 disks.

If you don't find our products at your Apple Dealer, tell him to phone Beagle Bros, 714-296-6400, **OR** his favorite software distributor.

ters, plus mail merge and electronic mail system. SofSys, 4306 Upton Ave. S., Minneapolis, MN 55410. $250.

**Format II.** Word processor with logic-sorting mailing list. Justifies type, wraps text; has one-key editing, menu prompting. Kensington Microware, 300 E. 54th St., Ste. 3L, New York, NY 10022. $375.

**Goodspell.** Dictionary companion disk to *Apple Writer* with 14,000 words. Flags words not listed when printing out. Apple/Special Delivery, 10260 Bandley Dr., Cupertino, CA 95014. $60.

**Graphtrix.** Matrix graphics system designed to add graphics, footnotes, and chapter capabilities to *Apple Writer* text editing system. Data Transforms, 906 E. 5th Ave., Denver, CO 80218. $65.

**Gutenberg.** User-definable character set, split-screen hi-res and lo-res text editing for text, program files. Formats any kind of page automatically (2, 3, or 4 column). Performs text block moves and deletes; paint program produces large illustrations integrated with text. Micromation, 1 Yorkdale Rd., Ste. 406, Toronto, Ont., Canada M6A3A1. $315.

**Letter Perfect.** Format-flexible word processor with ability to send control codes within body of program. Works with database files from *Data Perfect.* LJK, Box 10827, St. Louis, MO 63129. $149.95.

**Magic Window.** Word processing program simulates standard typewriter. 80-column text scrolls across 40-column screen. Three modes of disk file storage. Artsci, 5547 Satsuma Ave., North Hollywood, CA 91601. $99.95.

**Magic Window II.** Get 40, 70 (in hi-res), or 80 columns in this expanded version. Compatible with Pascal 80-column. With user-tailored, fast menu; underlining; global search and replace. Artsci, 5547 Satsuma Ave., North Hollywood, CA 91601. $149.95.

**Magic Words.** Proofreads files of word processors that use standard DOS and no character-encryption techniques for saving files. 14,000-word dictionary. Artsci, 5547 Satsuma Ave., North Hollywood, CA 91601. $69.95.



**MailMerge.** Overlay companion to *WordStar.* Add files at print time, sort items, specify variables. Command-driven. MicroPro, 33 San Pablo Ave., San Rafael, CA 94903. $150.

**Perfect Speller.** In-context spelling checker that integrates with *Perfect Writer.* Processes 4,000 wpm; has 50,000-word dictionary. Perfect Software, 1400 Shattuck Ave., Berkeley, CA 94709. $189.

**Perfect Writer.** Powerful, easy-to-use word processor. Advanced document design features undents, subheads, footnotes, quotations. Requires Z-80 card and 80-column board. Perfect Software, 1400 Shattuck Ave., Berkeley, CA 94709. $389.

**Personal Secretary.** At-home version of *Executive Secretary* minus electronic mail and alphabetical indexing. SofSys, 4306 Upton Ave. S., Minneapolis, MN 55410. $75.

**PIE Writer.** Business processor that allows 9,999 pages. With word deletion, auto indent, spooling, and typeahead buffer. Hayden, 50 Essex St., Rochelle Park, NJ 07662. $149.95.

**PowerText.** Does memos, letters, reports, and manuscripts without formatting each time. Good balance of automatic and user-defined functions. Beaman Porter, Pleasant Ridge Rd., Harrison, NY 10528. $199.

**ScreenWriter II.** Kidwell, Schmoyer. Formerly *SuperScribe II.* No extra hardware for lower case, 70-column display, printer spooling. Edits Basic, text, and binary files; complete search and replace. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $129.95.

**Sensible Speller.** Spell-checking program sports listable 85,000 words, extensible up to 110,000 words. Recognizes contractions, gives file word counts, incidence of a single word, and number of unique words. High marks for clear, logically organized documentation and simplicity of operation. Sensible, 6619 Perham Dr., W. Bloomfield, MI 48033. $125. *1/82.*

**Super-Text 40/56/70.** Zaron. Get 40, 56, or 70 columns without hardware. Design your own character sets. Basics of text editing. Character-oriented, floating-cursor edit with add, change, print, and preview modes. Muse, 347 N. Charles St., Baltimore, MD 21201. $125.

**Super-Text 40/80.** Zaron. Latest *Super-Text* update; letter documentation, footers and headers, expandable math mode, split screen. Muse, 347 N. Charles St., Baltimore, MD 21201. $175.

**Word Handler.** Elekman. Wonderfully simple program with straightforward documentation. Allows folded paper printout for two-sided printing. Silicon Valley Systems, 1625 El Camino Real, Ste. 4, Belmont, CA 94002. $199. *10/81.*

**WordStar.** Screen-oriented, integrated word processing system in CP/M. Requires Z-80 card. MicroPro, 33 San Pablo Ave., San Rafael, CA 94903. $495.

**Zardax.** Philips. Highly recommended. Single program includes all standard word processing features with considerable extras including communication by modem. Computer Solutions, Box 397, Mount Gravatt, Queensland, Australia. In the U.S.: Action-Research Northwest, 11442 Marine View Dr. S.W., Seattle, WA 98146. $295. *5/82.*

## Apple III

**Access III.** Communications program for time sharing and standalone tasks; accesses remote information services, minis, and mainframes. Apple, 10260 Bandley Dr., Cupertino, CA 95014. $150.

**Apple Business Basic.** High-level structured programming language for the III. Apple, 10260 Bandley Dr., Cupertino, CA 95014. $125.

**Apple III Business Graphics.** Converts numerical information into charts and graphs; only graphics program to take advantage of the III's capabilities. Apple, 10260 Bandley Dr., Cupertino, CA 95014. $175.

**Apple Writer III.** Lutus. Uses WPL (word processing language) to automate the process of text manipulation and document creation. Adjusts print format during printing, translates from typewriter shorthand to English or other language and back again. Apple, 10260 Bandley Dr., Cupertino, CA 95014. $225.

**Data Reporter.** Flexible database management system. Does form letters, patient files, labels, calculations, inventories, and employment records. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $220.

**EASy.** Executive accounting system with accounts receivable, accounts payable, and general ledger. Denver Software, 14100 E. Jewell Ave., Ste. 15, Aurora, CO 80012. $749.95.

**Hardisk Accounting System.** General ledger, accounts receivable, and accounts payable each handle up to 9,999 customers or accounts; inventory features five methods of evaluation. Also payroll, fixed-asset management, and mailing labels. Great Plains Software, 123 N. 15th St., Fargo, ND 58102. $395 to $595 per module.

**Mail List Manager.** Generates, stores, sorts, edits, and prints database files. Apple, 10260 Bandley Dr., Cupertino, CA 95014. $150.

**Micro/Terminal.** Access any in-house or remote database; set up and log only once. Built-in editor or edit off-line. Microcom, 1400A Providence Hwy., Norwood, MA 02062. $99.95.

**Pascal.** Program preparer with editor, compiler, disassembler, linker, filer, and system library. Features cursor control, text modeling, and formatting. Apple, 10260 Bandley Dr., Cupertino, CA 95014. $250.

**Personal Filing System.** Page. Form-oriented information management system allows storage and retrieval of up to 32,000 entries. Software Publishing, 1901 Landings Dr., Mountain View, CA 94043. $145.

**PFS: Report.** Page. Generates reports; performs functions that require sorting, calculating, and manipulating data filed with *PFS.* Software Publishing, 1901 Landings Dr., Mountain View, CA 94043. $175.

**Program Writer/Reporter.** Basic database code generator that creates interactive, standalone programs. Vital Information, 7899 Mastin Dr., Overland Park, KS 66204. $200.

**Quick File III.** Personal index card or filing system. Fifteen fields; file as long as disk allows; can be put on ProFile. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $100.

**VersaForm.** Landau. State-of-the-art business forms processor. Does invoicing, purchasing orders, mailing lists, client billing. Powerful, complex, worth getting to know. Hard disk compatible. Applied Software Technology, 15985 Greenwood Rd., Monte Sereno, CA 95030. $495.

**VisiCalc III.** Software Arts, Bricklin, Frankston. Just like it sounds: expanded memory, lower case, 80 columns. Four-way cursor movement. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $250.

**VisiSchedule.** Critical path PERT schedule planner. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $300.

**Word Juggler.** Gill. Word processor makes use of upper and lower case keyboard, 80-column display, and expanded memory. Print-out can be reviewed on screen prior to printing; multiple copies printed of selected pages. Quark Engineering, 1433 Williams, Ste. 1102, Denver, CO 80218. $295.

**Word Weaver III.** Fast word processor with unique feature: it can print out in shapes. Allows line spillover, does insert and delete, prints multiple files. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $99.95.

# RISING STARS


**What some have called "the best kept secret in the game software industry". Unleashed here are six new software games blending advanced graphics with precise elements of playability, challenge and intense fun for every gamer's tastes. Beyond these six new games are 26 more for you to consider . . . all we believe are the best software values for your money!**

**LEGIONNAIRE**—Real-time simulation of tactical combat in Caesar's time. YOU as Caesar command up to ten legions. Finest **full-color graphics** plus playability way beyond the state of the art! Cassette . . . $35.00

**V.C.**—Faithfully recreates unconventional conflict in Viet Nam. YOU command chopper and artillery units, and face task of protecting civilian population where the enemy hides among the people. Cassette . . . $20.00 Diskette . . . $25.00

**G.F.S. SORCERESS**—Sci-Fi adventure game. YOU are Joe Justin trying to clear yourself of a false charge of mutiny and get back to the Galactic Federation Starship "Sorceress". Beautiful full-color manuals provide useful clues. Cassette . . . $30.00 Diskette . . . $35.00

**ANDROMEDA CONQUEST**—Vast scale space strategy game of galactic colonizing and conquest among unique star systems with strange life forms and alien technologies that provide exciting exploration and battle. Cassette . . . $18.00 Diskette . . . $23.00

**MOON PATROL**—Arcade Pak™ game of lunar invasion. Beats any quarter-gobbling game around! Four levels of increasing difficulty present new attackers to battle. Fast, furious and fun! Cassette . . . $25.00

**TELENGARD**—Dungeon adventure in a mysterious underworld with 50 levels of ever-more-complex mazes to explore. Real time fantasy and role-playing game. Using wits, magic and fast thinking, gamers fight monsters and reap valuable rewards. Cassette . . . $23.00 Diskette . . . $28.00

® Trademarks for Apple Computer, Warner Communications, Tandy Corp., International Business Machines and Commodore International Ltd.

Available at finer computer stores everywhere!

AH QUALITY

If your favorite dealer fails to have the games you want, call us toll free 1-800-638-9292

## microcomputer games

® REGISTERED TRADEMARK OF MICROCOMPUTER GAMES, INC. DIVISION OF **The AVALON HILL Game Company**

Avalon Hill MICROCOMPUTER GAMES are compatible with the following computer systems:
Apple II®
Atari 4/800®
TRS-80 Models I/III & Color®
IBM P.C.®
Commodore VIC-20, CBM PET & 2001®


For specific information, such as machine compatibility, memory requirements, cassette or diskette availability and price, call us toll-free 1-800-638-9292 and ask for Operator 30, or write:
Avalon Hill Microcomputer Games, Dept. S-30
4517 Harford Road, Baltimore, MD 21214

# OPEN DISCUSSION

*Open Discussion gives you the chance to air your views and concerns, to seek answers to questions, to offer solutions or helpful suggestions, and to develop a rapport with other readers. It's what you make it, so share your thoughts, typed or printed, and double-spaced (please), in* Softalk's *Open Discussion, Box 60, North Hollywood, CA 91603. To ensure the inclusion of as many contributions as possible, letters may be condensed and edited.*

**Skirmishes on a Family's Front Lines**

My father-in-law has absolutely lost his mind over *Castle Wolfenstein.* For the last five months he has used his Apple for nothing else, and his study is wallpapered with castle maps.

The family is in despair. At dinner he relates the adventure of the day, complete with bullet counts and the number of SS men between grenade boxes. He tells us how he lures pursuing SS guards into side rooms where he robs them at gunpoint and leaves them stranded. He never fails to remind us that holding down the repeat key and space bar lets him open chests at double speed. He loves to describe blowing a hole in a wall so he can lob grenades at passing guards through it. His idea of ecstasy is recounting how he ran out the front door of the castle, seconds ahead of three agitated guards, and was rewarded by the sight of a pastoral landscape with a cloud the shape of a supersonic rabbit.

I desperately need to find a new game to distract him from his headlong pursuit of Nazi escapism. Any suggestions? Would *Wizardry* get his attention? He barely looked at *Robot-War. Softporn Adventure* amused him for a day. *Alien Rain* diverted him for nearly two weeks, but I need something more permanent. He's a General now, and today he found the war plans. It's only a matter of time before he gets promoted again. If that happens, I won't be the only field marshal in the family anymore.

Bruce D. Clayton, Mariposa, CA

**Hobbyist Turns Pro**

I bought my Apple over three years ago as a new toy. I didn't know a thing about hard, soft, and firmware. Learning Finnish was easier than learning assembly language. The hobby soon turned into a small (micro) business, and instead of learning how to program I bought programs to use. I became a user-hobbyist. One of the first things I wanted to do with my Apple was to maintain a mailing label database for the local Mensa chapter. With more than fifteen hundred names, the task seemed insurmountable.

Along came Bob Clardy and Synergistic at the West Coast Computer Faire two years ago. He had a neat little program called *Mail List Database* for sale for thirty dollars. A quick look at the manual and a demonstration of the fast sort on a binary file produced a sale. That program turned out to be one of my most used. It took three of us seven days to enter the members' mailing information onto three disks. Later, Synergistic came out with a sort/merge utility for the program. This allowed us to merge up to five files into a zip sort. The time required each month to enter new members and make changes of address was less than one hour because of the ease of entry and the error-free operation of the program. The runoff of labels also was fast, considering the old printer used.

The time came when we outgrew the need for labels, but we did need to maintain a local roster as we made print-outs available to local members once a year. Early in the game, I wrote to Bob Clardy about the possibility of printing the information on a single line, roster style. He wrote back some time later with a few listing changes, and I had a roster program. An updated *Mail List Database* has incorporated this feature. Because of our long-distance customizing sessions, I actually learned a good deal about Basic programming.

But the best was yet to come! Late last summer several of us in our group wanted to start publishing a literary magazine. The first couple of months we generated labels as we had before. But we foresaw the need to keep track of subscription numbers, expiration dates, and so on. Another letter to Bob produced much help on customizing the program. Our *Subscription Mail Label Database* now consists of subscribers entered in order of subscription, a field with the expiration date, the usual name and address fields, and an extra long zip field to cover foreign subscribers' country names.

We now have more than five hundred subscribers and it takes less than two hours each month to update, sort by zip, and print labels. I have sent our aggregate customizations to Bob Clardy, and he mentioned at this year's fair that a technical note will soon be available to all registered owners that details these changes. In all the time that I've been using this (steadily over two years) I have not had one single problem with the software. It is completely listable, modifiable, and trouble-free. If this isn't testimony to the advantages of unlocked disks, I don't know what is. All the people I talked to at Synergistic were extremely willing to help, and my letters were answered promptly. This is not to imply that other publishers haven't been helpful, but I want to underscore my feelings when I get three hundred dollars' worth of service and program from only thirty dollars.

Tod Wicks, Palo Alto, CA

**Databases Ahoy!**

Open Discussion provides an excellent means for readers to learn about software that is, or is not, worth their hard-earned money. I would like to call attention to a software package that is definitely impressive: It's *The General Manager* by Sierra On-Line. I am so pleased with this program that I hardly know where to begin.

*The General Manager* is a hierarchical database manager. Hierarchical is a term to describe the way it structures a database defined by the user. Such data structures are built around a parent-offspring relationship (offspring can be parents of subsets of offspring also). Without delving too deeply, if you think about this scheme for a few minutes it will become apparent that you can have multiple databases on-line whenever you are running the program. Furthermore, databases do not have to be related. The system automatically seeks and loads into a RAM card, if available, upon booting. It will run on a single disk system if it has to, but two disk drives are recommended, and three are optimum. It will also work with a Corvus or Santa Clara hard disk system without any patch or program modification.

Its most impressive feature is that it can be interfaced to a user-written program. It will generate DOS 3.3 text files that can be edited with word processors and text editors such as *Screen Writer II, Apple Writer,* and *Apple PIE.* The database can even be proofread by *The Dictionary, The Sensible Speller,* and a number of other proofreading programs. The text file feature gives the program incredible flexibility. You can either use the very capable built-in report module or customize your reports using your word processor.

Sierra On-Line devoted a whole chapter in the very thorough manual to explain how to write programs that access your database. Contrast this with the paranoia exhibited by many other publishers when it comes to revealing the inner workings of their software. The databases you build allow you to specify types and lengths of fields, create screens to look like forms that you are now using, output information from one field to another, or create a field solely for displaying data generated from computations performed on other fields. After you create your database, but before you begin using it, *The General Manager* runs a syntax check on it to make sure that your specifications are correct. Nice touch!

I could go on and on. I checked out such programs as *The Data Factory* and *DB Master* before I settled on *The General Manager.* Although my final choice was the least expensive of the many I tried out, I truly believe that it is the best at any price. It would be worth anyone's time to take a close look at this package if they're in the market for a powerful, user-friendly database management system. It allows the user to dictate his or her own terms instead of having to conform to some programmer's idea of how people should manipulate information.

J. M. Tarrani, USS *Tarawa*

**Sharing Time and a Good Experience**

I would like to commend Ed Magnin of Telephone Software Connection for his excellent *Terminal Program* and, more important, for his excellent support. I purchased the program via modem to use with my university's IBM system using the MUSIC timesharing system. When I called TSC after experiencing some difficulty with properly configuring to the half-duplex protocols of the IBM system, Ed placed several long-distance calls to our system in order to test patches to the program. He quickly modified the program to operate properly.

The program is versatile and reasonably priced. I recommend this company without qualification. Some of the big names in software should give this level of support.

Bill Allbritten, Murray, KY

**Keeping up with the Dow Jones**

As a result of your excellent article about *The Dow Jones Market Analyzer* I purchased the program. It does everything as noted, and the documentation is excellent and user oriented. Vendor support is also excellent as I received a prompt response to a written inquiry from RTR Software.

One of the features of the program that was not noted is its ability to treat most any indicator as a stock and use it as a base in making comparison charting. For example, I put in the Dow Jones closing prices for the last year, treating them like a stock with the name "DJI." Other indicators or indexes could be treated the same way. The only added requirement to chart this would be compiling the historical data, as it is not available for retrieval from Dow Jones.

The main disadvantage to the program is the lack of speed in charting. About one hundred fifty units (days, weeks, and so on) takes about a minute to project. Secondly, there is the lack of a menu printer program. The documentation does include suggested user jump routines that may be set up in the program disk by a program-wise user.

Merle Zmak, Clayton, CA

**More PIE Please**

Allow me to echo *Apple PIE* diehards and say to the new *PIE Writer* users that either *PIE* is an outstanding word processor. Because of the need to place formatting commands within text, and the inability to see exactly what the finished text looks like on screen while entering text, *PIE* is initially more difficult to learn than other word processors. However, *PIE* has the ability to enter and edit without having to toggle from an entering text mode to an editing text mode. This shortens total time at the keyboard.

I am attempting to do a mail list merge with a database sequential text file and *PIE.* Presently, *PIE* can only read a *PIE*-produced file. From each record it will read the fields sequentially. The fields must be separated by the block terminating character (<). The use of a file like this is limited. A file made for one form letter may not be compatible with a different form letter.

Does anyone out there know how to do a patch to *PIE* so that it will, with a new format command *.field x* (or with something more compatible that does not conflict with the *.fi* command), read field *x* from a file when requested and include it in the document? This would allow *PIE* to scan for a field when requested and not have to look at each field sequentially. The records would be sorted by the database and placed in a sequential text file for use in *PIE. PIE* would pick the field from each record and include the necessary information in the document. This feature would greatly enhance *PIE* and make it probably the most powerful word processor around.

Roland Leong, D.M.D., New York, NY

**A Verdict of Acquittal**

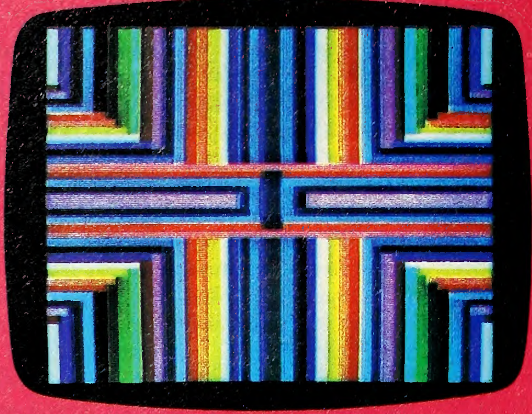
It was with mixed emotions that I read your review of *Client Management System II* in September *Softalk*'s Marketalk Reviews. It was for the most part extremely positive. For this I am extremely pleased as well as grateful. Our phone has been ringing off the hook, as calls keep coming in from all parts of the country since that issue hit the stands.

The criticisms, in part, were absolutely invaluable. Because of this, we have made a few changes. Back-up copies are now furnished at no charge upon receipt of a completed product registration card. Furthermore, the instruction manual is currently being reorganized and rewritten to include a hands-on tutorial.

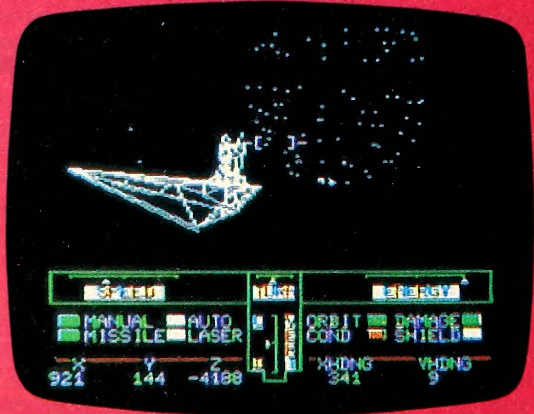
I must, however, take exception to the following statement that appeared toward the end of the review: ". . . the system, when configured as suggested in the manual, uses about thirty disks; in a large, busy office the disk storage could easily require more than a hundred disks and frequent disk swapping." A hundred disks is about fourteen megabytes! Fourteen megabytes on mini-floppy disks! Out of the more than one hundred attorneys that are currently using our software nobody is using nearly that many disks. The only offices that are using any more than eight disks are those that require each attorney to have his data kept separate in accordance with his own accounting requirements. Our Apple III version, using a ProFile five megabyte hard disk drive, will hold a thousand open cases, plus hundreds of critical dates cases!

According to our research, more than 90 percent of the law offices in this country consist of four attorneys or less. Our experience shows that the average attorney bills approximately forty to seventy cases per month. *Client Management System II* (mini-floppy version) holds more than 260 cases, on eight data disks. Your figures state that the system requires from thirty to one hundred disks (enough storage to hold from 990 to 3,300 cases). However, don't sell us short, *Softalk.* Our enhanced Apple III version (scheduled to be introduced at the San Bernardino Bar Convention in October) as well as our IBM pc and Displaywriter versions (scheduled to be introduced in December) will easily handle numbers in excess of those mentioned here.

I can only assume that this mix-up was the




Night
Mission
CREDITS BALL


MANUAL MISSILE
AUTO LASER
ORBIT COND
DAMAGE SHIELD
921 144 -4188
XHDNG 341
VHDNG 9

result of a misprint in your magazine, or our documentation was unclear about this point. On a lighter note, as an Apple user for more than four years and an avid *Softalk* reader, I wish to express my deepest gratitude for your magazine. It is by far the absolute best in terms of content and presentation.
David J. Kalmick, vice president,
Compu-Law, Culver City, CA

**Harvesting the Slack**
You now have the notable distinction of being the first major computing magazine to feature computers in agriculture. Congratulations!? Computing farmers have been around as long as computing anybody else. Farming is as sophisticated a business as any other business; the potential for farm computerization has never been more potent. So where has everybody been?

We realized over a year ago that the computing farmer was going to be neglected, so last winter we started our own full-fledged, farm-computing magazine (not mentioned in your article). We now have readers all over the continent wanting to know the things nobody else bothered to find out for them. Please tell your readers (do you even know how many of them are farmers?) that there's a magazine that pulls up the slack you other guys left behind.
Mark Gallagher, *AgriComp* magazine, 1001 E. Walnut, Suite 201, Columbia, MO 65201

**Micro Scopes in Focus**
The article in August *Softalk* by David Hunter titled "Down on the Farm with Apples" bothers me.

I think the author did an excellent job in researching and writing about the little-known and growing use of Apples on America's farms. It is certainly to be expected that our farmers would adopt and adapt the latest technology in their continuing quest to increase their incomes. This "selfish" motive has enabled them to outproduce most other farmers in the world.

Yet, the article begins on a contradictory note. In the preamble Genesis is quoted, in which Yahweh comments on Noah's sacrifice to him after the Flood, stating that he will never again "smite every living thing." The article proper then begins: "About ten thousand years ago, our Neolithic ancestors discovered agriculture. Before, for untold millenia . . ."

First the author quotes God, who, true believers hold, created the universe a few thousand years ago. He then expresses the usual recount of evolution that requires billions of years for its workings. It should be realized that, for all intents and purposes, the geological ages are synonymous with evolution.

The point is that either one or the other is true. Either God created the universe no more than some eight thousand years ago or life began perhaps four billion years ago in a warm mud puddle and blindly squirmed its way upward and onward, branching into different forms thousands of times and culminating in man. Either one or the other: there is no compromise. Any and all attempts to harmonize evolution and the Bible have been, or can be, shown to be entirely and utterly useless and nugatory.

Consider the fact that it is truly impossible to scientifically prove or disprove any particular concept of origins. The essence of the scientific method is experimental repeatability. The most outstanding scientist is incapable of observing or repeating origins. Incidentally, since scientists now reserve the term "theory" only for propositions that can be experimentally proven or disproven, evolution does not qualify for that classification.

Then, contrast the laws of thermodynamics with the fundamental tenet of evolution—that it is a process, essentially irreversible, that tends to ever higher levels of organization. If there is any law firmly established in modern science, it is the second law of thermodynamics, of energy decay. This law states that every system in isolation always tends to move from order to disorder—its energy tending to be transformed into lower levels of availability, finally reaching the state of complete randomness and nonavailability for further work. For the universe, this leads ultimately to heat death. Obviously, this second law by itself negates evolution.

David Hunter's article is excellent insofar as its main theme is concerned. But it starts off on the wrong theological foot so far as Christian believers, many of whom are farmers, are concerned.
Howard S. Balsam, Nashua, NH

**National Logo Invitational**
*The National Logo Exchange* is a monthly newsletter dedicated to the Logo teachers in the classroom. Each issue contains articles by fellow teachers, columns by informed educators, reviews, listings, tips, techniques, and opinions. As the practical methods of teaching with Logo emerge, we wish to serve as a vehicle for the exchange of ideas and strategies, and to help with the development of working Logo philosophies.

We would like to issue to David Greene and any other interested readers of *Softalk* an invitation to share their thoughts and experiences with other teachers by writing short articles for publication. In general, our readers are interested in practical matters: ideas and techniques that can be put to use right away.

For further information, write to The National Logo Exchange, attn: Susan Thompson, Box 5341, Charlottesville, VA 22905.
Tom Lough, editor, *The National Logo Exchange*, Charlottesville, VA

**Resounding Novation**
David Palmer's letter in September Open Discussion prompted me to finally write about what I consider to be the most monumental example of product misrepresentation in the microcomputer industry. The product, the Apple-Cat II by Novation, has been widely advertised as offering communication capability up to 1,200 bps. I had the misfortune to take Novation at their word.

Last May, I purchased an Apple-Cat II with the 1,200 bps capability. When I was unable to communicate with a mainframe computer at 1,200 bps (300 bps worked fine), I called Novation. Novation informed me, after some hemming and hawing and switching me from one party to another, that their Apple-Cat II would, in fact, communicate with another computer at 1,200 bps—but *only* if that computer was another micro with an identical Apple-Cat II setup. This limitation never quite gets into Novation's ads.

I certainly hope the new Hayes modem, which has just been released and is being advertised as having 1,200 bps capability, is truly capable of doing what the manufacturer claims it will.

Robert Berman, Vienna, VA

*The manufacturer responds:*

We at Novation are most unhappy to learn of the confusion that our advertising has caused. The information subsequently given to Robert Berman also appears to have been too narrow and somewhat confusing to him. I would like to attempt to clear the air by providing some additional information on the subject of Novation's product offerings.

Our Apple-Cat II product line is composed of two main units. The first is the basic Apple-Cat II board which plugs into a single slot of a standard Apple II. This board, when properly installed and operated, will support dial telephone line communications at data rates of 110, 300, and 1,200 bits per second. More properly speaking, we say that the Apple-Cat II is online compatible with the Bell 103 and Bell 202 standards.

The 103 modem standard is full duplex, meaning data may pass in both directions at the same time. It covers the speed range of 0 to 300 bps. The 202 standard is half duplex, meaning data passes in one direction at a time only, and it covers the speed range of 0 to 1,200 bps.

The Apple-Cat II product can play with computers that are equipped with modems compatible with these Bell 103 or Bell 202 standards. It is most definitely *not* the case that it will play only with Apple-Cat equipped systems.

Quite the contrary; we provide with every Apple-Cat II, at no additional cost, a complete disk-based communicating operating system called *ComWare.* This software will turn the Apple II into a communicating terminal at speeds up to 300 bps, full duplex, or 1,200 bps, half duplex. No *ComWare* is needed at the other end, just a computer that has a Bell 103 or 202 compatible modem. The software will do a lot more than this, such as file transfers, auto dialing from software directory, and so forth, but let's leave that for some other time.

In addition to the *ComWare* software, we also offer as an option a ROM-based firmware program. This program simplifies the use of the Apple-Cat II from user-written software and eliminates the need to boot a disk before operating the modem card.

The second main Apple-Cat product is our 212 Expansion Module. This is a second printed circuit board that can be mounted in a second Apple II slot or, optionally, can be operated completely off the motherboard using our Slotsaver kit. The 212 Expansion Module increases the capability of the Apple-Cat II. It permits communication with any Bell 212A compatible modem, operating at speeds of 1,200 bps in full duplex mode. Most of the major public database systems allow access through the Bell 212A.

Both our firmware ROM and the *ComWare* software fully support use of the optional 212 Expansion Module. We believe that we are currently the only company offering an integral 212 compatible product for the Apple II computer.

I'll be happy to explain further any details that remain unclear to Apple-Cat users or potential users of these Novation products.

David Lyon, director of engineering, Novation, Tarzana, CA



**Hung up on Students**

I have written a database program to use for student records and schedules, and I've followed pretty much the same format used by most databases. In the program I store thirty-six pieces of information on each student. The information is input as strings, one at a time. After each of the student's data strings is complete, it is stored in a random access data file before continuing to the next one.

When a series of students is entered, the computer hangs up right in the middle of inputting the information. I am forced to hit reset and run the program again to continue. This problem always seems to occur in the same place, about halfway through the thirty-six entries on the fourth student being inputted.

In September Open Discussion, a letter from Charles Wells indicated that he was having a similar problem with *File Cabinet.* What's causing this problem and what can I do to eliminate it?

Lynn Leopard, Chillicothe, MO

**Stringtime in November**

Charles Wells's letter in September Open Discussion gave a good explanation of why Applesoft Basic sometimes gets uncommunicative while it does garbage collection on the strings that it no longer needs in storage. He pointed out that this is the price Applesoft makes us pay for being allowed variable-length strings.

In addition to the nuisance of having your Apple go out to lunch at unexpected moments, the length of time Applesoft takes for garbage collection can get downright frustrating if your program uses a lot of string variables. Here is a short program that allows you to time the garbage collection function:

```
10 INPUT "HOW MANY STRINGS?";N
20 DIM A$(N)
30 FOR I = 1 TO N: A$(I) = "B" + "C": NEXT
40 PRINT CHR$(7)"STARTING GARBAGE
```

# It's 2 AM. Your Apple II is sending budgets to New York; purchase orders to Boston; a contract to St. Louis; and correspondence to every field rep in the country. Automatically. Transend, from SSM.

Transend software opens the world of electronic communications to your Apple. From sending mail over the phone to connecting with information banks—the doors opened by Transend are practically unlimited.

Take electronic mail. SSM's Transend 3 lets your Apple correspond *automatically* over phone lines with *up to 100* Apples—at any hour you choose. Your Apple dials each phone number, transfers data electronically, verifies that the data was transmitted intact, and provides a complete call status report. Transend 3's password feature means only authorized users see sensitive information.

## 8:37 AM. Your Apple receives mid-morning sales reports from the Apple in your New York office.

Transend 2 lets your Apple correspond over the phone with other Apples. Error detection features guarantee the accurate transmission of your valuable data.

## 1:52 PM. Your Apple displays current flight schedules and connects you to a ticket agent via The Source℠.

SSM's Transend 1 turns your Apple into an intelligent terminal connected to your corporate computer, a timeshare system, or any information service such as The Source. (In fact, all Transend software includes a subscription to The Source.)

You'll get business news, the most recent stock reports, advance UPI world news—even flight schedules—in moments. And new possibilities arise constantly.

## 11:53 PM. With the SSM Apple ModemCard™, your Apple is always ready to dial the phone.

SSM's 300 baud modem card fits conveniently inside your Apple. Advanced features include Auto-dial/Auto-answer for unattended operation, and Touch-Tone™ dialing (required for networks such as Sprint® or MCI Advantage™). The SSM ModemCard makes all other modems obsolete.

## 2 AM. The SSM Apple TimeCard™ lets your Apple work while you sleep.

The SSM Apple TimeCard lets your Apple Transend mail automatically at any hour. It keeps accurate time for other uses as well.

## Need help?

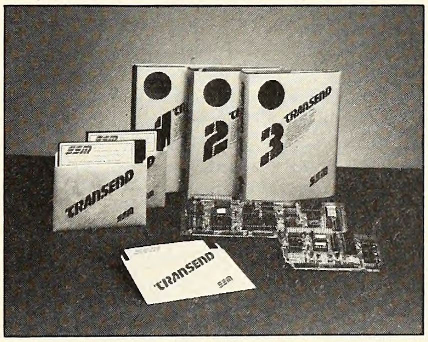
Simply Transend your question to SSM's 24-hour electronic message board. The next business day, your answer will appear on the board for easy reference.

## "How do I Transend my Apple?"

Easily. Just go to your dealer for a flexible choice of SSM products. You can purchase easy-to-use, prepackaged hardware/software combinations, or separate components to fit your needs and budget. SSM guarantees your satisfaction or your money back.

SSM Microcomputer Products Inc.
2190 Paragon Drive, San Jose,
CA 95131 (408) 946-7400,
Telex: 171171 SSM SNJ

The Transend family from SSM: Transend 1 (intelligent terminal software that lets your Apple talk to virtually any computer, including information services), Transend 2 (software that lets your Apple send verified electronic mail to other Apples), Transend 3 (full-featured electronic mail software with automatic mailing capabilities), the SSM Apple TimeCard, and the SSM Apple ModemCard.




SSM

Transend it.

```
   COLLECTION"
50 PRINT FRE(0)
60 PRINT CHR$(7)"DONE"
```

After you tell this program how many strings you want the garbage collection to work on while you time it, the program stores the strings in an array. Then it rings the bell so you'll know to start timing when the garbage collection begins, and the bell rings again when the garbage collection is finished. After you've experimented with timing the garbage collection for small numbers of strings (between 100 and 1,000), you're ready for the big time. Run the program again and tell it to work on 5,000 strings. Note the time when the first bell rings, and head for your back issues of *Softalk* to find those articles you always meant to reread. You'll have plenty of time—Applesoft's garbage collection for 5,000 strings takes about thirty minutes. Try it!

I got so frustrated at the slowness of Applesoft's garbage collection that I decided to write a program to do this function more quickly. As soon as I have finished writing the documentation for my program, I will begin exploring ways to make it available so others won't have to suffer so long while the Apple is out to lunch.
Richard Norling, Washington, DC

**CATS in the Public Domain**

In September Open Discussion Ellis McDaniels submitted an excellent letter stating the need for freer exchange of information concerning development of educational software. While I am a speech-language pathologist and not a teacher, users of microcomputers in rehabilitative settings share many of the problems expressed by teachers such as Mr. McDaniels.

The purpose of this letter is to inform those interested that the Audiology and Speech Pathology Service at the Veterans Administration Outpatient Clinic (425 South Hill Street, Los Angeles, CA 90013) has developed a series of reading treatment and testing programs on a single disk for the Apple II Plus. These programs are collectively called *CATS* (*Computerized Aphasia Treatment System*) and can be listed and copied; *CATS* is not copyrighted and is under public domain. While *CATS* was written for adults suffering from aphasia (a language disorder resulting from brain injury), it is also appropriate for younger people in need of reading testing and treatment.

*CATS* is available free of charge. Anyone interested in getting more information about *CATS* should write to me at the above address.
Richard C. Katz, Ph.D., Los Angeles, CA

**One Reader's Good Reading**

In reference to Ernie Brock of Sirius, whose letter questioning *Softalk*'s polling procedures for Bestsellers appeared in the September issue—I say, more power to *Softalk*! It seems that Mr. Brock's capitalist dreams are influencing his ideas, and Sirius Software too for that matter. He certainly should research his accusations before making them.

While on the subject of Sirius, I think that calling the people who sell *Locksmith* crooks is way out in left field. What is really crooked is charging five dollars for a replacement disk no matter what the problem is. Come on, what kind of policy is this? Perhaps Sirius and others in the industry should take a look at what the Carlston family is doing. I'm not even going to bring up the Joyport fiasco.

Now, on another subject, I think *Softalk* should print examples of what the programs listed in the tutorials should look like when they're run, if it's applicable. Either that, or a very clear explanation of what's supposed to happen when they're run. As it is, I'm not always sure if a program is running correctly, and I'm almost positive that the shape maker program isn't working properly.

Lastly, isn't that Bill Budge pictured at the opening of Beginners' Corner, of all places?
Thomas E. Burns, Stevensville, MI

*Bill Budge a beginner? Ridiculous! Or is it? Let's ask our readers. Drop a postcard to Softalk Budgie, Box 60, North Hollywood, CA 91603. If you think the "beginner" is Bill Budge, write "It's him all right!" If you don't think it's Budge, write "No way!" Also include your name and address. If it is Budge, one person drawn from all the yes cards will win a copy of the real Bill Budge's new game (whenever it arrives). If it isn't him, all no way answers will receive a set of* Softalk *InvisiTabs—and one person chosen at random will still get a copy of Bill Budge's next game.*

**All About Being Clobbered**

I was playing around with a program from All About Applesoft by Doug Carlston that appeared in *Softalk*, March 1982, and changed the following lines to:

```
 5 PRINT "Type a number between 0 and
   255": INPUT A%
10 POKE 1024 + X, A%
20 X=X+1: IF X<1024 THEN 10
```

This worked fine; then I added:

```
30 GOTO 5
```

and this resulted in a syntax error message. I ran it again, and the result was:

```
?syntax error in 65054
```

This clobbered the program, and even with line 30 deleted the same error message resulted. Typing *new* gave me a syntax error message but did work. I reentered the program without line 30, ran it, and got the same syntax error message in line 65054.

Next, I turned off the Apple and just started over, adding line 30, which worked once but then gave me a syntax error in line 30, then a syntax error in line 65054 over and over again. The only thing I can find out about 65054 is that it's in ROM. What's happening here?
Eloise King, Big Timber, MT

Adding *goto 5* to the end of the sample program put the entire program into a loop, as you intended. However, it did not reset the value of *x* to zero. The first time through the program, *x*

**RH ELECTRONICS, INC.**

COPYRIGHT © 1981 — PATENTS PENDING

566 Irelan, Buellton, CA 93427

(805) 688-2047

**8:00 TO 5:00 CALIFORNIA TIME**

**COMPUTER PRODUCTS**

DESIGNING • MANUFACTURING
ELECTRONIC ENGINEERING

# SUPER FAN II™

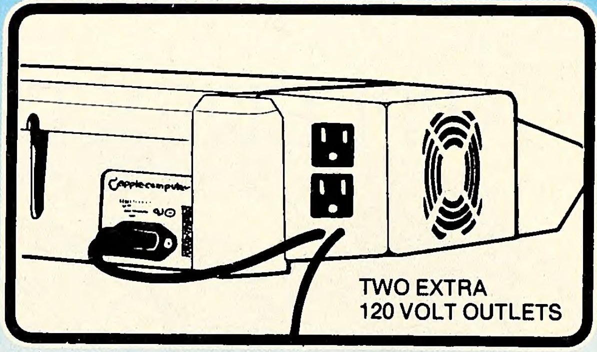
FOR YOUR APPLE II* COMPUTER

*One Year Warranty*

**$74.95**

**With Zener Ray™ Protection $109.00**

**MASTERCARD — VISA**




## *"COOL IT"*

- ALSO FITS ON APPLE'S* NEW MONITOR STAND
- RED PILOT LIGHT ON/OFF SYSTEM SWITCH
- CLIPS ON — NO HOLES OR SCREWS • REPLACEABLE SWITCH
- AVAILABLE IN 120V or 240V AND 50/60 HZ • DURABLE MOTOR
- REDUCES HEAT CAUSED BY EXTRA PLUG-IN CARDS
- SOLD WORLD WIDE • UNIQUE 1 YEAR WARRANTY
- TAN OR BLACK COLOR • QUIETEST FAN ON THE MARKET
- INCREASED RELIABILITY — SAVES DOWN TIME AND REPAIR CHARGES
- LOW NOISE DUE TO DRAWING EFFECT OF AIR THROUGH YOUR COMPUTER AND SPECIAL FAN AND MOTOR DESIGN
- TWO EXTRA 120V OUTLETS FOR MONITOR AND ACCESSORIES TURN ON WHEN YOU TURN ON YOUR FAN (NOT AVAILABLE ON 240V MODEL)

## SUPER FAN II™ WITH ZENER RAY OPTION $109.00

## ZENER RAY™ TRANSIENT VOLTAGE SUPPRESSOR

OUR BUILT IN ADVANCED DESIGN UNIT GIVES
**DRAMATIC COST SAVINGS** — STOPS ANNOYING DOWN TIME
**INSURANCE FROM** VOLTAGE SPIKES - GLITCHES
**DANGEROUS** VOLTAGE SPIKES CAN JEOPARDIZE YOUR COMPUTER SYSTEMS
**PROTECT** COMPUTER - DISK DRIVE - PRINTER AND MONITOR

NO CUTTING WIRES • WON'T VOID WARRANTY, JUST PLUG IN SUPERFAN II WITH ZENER RAY

*OTHER PRODUCTS BY* RH ELECTRONICS, INC.

**SUPER RAM II™** 16K RAM CARD FOR YOUR APPLE II. 2 YEAR WARRANTY $125

**GUARDIAN ANGEL™** AN UNINTERRUPTABLE POWER SOURCE $595

**12 VOLT TRANSVERTER** 12 VOLT — RUNS YOUR APPLE II COMPUTER AND AND 5¼" DRIVE FROM YOUR CIGARETTE LIGHTER $149

*Registered trademarks of Apple Computer Inc.

**DEALER INQUIRIES INVITED**

increased from 1024 to 2048. When you poke into these addresses, you are poking data directly onto the text page in memory.

However, the second time through the program, $x$ starts equal to 2048. This area of RAM is above the text page—in fact, it is where your program is stored. So the moment line 10 tries to poke some new value into 2048, it louses up your program.

If you would like to put the program into a loop so you can try different characters, you could use either.

```
30 RUN
```

or

```
30 X=0:GOTO 5
```

I hope this helps to clear up the problem.
Doug Carlston, San Rafael, CA

**His Own Boss**

I do not have the *DOS Boss* or *Bag of Tricks* utility, but I wish to reskew my disks. I would appreciate it if someone could reply through Open Discussion telling me how to reskew without having to purchase a utility package.
Marco Matchefts, Colorado Springs, CO

*Check out "The Speed Sector" by Don Worth and Pieter Lechner, an article that appeared in this year's April issue. Also, a couple of months later in August Open Discussion, Don Worth offered further clarification in the letter entitled "Just Skewing Around." Back issues, or reprints of any article, may be had by writing Softalk Back Issues, Box 60, North Hollywood, CA 91603.*

**Right in the Corner with Light**

In the August Beginners' Corner it is stated that using more than two or three if-then tests on string variables in the same program will cause a crash, according to Apple. *The Applesoft Basic Programming Reference Manual* (page 77) appears rather ambiguous on this: "*If* (string) *then* . . . when executed more than two or three times in a given program, causes the message

```
?FORMULA TOO COMPLEX ERROR
```

to be printed."

In the example given in Beginners' Corner, however, the variable (A$="Y") being tested was not a string at all. It would be evaluated simply as a variable. Moreover, Apple's caveat does not encompass string variables, only strings. At least on my Apple II Plus the following program never hangs:

```
10 GET A$
20 FOR K=1 TO 100
30 IF A$ THEN X=1
40 NEXT
```

Nor does it hang when I substitute for line 30 *if A$="Y" then X=1,* the specific test that concerned you.

Evidently the only variation on this theme that creates a problem is of the form *if "A" then. . . .* This will produce the error message as predicted. And it is, of course, a string—not a variable, not a string variable. It's a *string constant*, a term never employed by Apple, perhaps because they perceive it as redundant. After all, if it's not identified as a variable, then by default it's a constant, right?
John W. Field, Fairfax, VA

John Field is absolutely correct. I confess to misreading the manual, but I wonder why anyone would ever want to use *if "A" then.* This is logically exactly the same as *if "a"<>0 then*, which produces the "?type mismatch error" message. When used with a string (but not a string variable) both are completely meaningless when you remember that <> means "neither greater than nor less than" in the mathematical sense rather than "not equal to" (that is, not the same as) in the string sense. Nevertheless, since Applesoft doesn't realize that there's an implied conflict between types, it evaluates the expression anyway. Try in your example first

```
30 IF "A" THEN PRINT SOFTALK
```

and then

```
30 IF " " THEN PRINT SOFTALK
```

The first treats the string as something and prints *Softalk* two or three times before bombing; the second, recognizing the null string as nothing, prints nothing and causes no line feeds but bombs anyway. You'll also notice that *if "a" then* doesn't respond properly to *onerr goto* while *if "a" <> 0 then* does. Ah, the mysteries of Applesoft.
Christopher U. Light, Chicago, IL

**Indirect Access**

Can anyone tell me how to find out if a second disk drive exists from within a program without accessing the disk drive? Obviously a DOS command could be used, like *verify* (file name), but there you're accessing the drive.
Chris Riley, Pompano Beach, FL

**Missing the Value . . .**

I'm heavily involved with the Apple II, mostly in a teaching role, normally using Applesoft. As the local Apple polisher, I get asked many questions about other features for the Apple. One frequent area is that of UCSD Pascal, with which I'm only slightly acquainted.

It appears to me that the language is an abortion, and certainly unworthy of the lavish praise many heap upon it. It was so obviously designed by professional programmers operating from a mainframe, rather than designed for user convenience in personal computers, that it makes me puke.

One question that comes up quite frequently is, "How do I turn the printer on?" There is obviously no equivalent of *PR#1* or *Lprint* in the language, or at least in the Apple implementation of it. How could Apple Computer ever endorse a product with such a fundamental weakness? They must have glitter in their eyes; unfortunately, all that glitters is not gold!

With no thanks to Apple, their instruction manuals, or several Pascal for the Apple books, I have finally figured out how to use the printer to dump a disk catalog list or to transfer a text file to the PRINTER:, though hardly a sophisticated implementation. I've also figured out, after fruitless searches through the gobbledygook of the language manual, how to send output to the printer during a running program. Creating an imaginary file and writing to it is hardly what one would call user-friendly.
Chuck Walker, Lincoln, NE

**. . . Of Bilingual Doublethink**
To Chuck Walker: Use of the Apple Pascal filer utility, which permits you to send an entire file to the PRINTER: at once or list disk directories on the PRINTER: or screen, is the subject of a future Pascal Path. I hope that my past and future efforts may be of help to you and those you advise.

We always seek to explain the unfamiliar using familiar terms. Being unfamiliar with the Apple Pascal environment, you quite naturally and reasonably want to have it explained in terms of a system you understand, namely, Applesoft. You need a Pascal to Applesoft dictionary, which neither Apple nor The Pascal Path provides. I do recall, however, seeing one Pascal for Basic programmers book during my last visit to the bookstore. Check the sources for computer books in your area to see what's available, and good luck!

I'd like to issue a word of warning, though, to any experts in Basic who want to learn Pascal. You never become fluent in a foreign language as long as you persist in mentally translating between it and your mother tongue. This also holds true for programming languages. To use any language most effectively, you must first see problems and solutions in ways amenable to expression in that language. In other words, you must think in Basic, or think in Pascal, to take fullest advantage of either.

On the other hand, experienced Basic programmers often remark on the neatness and clarity of Basic written with a "Pascal accent," while Pascal programmers easily recognize (and grimace at) Pascal programs written by those who customarily work with Basic. Similarly, I enjoy hearing English spoken with a French accent (even if the grammar isn't always perfect), while the French find it appalling to hear their language "butchered" by Americans.

In any case, the task of learning a new programming language is made more difficult by any preconceptions (or misconceptions) that you bring with you from your experiences with other languages. For instance, why should Pascal include special output instructions tailored for the PRINTER: (such as Applesoft's *Lprint)* when regular, file-oriented I/O is equal to the task? For those with a background in Basic, I caution you to explore thoroughly the motivations and history behind a Pascal construct or feature before forming your opinion of it. As I try to show in my column, there is a reason, usually quite persuasive, for every Pascal feature (or lack of one).
Jim Merritt, Morro Bay, CA

**Positive and Negative Overflow**
This is a modified version of "Short Work of a Short Routine" from September Open Discussion. It places a minus sign immediately before a dollar-formatted number, when needed, in a field of length *L*. If overflow occurs, the string *V$* is filled with ****, except for the leftmost character position which either remains blank for positive or contains a minus sign for negative overflows. In line 61010, *V* is checked both for length and magnitude; without the latter check, an incorrect cents value could result for very large (exponential-format) numbers. Without the length check, the most significant figures in a dollar amount could be omitted. Line 61005 should be included to guard against near-zero numbers in exponential format, which may result from a prior computation of *V*.

```
61000 V$=" ":IF V<0 THEN
      V$="-":V=ABS(V)
61005 IF V<.005 THEN V=0
61010 IF V>9999899 OR V>10^(L-4)
      THEN V$=V$+RIGHT$
      ("********************",L-1):RETURN
61020 V$=RIGHT$("                   "+
      V$+STR$(INT(V+.005))+"."+
      RIGHT$(STR$(INT((V+100)*
      100+.5)),2),L):RETURN
```

To illustrate the subroutine's use, try this:

```
10 L=10 : X= (expression) : V=X:GOSUB
   61000
15 P$=V$
20 L=15 : Y= (expression) : V=Y:GOSUB
   61000
25 P$=P$+V$
30 L=12 : Z= (expression) : V=Z:GOSUB
   61000
35 P$=P$+V$
40 PRINT P$
45 GOTO 10
```

If the expressions used to compute *X*, *Y*, and *Z* yield reasonable dollar figures, the example print should be three properly aligned columns. Just for fun, try using the expression *X=−IE5+IE7*(RND(1))^6* to get some really interesting results!
Paul T. Burnett, Alamogordo, NM

**Just a Modern Guy**
I wish to say a few more things on the subject of *mod*. I noticed very early in using Applesoft that it was lacking the *mod* function that can be quite useful. *Mod* stands for modulo and that is a mathematical term for number systems where the remainder is used to represent a number. Generally, it is only defined in integer number systems. To illustrate the idea, consider that numbers like 47, 157, and 37 have something in common. That is, the right-hand digit is the same. A way to show this is to drop all multiples of 10, which is the base, of course. The number 157 is really 10x10x1 + 10x5 + 7. Dropping multiples of 10 leaves 7. We say that 157 =

7 modulo 10. Similarly 47 = 7 modulo 10 and 37 = 7 modulo 10. The three numbers are said to be congruent using the base 10. You might recall from plane geometry that congruent figures are identical in side length and angles. In other words, they are identical. Using modular arithmetic we might say that if X = 157 and Y = 47 that X = Y (modulo 10).

Modulo can use any number as a base. Using a base of 6, we might say that 8 is congruent or "equal" to 2. This is because removing multiples of 6 leaves 2 in either case. We can also add multiples of 6 to prove congruency. For instance, −4 + 6 = 2. So −4 is congruent to 8 and 2 also.

Now let us try to do the same operation in Applesoft. We want to remove multiples of the base, so we might say:

```
MODNUM = NUM − INT(NUM/BASE) *
  BASE
```

The *int* function finds how many times *base* divides into *num* without exceeding it. We would like to use a function like:

```
DEF FN MOD(N,B) = N − INT(N/B) * B
```

Unfortunately, Applesoft only allows one parameter so we have to specify the base. We should also only use two-letter names to avoid confusing them:

```
100 DEF FN M6(N) = N − INT(N/6) * 6
110 Y = FN M6(10)
120 PRINT "Y = "Y
```

Running this program produces *Y = 4*. This is the modulo 6 value of 10.

I find the modulo function useful for wraparound phenomena. Suppose we want to print a set of sixty-two characters on a line of twenty-six columns. We could specify the *x* location to use by the character number modulo 26 (line length). Here is the program:

```
100 C$="ABCDEFGHIJKLMNOPQRSTU
    VWXYZabcdefghijklmnopqrstuvwxyz
    0123456789"
200 FOR I = 0 TO 61
300 X = I − INT (I/26) * 26
400 Y = INT (I/26)
500 HTAB X + 1: VTAB Y + 1
600 PRINT MID$(C$,I + 1,1)
700 NEXT I
```

If the MOD function were available, statement 300 would be:

```
300 X = I MOD 26
```

Pascal has a function in addition to *mod* which is an integer divide called *div*. If that were available in Applesoft, statement 400 would be:

```
400 Y = I DIV 26
```

The modulo function always uses a zero origin starting point. The *HTAB*, *VTAB*, and *MID$* functions all use a unit origin which causes the "*+ 1*" in those cases.

The modulo function is also used to calculate the low order byte of a computer address, namely, address modulo 256. Another application area is in error correcting code theory. Modulo 2 (which is equivalent to *exclusive-or*) is the fundamental operation.
Rich Hatcher, Plano, TX

**A Shoe In**
Does anyone have a program for use in a shoe store? I need something to use as a general ledger and for stock control.
Harold Vaughn, Baytown, TX

**Mystery of the Meddling Monitor**
I am a high school student who enjoys programming with several of my friends on the Apples we have at school. Recently we encountered an interesting phenomenon that other readers of Open Discussion might help us explain.

We had two disk drives placed on top of an Apple II Plus with a green phosphorous NEC monitor sitting on top of the drives. Everything worked fine until one day some of our disks failed to boot. The drives would just whirl around indefinitely, never managing to find anything on the disks. At first we thought the problem was that the disks were old, but some of the new disks did the same thing, while others worked just fine.

Next we figured that the drives might be faulty, so our instructor decided to check the cards in them. While checking the cards he set the monitor aside. Nothing appeared wrong with the cards so he tried to boot the disks again, this time while the monitor remained sitting where he had placed it, beside the drives. Now they booted just fine. Apparently either the magnetic field produced by the monitor screen or just the weight of it pushing down on top of the drives was responsible for the problem.

I have never heard of a situation such as this. Does anyone have a more perfect explanation of the cause?
Tony Raney, Ozark, MO

**Oversight Uncovered**
I would like to make a comment regarding what I feel is an oversight by the manufacturers of the Apple. When one buys a typewriter from IBM, one of the standard items that goes along with it is a dust cover. That piece of machinery, up until a few years ago, was totally mechanical. If IBM thought it was important enough to cover a mechanical device from dust, I am sure it is just as important, if not more so, for the Apple to be protected from Mother Nature.
Stephen P. Bobko, Honolulu, HI

**Switch of Allegiance**
At first I did not side with those who felt that you should be publishing critical software reviews. This was because we were not paying for the magazine, and your first duty was to your advertisers. Now that your readers are charged for a subscription, it is as much your duty to steer us away from inferior software as it is to inform us of high-quality software.
David Temkin, Highland Park, NJ

**Boldface Lettering**
Open Discussion is my favorite section of *Softalk*. However, I do have one criticism. Each month an increasing number of letters refer to letters from previous issues. At the present rate, in six months no new material will appear—only a constant rehashing of old ideas.

I question printing any letter which argues that a previous letter should not have been printed. This is a common theme. Doesn't the second letter merely provide an additional reference point to the first? In the early months of *Softalk*, there was much discussion about what constituted appropriate advertising. Now I see letters asking that letters referring to the advertising not be printed. Finally, there is my own metaletter asking that these letters referring to the earlier letters not be printed.

Perhaps you will consider this letter to be a member of the class of letters that best not be printed. Would it help if I dared you to print this?
Bob Wiseman, Cincinnati, OH

**Sour Grapes, or Rather, Potatoes**
There is a reason I have written this letter on yellow lined paper from a pad rather than on a word processor. The reason is simple as well as sad. It is also the reason I am writing to Open Discussion.

Yes, I have an Apple II Plus, 48K, disk drive, and all. Yes, it works all right. Yes, I know how to use it. The trouble is, "it" and I don't get along. I'm stingy, and "it" always needs just one more accessory to do the required job. All the marvelous things it does best are of no use to me.

Gripe number one: I just don't like the display of program listings as I'm typing them. They are too close together, making them hard to read. It's even worse when they come back from a list command.

Gripe number two: There is no decent way to edit while programming; using I-J-K-M is just too cumbersome. Now don't tell me to buy something—I can't read the listing anyway.

Gripe number three: The keyboard feels gritty and there's no numeric keypad. Yes, I know what the response will be, but they're really nicer right next to the keyboard.

Actually, I do admire the audacity of the makers of the wormy thing. It does sell software, doesn't it? I'll never understand who has the patience to key in those programs. I would really appreciate any comforting words from some kind soul who can make me understand why they make the machine the way they do, or some suggestions that don't cost upward of $300. I remain frustrated in Idaho.
Mrs. George E. Heckler, Pocatello, ID

**Statement of Ownership and Circulation**

1.-2. Softalk (publication number 02749629); October 10, 1982.
3. Softalk is published monthly, 12 issues annually, at an annual subscription price of $24.
4.-5. Office of publication and general business offices are located at 11160 McCormick Street, North Hollywood, CA 91601.
6. Publisher: Al Tommervik, 11160 McCormick Street, North Hollywood, CA 91601. Editor: Margot Comstock Tommervik, 11160 McCormick Street, North Hollywood, CA 91601. Managing Editor: Patricia Ryall, 11160 McCormick Street, North Hollywood, CA 91601.
7. Statement of ownership: Softalk Publishing Inc., 11160 McCormick Street, North Hollywood, CA 91601; Robert Comstock, 11160 McCormick Street, North Hollywood, CA 91601; William Depew, 10432 Burbank Boulevard, North Hollywood, CA 91601; John Haller, 11160 McCormick Street, North Hollywood, CA 91601; John Mitchell, 10432 Burbank Boulevard, North Hollywood, CA 91601; Mary Sue Rennells, 11160 McCormick Street, North Hollywood, CA 91601; William V. R. Smith, 10432 Burbank Boulevard, North Hollywood, CA 91601; Craig Stinson, 11160 McCormick Street, North Hollywood, CA 91601; Margot Comstock Tommervik and Al Tommervik, 11160 McCormick Street, North Hollywood, CA 91601; Kurt Wahlner, 11160 McCormick Street, North Hollywood, CA 91601.
8. None.
9. Not applicable.

| 10. Extent and nature of circulation | Average number of copies each issue during preceding 12 months | Actual number of copies of single issue published nearest to filing date |
|---|---|---|
| A. Total number copies printed (net press run) | 81,122 | 97,927 |
| B. Paid Circulation | | |
| 1. Sales through dealers and carriers, street vendors, and counter sales | 3,625 | 16,000 |
| 2. Mail subscription | 3,395 | 10,185 |
| C. Total paid circulation | 7,020 | 26,185 |
| D. Free distribution by mail, carrier, or other means, samples, complimentary, and other free copies | 71,939 | 70,660 |
| E. Total distribution | 78,959 | 96,845 |
| F. Copies not distributed | | |
| 1. Office use, left over, unaccounted, spoiled after printing | 2,135 | 1,082 |
| 2. Return from news agents | 28 | — |
| G. Total | 81,122 | 97,927 |

11. I certify that the statements made by me above are correct and complete.
Margot Comstock Tommervik, Editor

# SOFTALK CLASSIFIED ADVERTISING

# SOFTALK CLASSIFIED ADVERTISING



*Softalk*'s classified advertising section offers a considerably less expensive way than display advertising to reach tens of thousands of Apple owners.

Classified advertising space is available at the rate of $10 per line for the first ten lines, with a five line minimum. Each line over ten lines is $25 per line. Ad copy should be received no later than the 10th of the second month prior to the cover date of the issue in which you want the ad to appear. Payment must accompany ad copy.

The publisher reserves the right to reject any advertising that he feels is not in keeping with the publication's standards.

Heads will be set in 10 point bold face, all capitals only. Italics are available for body text only; please underline the portions you would like italicized.

The body text of the ad will hold roughly 45 characters per line. Spaces between words are counted as one character. Heads will hold roughly 24 characters per line, with spaces between words counted as one character. Please indicate if you would like the head centered or run into the text.

Please write or call for additional information.

Softalk Classified Advertising
11160 McCormick
Box 60
North Hollywood, California 91603
Attention: Linda McGuire
213-980-5074


# Read the best magazine about home computer games available today.

It's tough out there in the gaming world. Fast fingers and good vision aren't enough. It takes brains to survive.

*Softline* can't give you more brains, but it can show you behind-the-scenes action with regular interviews of top game programmers and previews of soon-to-be-released products.

Each issue of *Softline* features reviews of the newest and hottest arcade-style, adventure, and strategy games for home computers.

Game graphics and programming tutorials and a column on education are regular features in each issue.

In the past *Softline* has had in-depth stories on war games, the unique movie *Tron*, the phenomenal game *Wizardry*, and the blossoming genre of mystery games for home computers.

But by far the most popular regular feature in *Softline* is High Scores. Before every issue we receive thousands of submissions from all over the country. The competition is fierce and the stakes are high. Beat the current high score on your favorite game and your name and score will be published in the next issue.

All this and more for a special low price of $9 a year. That's six issues of the best home computer game magazine available today. Try *Softline*. There's nothing else like it.

From the publishers of *Softalk*.

Send order with payment to:
Softline
11160 McCormick Street
Box 60
North Hollywood, CA 91603
213-980-5074

Holiday Special

$9 for a one-year subscription (six issues)
After January 1, 1983, one year subscription price is $12.
And, when you subscribe, let us know what computer you use.

After January 1, 1983, one-year subscription price is $12.

# THE BASIC Solution

By Wm. V. R. Smith

One of the most versatile features of the Apple is the hi-res display. Yet it wasn't until a bit more than a year ago that the Apple's hi-res capability was used in commercial software to simulate seventy columns of text. The machine language character generators in some of these programs have reached such a state of sophistication that the simulated text acts exactly as you would expect real text to behave. Once again, we will set out to prove that what the professionals do in machine language, we can do in Basic, just more slowly. This month's program allows you to print in seventy columns on the hi-res screen.

Why seventy? With hardware modifications, you can get eighty columns. Well, the number wasn't arrived at arbitrarily. The hi-res screen can display 280 points of light, called *pixels*, from left to right. Two hundred and eighty divided by seventy is four. This means that at seventy columns we have a width of four pixels in which to display each character. Because we have to leave a space between characters so they won't run together, this leaves us with characters three pixels wide. You just can't make them any smaller.

The routine is designed for efficiency and adaptability. It should be easy to adapt for use with your own programs; that's why it is structured as it is. The section at the end, starting with line 10000, is the first part to be run. It merely sets up an array, C$(X), and stores the data for creating the characters in it. This routine is at the end because it's used only once.

Each character is three pixels wide and five pixels high. Another way to put that—one that better describes how the actual plotting in this program works—is that each character is composed of five rows of three pixels each. Since each pixel can be only black or white, that means that there are eight possibilities for each row, as shown in figure 1.

```
· · ·  = 0
· · *  = 1
· * ·  = 2
· * *  = 3
* · ·  = 4
* · *  = 5
* * ·  = 6
* * *  = 7
· – black dot
* – white dot
```

Figure 1.

Now perhaps you can see how these eight possible rows can be combined to create a character. Figure 2 shows how the letter A is created.

```
· * ·  = 2
* · *  = 5
* * *  = 7
* · *  = 5
* · *  = 5
```

Figure 2.

Each row of three pixels is a three digit binary number; the digits to the right in the figures are the decimal equivalents. Put five rows of decimal equivalents together and you get a code from which the program can reconstruct a character. An A is created from the code 25755. That's how all the numbers in the data statements at the end of the program were determined. Each represents one character. You should be able to modify this character set to your own special uses; for instance, foreign alphabets, lower case, or graphics characters.

The program works in steps that break the job down to its component parts and perform them sequentially. Lines 1000 through 1020 accept input from the user, put it into O$, then send it to the output routine. These lines are separated from the others because if you use the routine from your own programs you won't need this input section.

Lines 400 through 510 take the string in O$ and break it into individual characters. The program then translates each character into an ASCII code, puts the value into the variable C, and sends it to the routine starting at line 300.

The routine in lines 300 through 390 determines where a character goes on the screen, looks up the character in the array C$(X), and determines from the code how each of the five rows should look. Then it puts the number code for each row into PL and sends it to the row plotter in line 100.

Lines 100 through 260 plot a single row of three dots based on the number in the variable PL. Though this is the last step in the process, it appears first in the program because it's used most often. This may seem to confuse the issue, but it speeds things up a bit.

A possible use for printing simulated text characters on the hi-res screen is to label hi-res artwork or graphs. Whatever your use, you just have to put the string into O$ and *gosub* to line 400. If you want to change the way the program determines where on the screen to put it, the variables X1 and Y1 in lines 310 and 340 determine the screen coordinates. If you don't want the routine to clear the rest of the line (this feature can destroy parts of anything you might be labeling), the section that does that begins at line 470.

Happy computing!

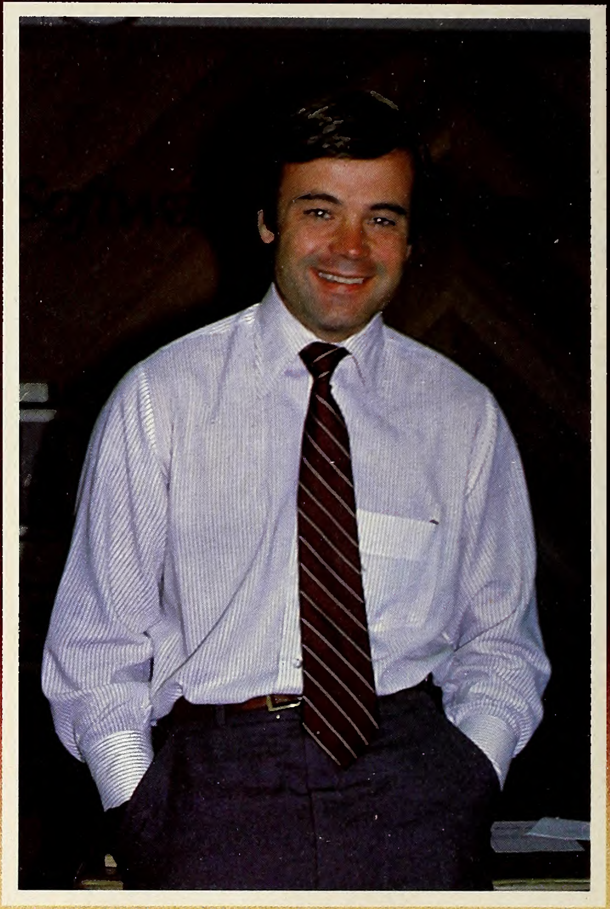
### A Seventy Column Output Routine

```
10 GOTO 10000
100 REM ** PLOTTER
110 ON PL GOTO
    130,150,170,190,210,230,250
120 HCOLOR= 0: HPLOT X1,Y1: HPLOT X1
    + 1,Y1: HPLOT X1 + 2,Y1: RETURN
130 HCOLOR= CO: HPLOT X1 + 2,Y1
140 HCOLOR= 0: HPLOT X1,Y1: HPLOT X1
    + 1,Y1: RETURN
150 HCOLOR= CO: HPLOT X1 + 1,Y1
160 HCOLOR= 0: HPLOT X1,Y1: HPLOT X1
    + 2,Y1: RETURN
```

```
170 HCOLOR= CO: HPLOT X1 +1,Y1:
    HPLOT X1 + 2,Y1
180 HCOLOR= 0: HPLOT X1,Y1: RETURN
190 HCOLOR= CO: HPLOT X1,Y1
200 HCOLOR= 0: HPLOT X1 + 1,Y1:
    HPLOT X1 + 2,Y1: RETURN
210 HCOLOR= CO: HPLOT X1,Y1: HPLOT
    X1 + 2,Y1
220 HCOLOR= 0: HPLOT X1 + 1,Y1:
    RETURN
230 HCOLOR= CO: HPLOT X1,Y1: HPLOT
    X1 + 1,Y1
240 HCOLOR= 0: HPLOT X1 + 2,Y1:
    RETURN
250 HCOLOR= CO: HPLOT X1,Y1: HPLOT
    X1 + 1,Y1: HPLOT X1 + 2,Y1
260 RETURN
300 REM ** OUTPUT A CHARACTER
310 X1 = ((VX - 1) * 4) + 1
320 FOR X = 1 TO 5
330 PL = VAL ( MID$ (C$(C),X,1))
340 Y1 = ((VY - 1) * 6) + 1 + X
350 GOSUB 100: NEXT X
360 VX = VX + 1: IF VX < = 70 THEN 390
370 VX = 1:VY = VY + 1: IF VY < = 26
    THEN 390
380 VY = 1
390 RETURN
400 REM ** GOSUB HERE TO PRINT
    STRING IN O$
410 OL = LEN (O$)
420 FOR LO = 1 TO OL
430 C = ASC ( MID$ (O$,LO,1)) - 32
440 IF C < 1 OR C > 61 THEN GOSUB
    360: GOTO 460
450 GOSUB 300
460 NEXT LO
470 REM ** CLEAR REST OF LINE
480 HCOLOR = 0
490 FOR X = 0 TO 4: HPLOT X1 + 3,Y1 + X
    - 4 TO 279,Y1 + X - 4:NEXT
500 VX = 70: GOSUB 360
510 RETURN
1000 REM ** START OF ROUTINE
1010 INPUT O$: IF O$ = "" THEN GOSUB
     500
1020 GOSUB 400: GOTO 1000
10000 REM ** INITIALIZATION
10010 DIM C$(61): FOR X = 1 TO 61
10020 READ C$(X)
10030 NEXT X
10040 VX = 1:VY = 1
10050 HGR :CO = 3
10060 GOTO 1000
10070 REM ** DATA
10080 REM ** CHARACTERS ! THROUGH]
10090 DATA
      22202,55000,25052,27672,51245
10100 DATA
      24257,22000,12221,21112,27225
10110 DATA
      02720,00212,00700,00033,11244
10120 DATA
      75557,22222,71747,71717,55711
10130 DATA
      74717,44757,71111,75757,75711
10140 DATA
      22022,22024,12421,07070,42124
10150 DATA 71202,35742
10160 DATA
      25755,65656,34443,65556,74747
10170 DATA
      74744,74757,55755,72227,11157
10180 DATA
      56465,44447,77555,57775,25552
10190 DATA
      75744,25577,75655,34216,72222
10200 DATA
      55557,55552,55577,55225,55711
10210 DATA 71247,74447,44211,71117
```

The arcade-warp is open!
YOU CAN RUN BUT CAN'T HIDE. YOU CAN KILL BUT CAN'T ESCAPE.
Tubeway! It's an insidious invasion route created by beings from a parallel universe—a strange, geometric universe. You're trapped on the rim as their fleet swarms out of the warp on a voyage of conquest. The battle is yours alone—and it's far from easy because normal strategy doesn't work! . . . you have to fight by their strange, geometric rules!
Here's the fastest, most fascinating of the new style space games. So involving and exciting it's destined to become an all-star, all-time hit. Be one of the first to take on the challenge of the lightfast Tubeway!
$34.95 for the Apple II*. At your computer store or:
TUBEWAY
DATAMOST INC
9748 Cozycroft Ave., Chatsworth, CA 91311
(213) 709-1202
VISA/MASTERCARD accepted. $2.00 shipping/handling charge. (California residents add 6½% sales tax.)
pple II is a trademark of Apple Computer, Inc.
COREY WOLFE

# Exec Software Publishing:



## In the Front File, Personally

BY JONATHAN MILLER

About two months after Software Publishing Corporation launched its first product, general manager Janelle Bedke received a disturbing call. It was from a dealer who'd bought *PFS*—the *Personal Filing System* program—and apocalyptically pronounced it *too simple.* He was typical of the breed at the time, a disguised hobbyist who delighted in intricacies. To him, a program wasn't a program unless it had all those bells and whistles. Yet a week after his first call, the guy was back on the blower, only this time with a grudging admission.

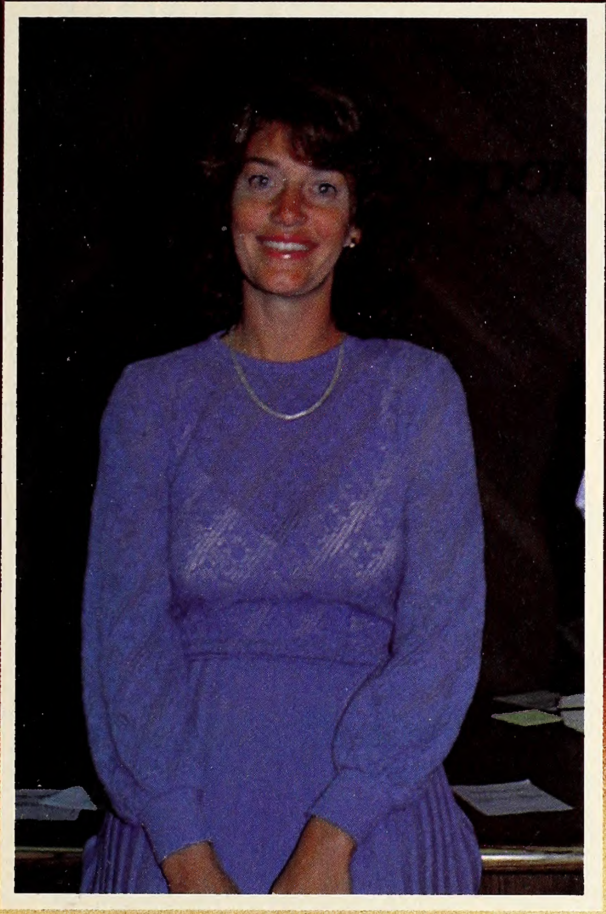
"It's deceptively simple," the dealer conceded. "When I got in and used it, it was exactly what I wanted."

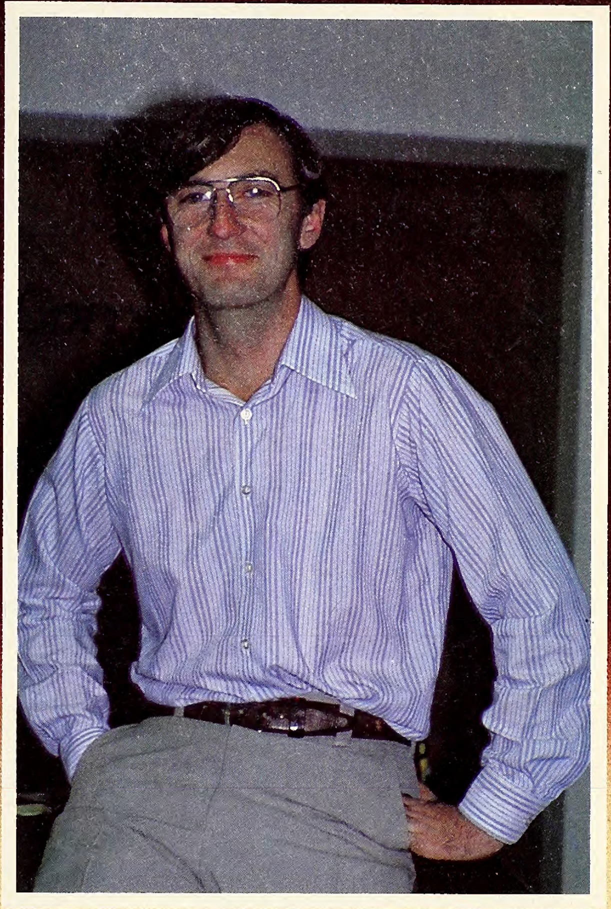
Says Bedke of the first call, offering a story postscript with a "more is less" moral, "I knew he wasn't right. I was running our entire company on that program. Order forms, checking accounts, equipment purchases. Most programmers then were proud to have more features. We were proud to have less."

The dealer's about-face made one of Bedke's twelve-hour days in November 1980, when she was Software's only full-time employee. She was directing operations single-handedly from the proverbial Silicon Valley garage, while partners Fred Gibbons and John D. Page held on to jobs at Hewlett-Packard. Years of software experience had gone into designing the company's deceptively simple product, and the effort was already paying off. Orders, which were a respectable garage-size 70 the first month, jumped to 159 by October. Then, in November, Bedke—manual writer, boxes stuffer, and all-purpose gopher—received orders for 500 more *PFS* packages. Software had a winner, all right, and Bedke couldn't help but savor the delicious frustration. She'd temporarily run out of product.

Since Software's auspicious launching, in September 1980, the corporation has been writing its own Silicon Valley success saga. One million in sales (and an unheard of profit) the first year in business, four million the second—the recession notwithstanding—and a projected nine million in fiscal 1983. The company has been so successful that venture capitalists were courting them six months after launching. With their product a hit and their more-is-less philosophy vindicated, the trio of Bedke, Gibbons, and Page was at the crossroads. To expand or not to expand, that was the question. Recalls Gibbons, "We just sat around one day and looked at each other and said, 'Now what do we do?' "

What they could have done was go with the flow. They could have paid themselves comfortable salaries and waited in leisure for John Page to write his next program. Or, as wind-surfer and pilot Gibbons phrases it, they could go for all the marbles. They could take the plunge—arranging for a $250,000 transfusion from Jack Melchor Venture Management of Los Altos and hiring their first employees.





"We felt that the timing was positive," Gibbons explains. "We had the ideas, we had the product, and we just wanted to be in the marketplace."

Software wants to be in the market in a big way. They're working to become a household word among first-time computer users looking for a simple package of office and professional programs priced around $100. To judge by the second-year numbers, the trio is gaining on that management objective. A cumulative total of one hundred thousand units sold; three programs—*PFS, PFS:Report,* and *PFS:Graph*—in *Softalk*'s top ten business management systems; and plans to introduce a fourth *PFS,* an easy-to-use word processor, sometime in the first half of 1983. Staffing is scheduled nearly to double, from thirty-nine to sixty-five, by next September. Additional plans call for some fine-tuning of the advertising, greater penetration of European markets (now 10 percent of SPC's sales), and, perhaps most important, the release of *PFS* programs compatible with two other popular machines, the IBM Personal Computer and the TRS-80.

Gibbons takes pains to stress that the expansion to other computers implies no disrespect to Apple. "We have a tremendous loyalty to Apple; they're still the leader," he says. It's just that a company, especially one with a growing number of shareholders, is obliged to leverage its investments. Which is Gibbons's diplomatic way of saying you don't want to put all your eggs in one basket, not in a market as volatile as computers.

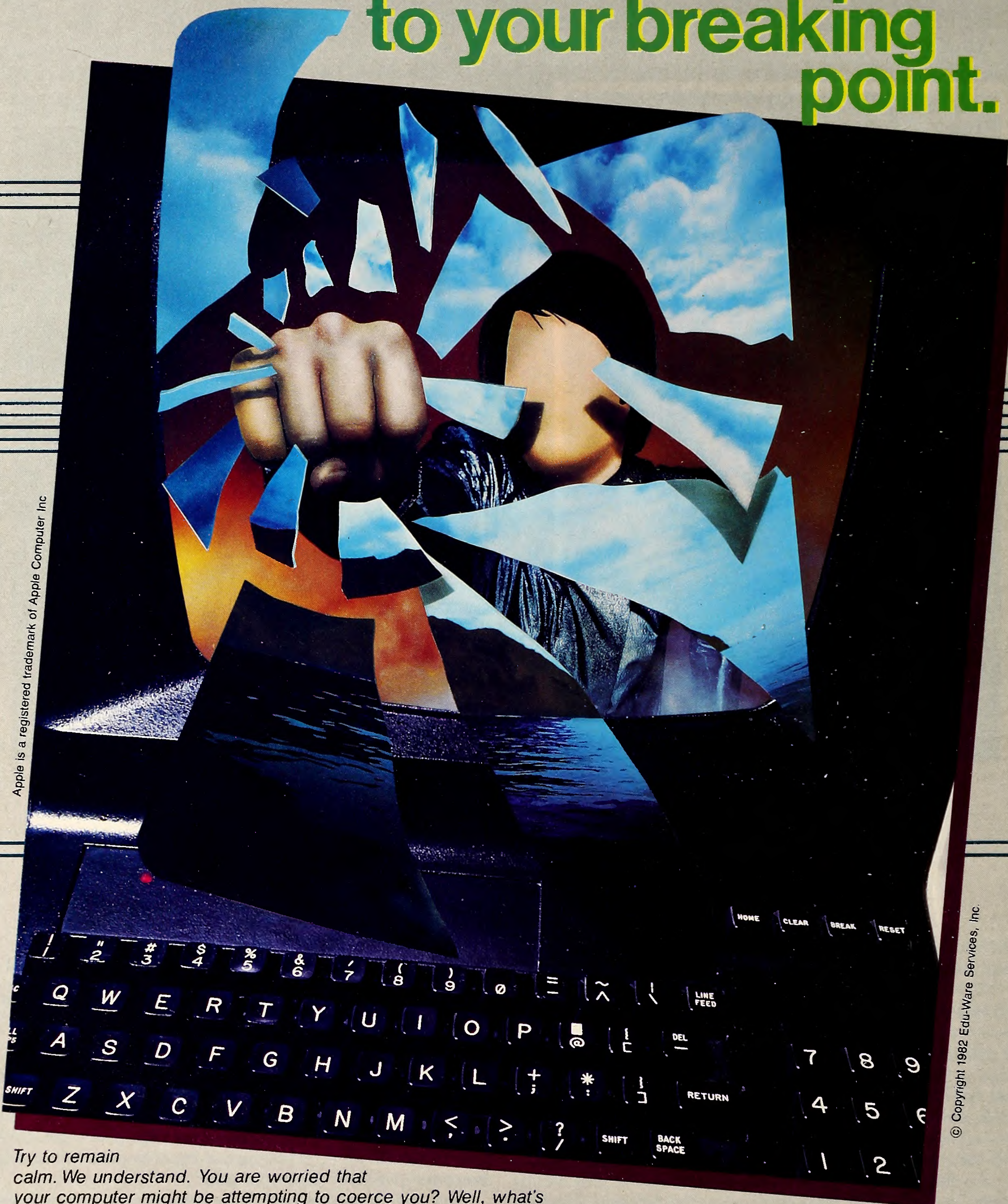
The management trio from Software Publishing Corporation has reason to smile. They've parlayed a good idea for a packaged series of easy-to-use office management programs into a $4 million-a-year business. The principals in this Silicon Valley success saga are, from left to right, Fred Gibbons, president; Janelle Bedke, general manager; and programming guru John D. Page, software development manager.

"You make 90 percent of your investment doing the first implementation of a program," programmer John Page points out. "With the other 10 percent of the effort you can put it on two or three more machines and double or triple your sales."

This shift in emphasis will be reflected in Software's new ad campaign. The sell is softer, even tutorial; the approach conceptual rather than product-oriented. The campaign targets Software's principal market, typically the inexperienced user and self-employed professional. The intent, says SPC, is to reach the potential purchaser a step earlier in the buying cycle with ads that zero in on why he needs a computer as much as why he needs *PFS.*

"Two years ago, when we started, this was a technologically knowledgeable marketplace," explains Gibbons. "Today, the market is much less computer literate. We want to take a leadership role in helping educate these people."

# Welcome to your breaking point.



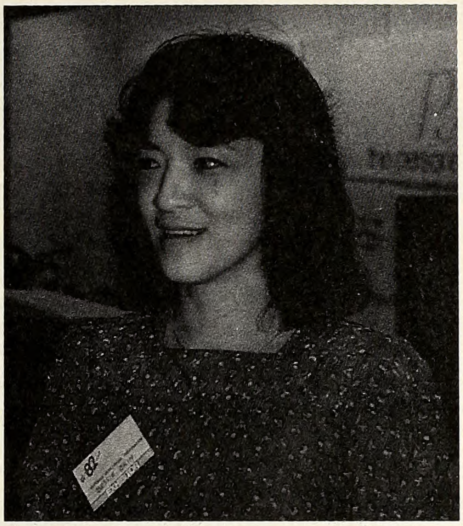
*Try to remain calm. We understand. You are worried that your computer might be attempting to coerce you? Well, what's a little coercion among friends?*

*As a new arrival on The Island, you will be expected to divulge certain information. You will not leave until we obtain it . . . and we will. Of course, when we gain, you lose.*

*Please endeavor to maintain the proper perspective. A pleasant adventure can so easily lead to obsession . . .    Applesoft, 48K, DOS 3.3 . . . . $32.95 Available at computer stores on finer Islands everywhere. Exclusively from if™ interactive fantasies, a division of Edu-Ware Services. P.O. Box 22222 • Agoura, CA 91301*

**Prisoner 2** *A science fiction nightmare by David Mullich.*

™ interactive fantasies



The task of putting a charting program on the charts fell to Bessie Chin, co-author and project manager of research and development for *PFS:Graph*.

**Keeping It Simple.** To appeal to that market, according to the Software trio, you have to demystify the computer. When the powerful *WordStar* system hit the retail outlets a couple of years ago, the typical untutored purchaser blamed himself, not the program, when he found it too complicated to use. Today, the buyer blames the system—and rightly so in Page's view.

"The hardest thing to do in programming is to know what to leave out. A programmer's path of least resistance has always been creeping elegance and it has to be stamped out. It's easy to figure what to put in. You can think of an infuriating number of little gizmos and options that have some conceivable use, but pretty soon the program's incomprehensible and loses all consistency."

Software's commitment to consistency and simplicity is rooted in the varied experiences of its management team. They've been able to build a better program family, they say, because they understand all aspects of the computer business—from the hardware to the software to the marketing. In the case of Gibbons, a Harvard MBA, resident cheerleader, and founding father, the Software idea took hold when he was developing vertical business packages for Hewlett-Packard minicomputers in 1979.

"The work was interesting enough," Gibbons recalls. "We wrote the programs, but the difficulty was that we had to train people on what is accounting. It was very support and service intensive."

Down the hall from him at this time was John Page. He was having similar misgivings about the state of the software art. A transferee from Hewlett-Packard's London, England, division, he was plugging away designing software for data management systems on the H-P 3000 minicomputer yet felt it wasn't really getting the job done as it should. For one thing, Hewlett-Packard couldn't recruit enough people into data processing who knew enough about computers.

"Computers had stagnated in the early 1960s in their complexity of use, but they were proliferating all over the place. People in data processing departments didn't know what was going on with these machines. And then when Fred came along, it sort of crystalized my ideas about designing easy-to-use software."

It had been Gibbons's initial plan to get outside authors to develop the product with SPC publishing programs for a royalty—hence the Software Publishing name. But his encounter with Page suggested another approach, and he found himself musing aloud about how great it would be if someone did a filing system for the self-employed professional. That thought ultimately became *PFS*, the first in the packaged series of integrated products designed, engineered, manufactured, and marketed by Software.

To design for an Apple you first had to have one, so Gibbons went to the source. "I talked with Steve Jobs (cofounder of Apple). He was encouraging and arranged to give us equipment at a store discount." For the next eight months, Page labored by day and created by night, fine-tuning the *PFS* program. Shortly after launching, Gibbons realized SPC couldn't be run on the fly. He needed an experienced hand at the helm and it so happened that Bedke, another Hewlett-Packard alumnus, was

Keeping it simple: left to right, software engineers Lori Cameron, Ed Mitchell, Brad Crain, Steve Hill, Dave Blair, and project manager Anne Michon.

available. After eleven years at H-P, she had decided in November of 1979 that she was ready for a new career direction. Fred made her an offer the Godfather might have refused, but it was just the opportunity she was looking for.

"When I called her up, I asked how would you like to do this for stock in the company and no pay," says Gibbons, laughing. "And she says, 'But what do I have to do?' 'Everything,' I said. 'Write the manual, stuff the boxes, negotiate with Apple, be the gopher, and work with and pacify John.' "

Bedke worked closely with Page because she wrote the manual as he developed the program. That collaboration has become a metaphor for what the company perceives as the key to its success: a few ideas well thought out, to borrow the cadences of the U.S. Marines ad.

"I remember when we were doing the manual," Bedke recalls. "I'd go back to John and say, 'I can't write this, it's too complicated. I can't write it so I can read it.' And he'd say, 'Yeah, maybe you're right,' and alter the program." Adds Page, "If you can't explain it, it's not right. We take the position that the manual is as important as the program itself."

**The Acid Test.** One of the ways he tests a program feature, says Page, is to explain it to somebody else. If he can't do that in a few choice words, can't do it without rendering his hominoid guinea pig senseless, it's not that he explained it badly, just that he designed it wrong.

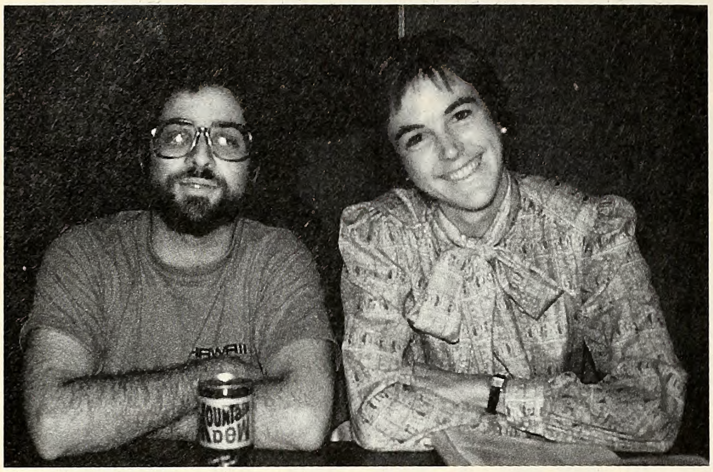
Creative give-and-take stamps all interdisciplinary relations at Software, says the management trio. The operative premise is simple enough. In mass product design, a few good heads are better than one. So it follows as no surprise that Software products are developed by four-member teams consisting of a program director, manual writer, marketing person, and manufacturing manager.

"We don't tell people how to do their jobs," says Gibbons. "After all, we hire them to be creative. What we do expect is that those who sign on be committed to our management objective—developing easy-to-use products for the low end of the package software market."

The bottom line, of course, is quality control. When you design, manufacture, and market your own product, you necessarily have more of it. According to Gibbons, Software is the only company at its level of success that engineers its own products.

"VisiCorp didn't, Microsoft didn't, MicroPro didn't, and Digital Research didn't," he says excitedly, "so we have been the only company that took an integrated engineering and marketing approach right from the beginning." Software further enhances its product control through in-house manufacture—assembling the disk, manual, and boxes right on the premises, in its airy new office complex (sixty-eight hundred square feet, upstairs and down) on Landings Drive in Mountain View, California.

Such organization, unique for a small company, characterizes the trio's professional approach to management. The implied message: We're steady and dependable, just like those nearby Moffet Field pilots who bank their prop-engine P3 sub watchers right over Landings Drive,



Software engineer Jeff Tucker with applications market manager for the professionals, Leslie Larson.

regular as clockwork. The right stuff and all that. They keep their cool, in short, which is why they managed to avoid the common mistake of releasing a program too soon. They stuck to their basic plan, which, to summon the Marines once again, has been to produce a few good programs. That has meant devoting more time to planning and to regular product-team meetings and resisting temptations to jump into other lines. After their success of the first year, SPC almost jumped into educational and entertainment software but finally pulled back. They decided against it, says Gibbons, for the simple reason that the company wasn't ready.

"It's like what John says about programming; it's what you leave out."

Being courted by venture capitalists could easily turn a wide-eyed programmer's head, but the trio from Software, to borrow one of Gibbons's favorite words, stayed focused. "We approached that like we would the picking of a partner," says Gibbons. Software considered three offers, finally deciding on the Melchor group for two powerful reasons—their offering price on Software stock and their management experience should the trio seek their advice (which they gladly do). In point of fact, SPC has actually raised two rounds of venture capital—the quarter million in April 1981 and another million a year later. Underwriting the second round was the Melchor Group, New Enterprise Associates of San Francisco, Crown Associates of New York, and the investment banking firm of L. F. Rothschild, Unterberg, and Towbin. The SPC partners deliberately sought second-round financing from other geographic regions, explains Bedke, because the company distributes nationally and hopes one day—she can't say when—to go public.

In any event, if a man be judged by the company he keeps and a businessman by his investors, Software is in very good company indeed. Owning a piece of their rock are some of the valley's brighter luminaries—Bob Noyce, president of Intel; Les Hogan of Fairchild Industries; and Ken Oshman of Rolm Corporation. "We're the star in their portfolios," Gibbons gloats. While these investors are obviously looking for a good return, he says, he feels their support is inspired by a larger motive. What these investors really want, says Gibbons, is to see the computer industry grow, and what more promising field of green could there be than an up-and-coming company in the burgeoning software industry? "They want to be among the people planting seeds back in the ground and continuing the technological revolution."

**Job Satisfaction.** On a day-to-day basis, it's doubtful that Bedke, Gibbons, and Page see themselves as technological crusaders, but it's evident that the trio is very pleased with their accomplishments—as individuals and as a company. His commitment to writing the first *PFS* program, for example, put John Page straight into what he wanted from his work.

"My heart was in building stuff. I like creating things, particularly things that turn out to be useful for a lot of people. When you can build a company the way you think it should be run, build a product the way you think it should be built, and then see them get the kind of acceptance that we have . . . well, it's kind of addictive. It's a ball, really. The highs you get out of it far outweigh the problems."

The problems, such as they are, have been little ones, the normal pains that come with growth. Like seemingly endless conferences and personnel interviews, getting carpet for the atrium stairway, forcing yourself to take a Hawaiian or Virgin Island vacation (as they did), and dealing with the Great Unknown. The latter refers in part to a squadron of high-flying bees that no one has ever seen but that has been leaving droppings on cars in the parking lot. In a huge corporation like Hewlett-Packard such matters are handled by unseen people in personnel and building maintenance, but when you run your own show, the buck begins and ends with you. There are times, Page reflects, when it seems like everything's going to fly apart, that they're losing control. But then, he adds, that uncertainty forms an integral part of the fun and excitement that is building your own company. "It's a ball, really."

And what of Software's future in the brave new computer world, a world that will see Macy's and Sears peddling product along with Computerland? "Excellent," says Gibbons. Software Publishing Corporation is ideally positioned to grow with the mass consumer market, he believes.

"We're a safe buy. We don't let the customer fail and I think the dealers are getting that. If the dealer has a problem, we'll step right up there."

Three years ago, when software products were in short supply, shelf space wasn't a problem, contends Gibbons. A new product could win credibility and generate enough cash to underwrite an enhancement cycle. Today, the individual author is competing against firms like Software, VisiCorp, and Microsoft, all of which are producing integrated program families. They're building a name as well as a product.

Ultimately, Software would like to assume a leadership position among this select group.

"I find that we're in the lucky situation of having found our niche," observes Gibbons. "Sometimes those niches close up because the giant comes in and squashes you, but in our industry there are no giants. Nobody's going to squash us." ▪

# IT'S TIME KIDS STARTED USING STRONG LANGUAGE.

We encourage it.

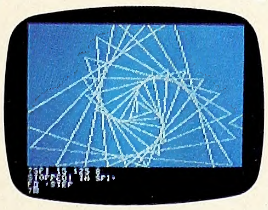
Because now the most powerful educational language is available on the Apple Personal Computer.

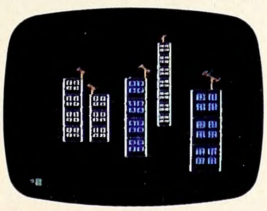
Presenting Apple Logo.

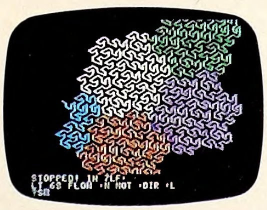
It's not just a programming language for computers, but a learning language for people.

Enough so that anyone, working with Apple Logo, can easily learn the programming principles once reserved for college courses.

Apple Logo encourages you to break problems into small steps, and then shows you how to make those steps automatic.







It does all this interactively.

For instance, if you accidentally type "foreword," instead of forward, Apple Logo responds with "I don't know how to foreword."

There is no such thing as a mistake with Apple Logo, only logical statements telling you what needs to be done to make the program work. So the student programs the computer. Not the computer the student.

And as you learn, Apple Logo learns with you. So whether you're a student of 5 or 55, you'll always be challenged—but not overwhelmed.

Apple Logo runs on the Apple II with 64K. And it comes from Apple, the leading personal computer company in education—with the largest library of courseware at all levels.

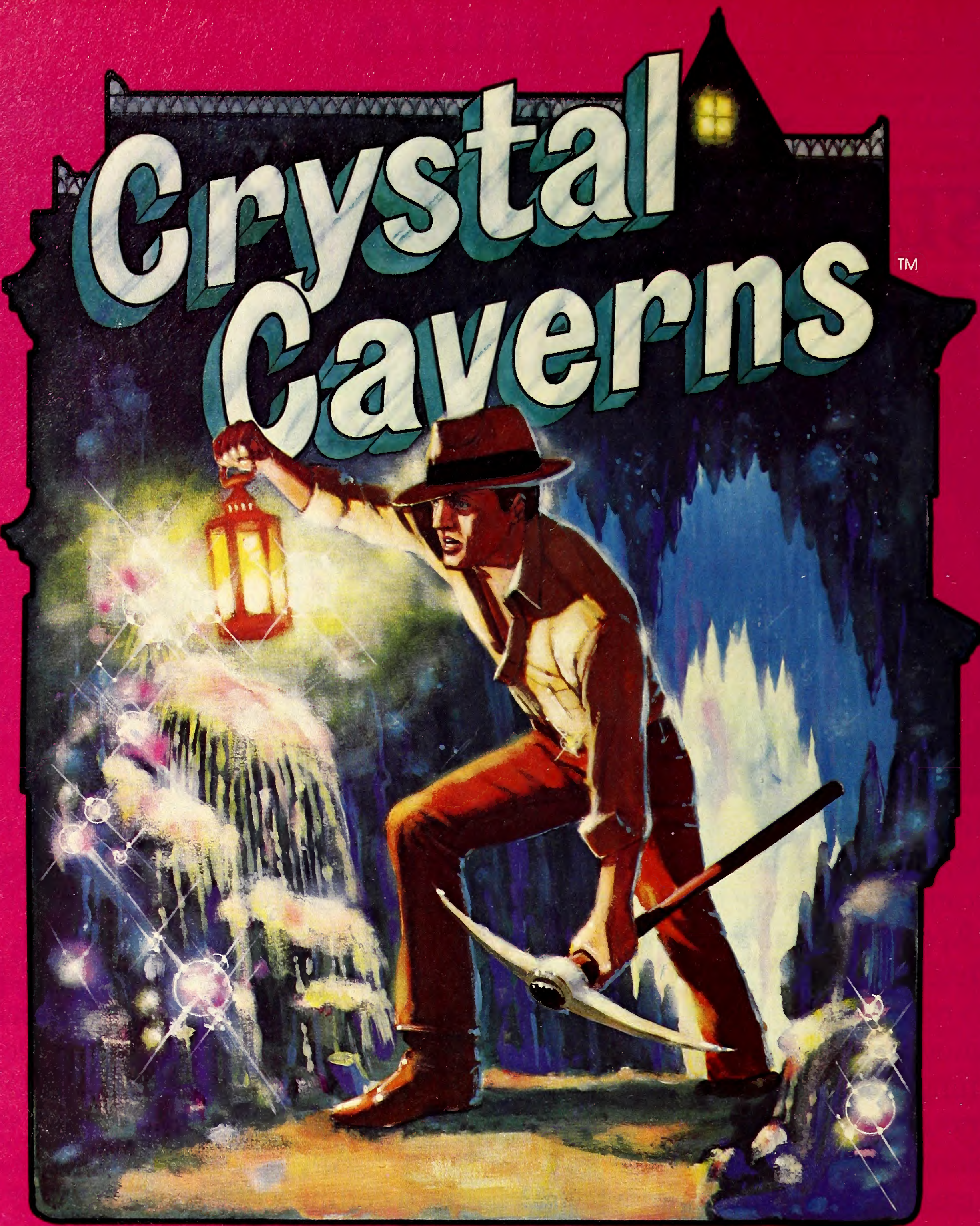
Apple Logo. It can make getting to know a computer the most positive of learning experiences.

Your kids will swear by it.

The personal computer.

apple®

For more information, call (800) 538-9696. In California, call (800) 662-9238. Or write: Apple Computer Inc., 20525 Mariani Avenue, Cupertino, CA 95014.
Apple® Logo is a product of Logo Computer Systems, Inc., 222 Brunswick Boulevard, Point-Claire, Quebec, Canada H9R1A6.




## The treasure hunt that tests your wit, your imagination–and your nerve!

Somewhere beneath the crumbling mansion lie gold, rubies, a rare and priceless painting . . . treasure enough to start your blood pounding! But what mysteries—and dangers—wait for you? Your guide will assist you in your quest: you can use multiple commands and full sentences. But this is no place for the faint of heart or the slow of wit. If you're ready to be tested as never before, you won't want to be anywhere else but deep within the CRYSTAL CAVERNS!

Available from your local dealer, or call:

**1-800-343-1218**

(in MA, call 617-937-0200)

**Crystal Caverns: #19409**

**Apple II disk, $34.95**

HAYDEN **SOFTWARE**


# Bring her back alive–if you can!

This case has more twists than a Martini drinkers' convention. You're Al Club, private eye, and you've got to grab a beautiful heiress from the gang that snatched her. (Well, maybe she isn't beautiful. But her money is.) Along the way there's a murder or two, and plenty of clues—if you know where to look. Miss one and who knows? One of those murders might be your own! You've got until midnight—but you can save precious moments with a few tricks of your own, like multiple commands and full sentences. CRIME STOPPER will keep you running 'till the wee hours.

Available from your local dealer, or call:
**1-800-343-1218**
(in MA, call 617-937-0200)
**Crime Stopper: #19509**
**Apple II disk, $34.95**

HAYDEN **SOFTWARE**

# NEWSPEAK

☐ **Aftermath.** Two months after the Us Festival, the dust has finally settled in Devore, California. The helicopters and RVs have gone away. The freeway is wide-open and free.

Not to be overly dramatic, the spirit of the festival lives on and is spreading around the world. Soon the Us Network will be operational and the spirit will travel farther and faster.

Just what is the Us Network? During the Labor Day festival it was a two-way video linkup with the Soviet Union. Festivalgoers saw a traveloguelike short about Moscow one night and live scenes from a Moscow radio station the next night. The latter consisted of three hundred Soviets gathered on a sound stage dancing to the music of Soviet group Stas.

The video feed from the Us Festival was featured one evening on the Soviet Union's only major television station, Bremja. Generally a news and educational medium, Bremja features three minutes of cultural offerings every night. The broadcast from the Us Festival was given the whole three minutes and was seen by approximately 150 million Soviets.

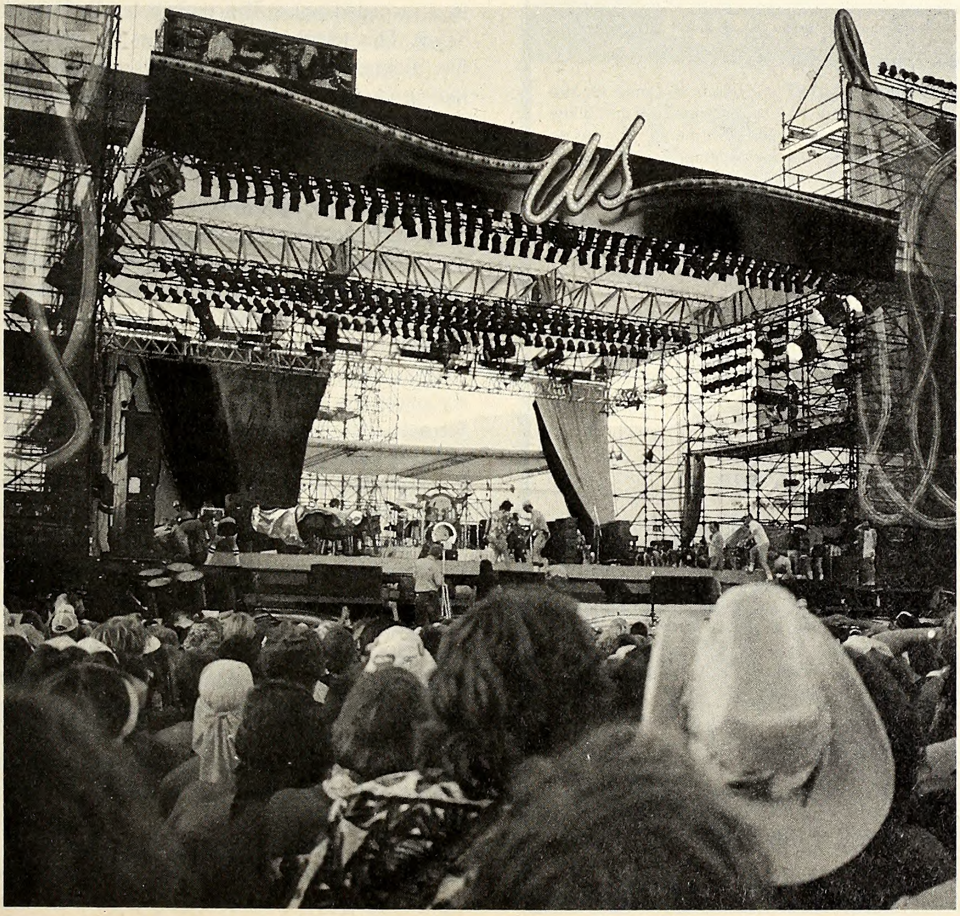
"It was a big event in the Soviet Union. Unheard of," said a Unuson spokesperson. "This was the first demonstration of the Us Network's capabilities for interactive communication."

The full scope of Wozniak and Unuson's intentions with the Us Network is staggering. In the years to come, universities, colleges, high schools, and special computer centers around the world are projected to be linked together with the help of computers and space-age technology.

Using a Westar satellite, educational programming, future Us Festival concerts, lectures, and conferences could be downloaded to many different locations. At first the system will consist mainly of straight video networking and will be targeted at universities and colleges.

A likely scenario runs something like this: Bigtown State contracts out a network system, with the initial cost falling on Unuson. Bigtown is expected to pay back its debt with profits earned from charging admission to broadcasts of future Us Festivals. The system would also be used for sending and receiving educational programming and other services, like news.

Currently, Unuson is talking to a dozen uni-



versities on how best to set up this kind of system. Once a firm video network is established, personal computers will enter the scene to make the system more interactive. Even individual homes may someday become part of the Us Network.

The idea is to create a broader sense of community and be able to share human resources better than before. It's the feeling of Wozniak and Unuson that science and art together can help unite people and improve our worldly situation.

One additional advantage will be not always having to rely on the telephone system for long-distance interactive communication. The use of satellite technology should greatly enhance the viability of the Us Network.

There may be another Us Festival at the same location in Devore as early as Easter 1983. Regardless, the Us Network should be around for a long time and benefit an ever increasing number of people.

□ **Touch-Tone Therapist.** At the Veterans Administration Medical Center in Birmingham, Alabama, the burden of follow-up care for the speech, language, and hearing impaired has been lightened a little bit—electronically. Since therapy clinicians can't make house calls, it was necessary to invent a computerized version of one that did. Its name is Remate (remote machine-assisted treatment and evaluation), the brainchild of Dr. Gwenyth Vaughn, audiologist and speech pathologist at the medical center.

What Remate does is call patients via Touch-Tone phone. It asks the patients if they are ready in a recorded voice. If a patient is, the lessons begin. This telephonic therapy (as Dr. Vaughn has nicknamed it) doesn't replace traditional face-to-face drill work with a clinician; it's designed to complement it.

Remate can provide drill work in several areas.

In speech therapy, words and phrases are repeated by the patient and then played back for him to hear, using the patient's own voice.

An attachment called the Telewriter permits handwriting tests, and the handwriting is modem-translated for analysis.

For stroke therapy, a patient is asked four to six questions with a measured response time. Remate can recognize whether the answer is right or wrong.

Traveling for long-distance therapy can be expensive and exhausting—both physically and emotionally. Three factors figure into a veteran getting quality therapeutic follow-up care: parking, weather, and travel, according to Dr. Vaughn.

"The average visit is 175 miles round trip per patient, and most of them try and come for daily care. The cost to travel averages forty-two cents a mile these days. Remate can save a patient about five thousand dollars a year and provide him with care as good as face to face. In fact, it's identical."

And it's vastly less expensive for the medical center as well without sacrificing the quality of the therapy. "We can save five million dollars for every thousand patients over a six-month period. That's for two sessions a week of drill work per patient."

The Remate system is a combination of micros and mainframes that work together to make Touch-Tone therapy seem easy. Hardware configurations were designed by Dr. Thomas Donald, who set up the "black box" housing the line interfaces connecting the various machines. John Kramer, computer specialist with the medical center, is the man who keeps it all up and running.

"The medical center has twelve Motorola 68000 micros that do all the talking, typing, writing, and dialing," Kramer explains. "Each one can handle twelve phone lines at a time. They're not set up yet for automatic dialing—the software hasn't been written—so the calls are made manually at the moment.

"Four Motorolas are each connected to three PT 11/22s over high-speed channels. They have operating systems that can handle multiple tasks. These machines hold the programs currently being used.

"The central computer is a PDP 1144. It handles all the evaluation and research as well as the scheduling. It has a catalog of about thirty programs, or lessons, for clinicians to choose from. Remate is only in its infancy—the PDP is set up to hold a thousand different lessons."

The prize piece of equipment in the Remate system is the TEL-plus, a Touch-Tone telephone with a device that lights up when buttons are pushed next to pictures or words. This is especially useful when treating stroke victims with aphasia, a condition characterized by understanding and comprehension within the brain but no discernible way to communicate it.

For the last three years, Dr. Vaughn and her crew of computer specialists have been wrestling with getting Remate set up. "We kept having to get things approved, approved, and approved," she explains. Funding was handled by the Veterans Administration Rehabilitation Research Department, set up to support research, do reviews, and award projects. "We just got an award to go another three years," she notes.

"We're going to study the telephonic system versus face-to-face care. This is called efficacy testing. It's a crucial undertaking, since no actual data exists at the moment. We're confident about Remate here at the medical center, but we're going to have to prove the system really works."

To date, some eight thousand patients have benefited from clinician-assisted telephonic therapy. Remate can handle many more. As Dr. Vaughn says, "We're aiming for a nationwide VA service network for speech and hearing impaired persons. Remate could be enlarged at a very small cost." Right now, several Veterans Hospitals across the country are set up to call in and schedule their own local therapy using Remate.

"One person can monitor all the input right now," Kramer explains. "But it's hectic in a crunch, I'll tell you." And he's the one who would know—but it looks like Remate can handle the load.

☐ **Customized Taxes.** A controversial bill that exempts custom computer programs from a 6 percent state sales tax has been signed into law by California Governor Edmund G. Brown, Jr.

Industry sources say the legislation, the first of its kind nationally, could have pace-setting implications given California's traditional leadership role in technology and lawmaking. The measure, which was pushed by a fledgling software group that felt it was being victimized by a tax-strapped State Board of Equalization, upholds the industry's longstanding view that customized programs are one-time services which are not subject to the sales tax.

The California bill only applies to customized computer programs that are developed primarily for mainframes. It does not affect mass-produced, commercial software programs sold for personal computers. The latter will continue to be taxed as tangible personal property, as they are in most states.

The new measure was lobbied through the California legislature by a coalition of companies under the banner of the Software Services Association. Paul Connolly, a member of the group and the vice president and general manager of Informatics of Palo Alto, said they had sought legislative relief as a defense against the policies of the State Board of Equalization. In the past year, he says, the board slapped scores of software developers—some of whom weren't even aware of the policy's existence—with assessments for unpaid sales taxes going back from three to eight years. One firm received a back-tax bill of $285,000, he says.

The association appealed to the legislature, according to Connolly, because there was no effective administrative remedy. Assessment appeals are not heard by an independent review authority, he says, but by the board itself. State law does permit a company to sue the board in civil court, Connolly adds, but only after they've gone through the appeal process *and* paid the disputed taxes. By that time, many companies could have wound up out of business.

The bill became a hot political issue this past summer when a member of the board labeled the legislation a hundred million dollar special interest tax break for "some of the most affluent members of our society." Industry sources ridiculed the figures of William M. Bennett, claiming he was confusing sales with taxes. Connolly says it's doubtful that annual statewide sales of custom programs total $100 million, and only a portion of that would be subject to tax anyway under the board's interpretation of the regulations. "This is not a tax break," he declares. "Sales taxes are paid by the customers. All the companies do is collect the monies and pass them on."

One provision of the bill does provide a special, one-time tax consideration, however. J. D. Dotson, head of the board's business tax department, characterized as "very unusual" a condition refunding $1.2 million in previously paid sales taxes to custom program purchasers. The rationale behind the refund, Dotson explains, was that the taxes should never have been levied in the first place, because custom-designed programs are a service not subject to taxation.

Dotson, whose department furnished Bennett with his $100 million tax-loss figures, still believes that the board and his department were correct in their interpretation of the regulations. "Any manufactured article has a lot of labor and service involved," he says, "but the ultimate thing that the customer uses is a piece of tangible personal property—a machine-readable program."

According to Dotson, his department had been following this interpretation since 1975 when the board adopted it as official policy. During that time it had been enforced "equally and fairly," in line with the availability of audit staffs. In the last couple of years, however, at least according to Connolly, the enforcement has become more aggressive. Did this suggest, as some in the industry have speculated, that a money-strapped state government was casting about for additional revenue sources? Not by Dotson's lights. "There's no truth to the charge that we did it to raise money."

And now, as if to mock the notion, they have to give some of it back.

☐ **May the Source Be with You.** A portrait of the average videotex user emerges from the 1982 subscribers survey taken by the Source. The results indicate that the average user of the electronic information and communication service is forty-two, three years older than last year's average.

Eileen Friend, marketing research director for the Reader's Digest Association, owners of the Source, says that the data gathered by the survey suggests "a more serious approach to videotex services" and rebuts the theory that personal computing is only a hobby for young people.

"Twenty percent of subscribers now use the Source most often from their place of work. Fifty-three percent say they spend at least half their time on the Source for either their primary or their secondary occupations."

Subscribers this year tended to form networks with each other, and twice as many as the year before contacted the Source on a daily basis. The highest occupational percentage was among professionals (doctors, lawyers, CPAs).

The eight-page questionnaire was distributed to four thousand randomly selected subscribers on May 16 and achieved a 39.9 percent response rate before the June 20 stop date.

☐ **Crossing Borders.** Shycon Associates has been in the business of computer logistical models for twenty years. Their latest, Logistek-Europe, is a financial model of the European continent as a unified system.

The model figures transportation costs between countries, warehousing costs, inventory costs, currency exchange rates, raw material imports, transfer pricing, taxes, and other factors; then sets up a what-if scenario to allow companies to try different strategies for satisfying market requirements at the lowest total cost.

Company president Harvey Shycon refers to the model as a "hybrid structural simulator/optimizer. It attacks a previously impenetrable area of corporate fat: inefficient manufacture and distribution operations caused by national boundaries." The end result is the elimination of guesswork from inventory, distribution, and transportation decisions and identification of different levels of customer service for the creation of a specific customized business strategy.

☐ **The Bank Shot.** The banking business is feeling the impact of the personal computer revolution—seven years before it thought it would. According to the report *Consumer Electronic Banking,* just issued by International Resource Development, a specialized management consulting and market research firm, the personal computer has made home banking a reality far in advance of the predictions of retail banking experts.

"Small and medium-size banks have been cozily assuming that bank-at-home wouldn't become really important until videotex systems became widespread—perhaps around 1990," says IRD study project manager Carl Frankel. "Now they see 1983 as the year in which they may lose all their remaining profitable customers, having already lost many big earners to Merrill Lynch in 1981 and 1982." Merrill Lynch, along with American Express and other "financial supermarkets," now offers various telebanking services to the consumer that compete with traditional bank offerings.

The past two years have seen an overwhelmingly positive response from consumers to the conveniences of paying bills, switching deposits, and performing other fiscal chores in the privacy of their own homes. "The average upper-income banking customer," says Frankel, "hates standing in line."

The manufacturers of specialty terminals like Micro Peripherals's Bank Box or Chase Manhattan's Telecredit Paymatic are likely to feel the effects most acutely as their market continues to erode before the advance of the home computer. Of the three million personal computers that will be in place by the end of 1983, about two million of them will be in the hands of individual consumers, says IRD. Frankel describes the specialty terminal manufacturers as "sucking their teeth" as they watch the personal computer wave engulf many of their projected markets.

Additional direct impact of home computers on banking in the near future will result in fewer tellers and smaller buildings, which will result in turn in a major reconfiguration of traditional bank investment policy. "As early as 1987," Frankel predicts, "the banking industry may become a net seller, rather than a net buyer, of top retail exposure real estate."

Having grasped the fact that the families buying home computers are the same families that have long provided the deposit base for commercial banking, Chemical Bank now has a personal computer telebanking service, "and soon Citi, Chase, BankAmerica, and other biggies will be sweeping into the market." ☐


That's right — anything can happen, and it usually does! Because with a Scott Adams Graphic Adventure, the fantastic is as close as your computer — and your own fertile imagination. Each S.A.G.A. features the finest in hi-resolution graphics — graphics which compliment and enhance the classic text — and, they can be toggled on or off with a single keystroke, too. Plus, optional Votrax Type 'N Talk™ and printer compatibility give you the flexibility to add new dimensions to your Adventuring — when you're ready.

The incomparable S.A.G.A. Series — experience the magic yourself. But be prepared for anything.

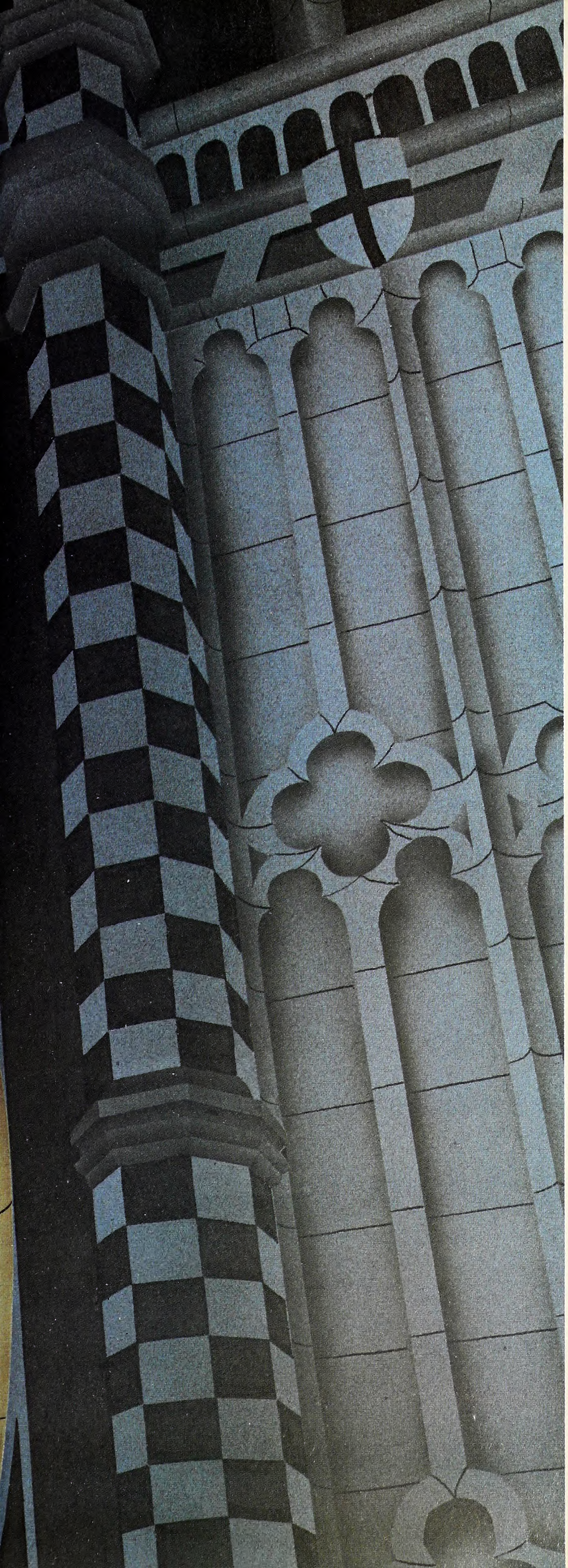
| | | |
|---|---|---|
| **S.A.G.A. #1 — Adventureland** (Skill Level: Moderate) | | |
| APPLE 2 PLUS 48K Disk (DOS 3.3 req.) | 042-0201 | $39.95 |
| ATARI 48K Disk | 052-0201 | $39.95 |
| **S.A.G.A. #2 — Pirate Adventure** (Skill Level: Beginner) | | |
| APPLE 2 PLUS 48K Disk (DOS 3.3 req.) | 042-0202 | $39.95 |
| ATARI 48K Disk | 052-0202 | $39.95 |
| **S.A.G.A. #3 — Mission Impossible** (Skill Level: Advanced) | | |
| APPLE 2 PLUS 48K Disk (DOS 3.3 req.) | 042-0203 | $39.95 |
| ATARI 48K Disk | 052-0203 | $39.95 |
| **S.A.G.A. #6 — Strange Odyssey** (Skill Level: Moderate) | | |
| APPLE 2 PLUS 48K Disk (DOS 3.3 req.) | 042-0206 | $39.95 |
| ATARI 48K Disk | 052-0206 | $39.95 |

PRICES SUBJECT TO CHANGE

Also, look for the Scott Adams Text Adventure Series on these fine computers:
TRS-80 • COMMODORE VIC • NEC PC-8001
TEXAS INSTRUMENTS 99/4 • CP/M Z-80

Adventure INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC.

To order, see your local dealer. If he does not have the program, then call 1-800-327-7172 (orders only please) or write for our free catalog.
Published by ADVENTURE INTERNATIONAL
a subsidiary of Scott Adams, Inc.
BOX 3435 • LONGWOOD, FL 32750 • (305) 830-8194





# It's III O'Clock and All's Well

## BY JOHN JEPPSON

*Hickory Dickory Dock*
*Where is the Apple III clock?*
*The program is done,*
*But the chip won't run,*
*Hickory Dickory Dock, Tum Tum.*

Considering that Apple III made its first public appearance less than two years ago, it's amazing how far the machine's development has come. The 96K Apple III is gone; the 256K machine is common. A second-generation operating system, SOS 1.1, has been distributed to all users, and even that has been further upgraded. Extensive revisions of Utilities, Business Basic, and the Apple II Emulator are in the hands of every registered owner. And *all* early models of Apple III have been recalled and exchanged to eliminate the disastrous problem of chips working loose in their sockets (an exchange program possibly unique in modern manufacturing, not just computers). In fact, Apple III is definitely "up and running" with major new programs and applications appearing all the time. Just one dangling, unfulfilled promise remains: the Apple III clock.

**Ticks and Bugs.** The great clock flap began with the earliest advertisements announcing Apple III. The machine was to have its own internal real-time clock with battery back-up. The exact time and date would be available instantly to all programs, and the machine would automatically stamp each file with the date and time of creation and modification. A really neat feature, and a big selling point whose appeal in our time-pressured society far exceeds any practical value that might actually be realized. After all, you don't really need the clock, but it is . . . neat.

Then—disaster. After months of waiting, the first Apple IIIs were shipped. They crashed, repeatedly, with "System Failure" plastered across the screen. The chips wouldn't stay in their sockets. Immediately Apple began exchanging defective machines for new ones, no questions asked. But word of the problems spread quickly. And to top it all off, the clock didn't work. And wouldn't work. Nothing could be done. The available clock chips were no good and there was no other source. One can imagine the frustration and rage at Apple Computer. Geologists doubtless believed Cupertino was having earthquakes. Finally Apple simply withdrew the clock from its specifications. The clock was no longer part of Apple III, and if you paid for one they sent you a fifty dollar rebate.

So it was over, right? The clock was gone, right? Not on your life. The clock is gone only from the ads. At position B3 on the motherboard of every Apple III is an empty chip holder neatly labeled "58167." And over on the other edge is a nice round hole labeled "batt." The circuits are all there. And the access routines are also there, in SOS, the Sophisticated Operating System. When, eventually, you get the clock and the battery, you can just plug them in and you'll be in business. The clock is far from dead. It's merely in suspended animation, awaiting more fertile times.

By a strange coincidence National Semiconductor recently announced the MM58167A real-time clock chip for less than fifteen dollars in May 1982. If you buy one of these chips and stick it in that empty chip holder (orient it the same way as its neighbors) it works just fine.

For intrepid tinkerers, the battery hole on the motherboard is "pos" on the bottom and is marked "−" on top. Three penlight batteries in a holder from Radio Shack will give you 4.5 volts,which is just about right. By placing the battery holder up front next to the speaker you can get to it easily should the batteries ever need to be replaced. You should then connect a couple of longish wires to the battery site on the motherboard. Apple warns that major permanent changes to the computer may jeopardize the warranty, so we don't recommend that you solder these wires. It


PUT ONE OVER ON APPLE II.
Introducing Station II.
The Apple II Support System.
There's more to Station II than meets the eye.
It's the one accessory that secures, protects and organizes your Apple® system. It can even keep your Apple cool. Station II pulls your computer, peripherals, cords & cables together into an attractive, easy-to-use, integrated system.
The key is a power switch.
• Key controls 3 built-in power outlets.
• One turn of the key powers your Apple, monitor and another peripheral.
• The switch takes the wear and tear of rebooting, saving you the cost of replacing your Apple's power supply.
Hours of computing comfort.
• You can adjust the distance from keyboard to monitor.
• Monitor is angled for maximum viewing ease.
Station II is a perfect fit for Monitor III.
The key is security.
• With Station II bolted down, you can lock your Apple inside. Secures Apple and accessory boards against theft.
• You can lock your system "on" or "off" and control who uses it.
It can cool your Apple.
• An optional fan installs internally on molded-in fan mounts. Industrial-quality, quiet-running.
• Air baffles efficiently direct air flow through your Apple, extending computer life.
A surge suppressor protects your Apple and programs.
• Intercepts voltage spikes.
• Attenuates high-frequency noise.
There really is more to Station II than meets the eye. So put one over on your Apple II today.
One cord powers your entire system.
trace
Trace Systems, Inc. 1928 Old Middlefield Way
Mountain View, CA 94043 (415) 964-3115
®Apple is a registered trademark of Apple Computer, Inc.

is rumored that this is the approach Apple will take when it finally makes an "insert" package available.

**A Variety of Clocks.** Just so we are all talking about the same thing, there are a number of different clocks associated with Apple III. So far we have been discussing the functioning clock chip MM58167A, but there are several others. We shall try to keep them straight as we go along. One such clock, for example, is the 6502 CPU clock, which, in Apple III, runs at around 1.8 MHz. This clock governs operation of the 6502 processor. It is very important to the computer, but it has nothing to do with keeping track of dates and times-of-day. We shall not refer to it again.

Another clock is the "system time and date." If you have an Apple III without a functioning clock chip (as all Apple IIIs are currently delivered), you still have a "system clock." But it isn't really a clock at all, merely a set of numbers in memory. You may set these numbers to any time and date you wish, and that becomes the official time and date for the operating system. These values are then used to mark the time on files and are accessible to programs through the system's access routines. If you don't have the MM58167A chip, then these remembered numbers are the system clock.

**Of Time and Space.** The system time and date values are held in a very special part of memory. Apple III's memory is much too large for all of it to be handled directly by the 6502 CPU chip. Instead, various 32K banks of memory are switched in and out, two at a time, so that the 6502 always has exactly 64K bytes of memory to play with. Any other quantity would make it very unhappy. One bank, the "s" (system) bank, is almost always on-line. It holds SOS, which monopolizes memory locations BC00 to FFFF. We are talking now about RAM memory, which is volatile and is lost when the power is turned off. Every time you boot the machine, SOS is reloaded into this section of RAM from the file SOS.Kernel on the boot disk. Whatever was previously stored in that RAM is overwritten and lost. You start again with a clean slate.

But there is one small area of the system bank that is normally not RAM at all. This is a thirty-two byte stretch from FFD0 to FFEF. Apple III has two *6522 Versatile Interface Adapter* chips, or VIAs for short. These are "helper" chips that handle intake and output and are deeply involved in the complexities of bank switching by which Apple III manipulates its huge memory. Each VIA is allocated sixteen memory locations for its access ports. So when the 6502 CPU addresses locations FFD0 to FFEF it gets these VIA ports. The section of ordinary RAM memory that otherwise would lie in this area is normally switched off-line and can be reached only by a special technique known as 8F extended addressing. It turns out that this special area of RAM, being off-line, is not overwritten during a boot. In that sense it is protected, and it is, in fact, the only area of Apple III's memory that can be counted on to retain information through a reboot. So that is where they put the system clock. Once set, the system clock will retain its values through successive (control-reset) bootstraps until the values are deliberately changed or the power is cut.

**As Time Goes By.** When you insert the MM58167A chip, the system clock is instantly transformed into a real, ticking clock. Now, when you set the clock, you set both the functioning clock chip and the time and date values in protected RAM. The time and date values serve as a back-up for the clock chip. They hold the last known valid time should the clock chip later be found defective.

SOS provides two access procedures for communicating with the system clock. Both are SOS calls. They are named *set time* and *get time*. SOS calls are machine language commands that execute various subroutines in the operating system. There are some thirty-five different SOS calls and they perform a wide variety of complicated tasks. You should think of them as new 6502 op-codes. SOS calls may be used directly from any assembly language functions or procedures you may write. High-level languages, such as Basic and Pascal, also define commands for talking to the system clock. But when these high-level commands are executed, their code paths must always flow through the common pathway of these two SOS calls.

The *set time* SOS call accepts information in the form of an eighteen byte sequence of ASCII digits that contains the time and date information. *Set time* always attempts to set the hardware clock, the MM58167A chip, whether it exists or not. And it also stores the information in protected RAM as just discussed.

*Get time* first attempts to read the clock chip. If there is no operational clock chip then *get time* retrieves the last valid time and date from its storage area in protected RAM. And finally, if there is no valid time and date in RAM, then *get time* returns a string of eighteen ASCII zeros.

Since all languages and application programs always access the system clock through these two SOS calls, it follows that all programs written in those languages will continue to work in the same fashion whether there is a functioning clock chip or not. If there is a functioning chip, you get a current time and date. If not, you get the time and date values stored in RAM. But you don't have to make any changes in your programs when you install the chip. You just start receiving more up-to-the-minute information. This is an excellent example of Apple III's ability to cope with upgrades of both hardware and software while continuing to support programs already in use.

**Using the System Clock.** Some application programs such as *Apple Writer III* have special commands with which you may set the system clock (that is, the functioning chip and/or the time and date memory values). The computer then uses the current values to time stamp files whenever you save your work to disk. This can be a great help in keeping track of files. The time stamp is displayed with the file name whenever you catalog a disk.

Programs written in Apple III Basic can read the time and date values very easily. Although it doesn't say so in the current Basic manual, Date$ and Time$ are reserved variables. Date$ returns the date in the format yy/mm/dd, and Time$ returns the time as hh:mm:ss. These are read-only variables. If you attempt to assign a new value to one of these variables you get zapped with a syntax error. Reading the system clock from Basic is easy, but setting it is considerably more complicated.

One way to set the system clock from Basic is with the Basic program *Timeset* which comes on the Business Basic disk. *Timeset* displays the current clock values and accepts changes. Any new values are then transferred as parameters to a subroutine in the Timeset.Inv invokable module. Invokable modules are machine language procedures, and this one is particularly simple. It pops the specified values off the stack and stores them in an output buffer. Then it issues a *set time* SOS call and the job is done.

The program *Timeset* is a separate, independent, Basic program. It works very nicely but you cannot (easily) use it from the middle of some other program. Oh yes, it can be done, in a roundabout way, using exec files or by chaining from one program to another. But that's messy. The intention seems to be that you will use *Timeset* to set the system clock when you first turn on the machine and then read those values as needed from any programs you may subsequently run.

**Planned Obsolescence in the Year 2000.** If you want to set the system clock from your own Basic programs you can invoke the Timeset.Inv module yourself. Unfortunately there is no available documentation for using the module except what you can ferret out by listing and studying the *Timeset* program. Briefly, what is required is a set of integers, each of which specifies one digit of the time stamp. In this case eleven digits must be specified. They are sent to the invokable module as parameters of the external procedure Timset (no "e"). The digits are yy mm dd w hh mm, which in your program you might store as the eleven integer variables Y1%, Y2%, M1%, M2%, D1%, D2%, D3%, H1%, H2%, MI1%, and MI2%. SOS always sets the century value to 19. You don't get to change that. Also, this invokable module will not accept values for the seconds. They will be assigned 00.

Notice that w is stored in D3%. W is a value between 1 and 7 that represents the day of the week, where Sunday is 1. The *Timeset* Basic program actually calculates this value from the year, month, and day using a formula known as Zeller's Congruence. But you really don't need to do this. Just assign any value in the appropriate range whether it's correct or not. Something must be sent, but the particular value chosen is ignored by the subsequent *set time* SOS call. The SOS routines prefer to calculate the correct value for themselves.

When you have input or otherwise assigned all these integers, you send them to the Timeset.Inv module, and thence to SOS, by the command:

```
100  PERFORM TIMSET (%Y1% * 256 + Y2%, %M1% * 256 + M2%,
     %D1% * 256 + D2%, %D3%, %H1% * 256 + H2%, %MI1% * 256
     + MI2%)
```

**Giving Pascal the Time of Day.** Pascal has two separate schemes for setting and reading the system clock. One is for use in programs; the other is a command in the Pascal Filer. The Filer date command is activated by pressing D. It displays the current time and date and accepts changes. The Filer date command is unique in that it not only sets the clock chip (if there is one) and the time stamp in protected RAM, it also stores the date in the file System.Miscinfo on the Pascal System Disk. This disk time stamp is then available should you later boot Pascal and the system is unable to find either a functioning clock or a valid time and date in protected RAM. Only the date is stored on disk; not the time. The date is reduced to two bytes by an ingenious coding system and is stored at bytes 002E and 002F of System.Miscinfo. Expressed as hexadecimal nibbles, these two bytes are DMYY where *M := $month; D := $day MOD $10*; and *YY := ($year * 2) + ($day DIV $10)*. In other words, the year value is doubled and then summed with the carry-from day. If the result is an odd number, you know that *D := D + $10*.

From within Pascal programs all communication with the system clock is performed by four procedures in the Applestuff Library Unit. They are Date, Timeofday, Clockinfo, and Settime. The Date and Timeofday procedures each take a string variable parameter. They each return an ASCII string, yyyymmdd or hhmmss, respectively. Clockinfo takes eight integer parameters and returns integer values for year, mon, day, dayofwk, hr, min, sec, and thou. Finally, to set the system clock from a Pascal program, there is Settime (T), where T is a string variable of eighteen ASCII characters. The string must be in the format yyyymmddwhhnnssuuu, which is exactly the same format used by the *set time* SOS call. This last procedure, Settime (T), is similar to the Timeset.Inv invokable module used in Basic. It accepts the parameters, does some range checking, and issues a *set time* SOS call. Once again, the w value and the uuu millisecond values must be sent but need not be accurate. The SOS call ignores them.

**Even More Clocks!** If you are not content with the system clock you can buy several others. One popular external clock is the Thunderclock, made by Thunderware, Inc. This is a completely independent clock on a peripheral card. It has its own battery back-up and fits in one of the expansion slots in Apple III. The Thunderclock is actually designed for Apple II but will also work in Apple III if you know how to set it up. Unfortunately, the Apple III documentation that is available is a bit sketchy.

Whenever you buy a new device for Apple III you must not only plug in the device, you must also install the device driver that comes with it. Apple dealerships distribute a disk containing the necessary driver for using Thunderclock in Apple III. A driver is a program. It is written especially for one particular device, and it acts as an intermediary between SOS and that device. SOS sends the driver a read or a write request and the driver then translates that request into the special requirements of that one device for which the driver was written.

A new driver must always be placed within the collection of other drivers on the boot disk. The entire collection is kept in the file SOS.Driver. You must use the *System Configuration Program* on the Utilities disk to add your new driver to SOS.Driver. Complete instructions begin on page 5 of the *Standard Device Drivers Manual*. It is very important, when installing the Thunderclock driver, to specify a slot assignment. This tells the driver where you have put the Thunderclock card. You also have an opportunity at this time to review and perhaps change the device name. Every device (driver) has a name. It is essentially a file name except that it begins with a period. The Thunderclock driver is generally called *.tclock*, but with the *System Configuration Program* you can change the name to anything you wish. All that really matters is that whatever name you choose, it must be the name your program uses when it wants to talk to the Thunderclock. If your program uses any other name, SOS will be all at loose ends and will zap you with a device not available error.

**The What's-Its-Name Driver.** It is this confusion over device names that has caused much grief and confusion for people trying to make Thunderclock work in an Apple III. The *System Configuration Program* is confusing enough to Apple III beginners, but, to compound the problem, the people who put together the Thunderclock Driver disk made a rather distressing error. They used two different names for the driver.

On the Thunderclock Driver disk is a file called Tclock.Code. This file contains the new driver, and in this version the driver's name is *.tclock*. In this version also, the slot number is unassigned. Tclock.Code is the file you are expected to use when you add the Thunderclock driver to the various SOS.Driver files on your own boot disks. Using the *System Configuration Program*, you read in the old SOS.Driver file from one of your boot disks. Then read in the new driver from Tclock.Code, assign whichever slot holds your Thunderclock, and "Generate a New System." The new SOS.Driver file thus created is then written back onto the boot disk you started with. That disk now holds the Thunderclock driver and, once that altered disk is booted, your programs can access the Thunderclock by means of the name *.tclock*.

So far, so good. But at this point somebody decided that the disk containing the new Thunderclock driver, which Apple would distribute, should also be a boot disk, and further that this boot disk should contain a demo program that could be used to set the Thunderclock. The idea was good. The execution leaves something to be desired.

In order for the demo disk to be a boot disk it must, of course, contain all the usual SOS files required of all boot disks. And it does. It is, in fact, a copy of Business Basic. And, to the SOS.Driver file of this boot disk, they added the new Thunderclock driver. Naturally, in a working boot disk, the slot assignment must actually be assigned. So they assigned slot 2. And that is why the demo boot disk will work if your Thunderclock is actually in slot 2, and otherwise it will not.

**The Reckless Driver.** If your Thunderclock is in one of the other slots you could, of course, fire up the *System Configuration Program* and change the slot assignment in the Thunderclock driver, in the SOS.Driver file, on the demo boot disk. But that's asking quite a lot of a total Apple III novice who has not yet attained the level of familiarity with drivers that you, gentle reader, now possess. Worse yet, if the demo program cannot find the Thunderclock (perhaps because it is not in slot 2), it

# Find out what Apple's got in store for Christmas.


All you have to do is call and ask for the free Apple® Gift Catalog.

It's full of neat gifts for the Applemaniac on your list—gifts you won't find in any store.

From gold Apple jewelry to smoked-plexi diskette holders to Apple jigsaw puzzles to Apple coffee mugs to Apple Cross™ pens to Apple T-shirts and Apple note pads.

All very reasonably priced and very well made.

Of course, to get your Apple gifts before Christmas Day, we must have your orders by December 1. The latest.

So call now, before this brilliant idea slips through the fingers of your mind.

apple®

The most personal computer.

(800) 538-9696. In California (800) 662-9238.

doesn't say so. The program just fails. The error message refers to a bad subscript. What has happened is that the program has tried to read Thunderclock and assign the time stamp to a string variable. If there is no clock, the string variable ends up with meaningless and illegal values. Then the program tries to use these values as subscripts for indexing an array. This works great if the string contains a valid time stamp from Thunderclock. But an illegal value gives a bad subscript error to the program and heartburn to the user. So the slot assignment problem is likely to be a source of grief. One cannot really blame the creators of the disk for requiring slot 2. After all, they did have to assign some slot number, and it might as well be that one. But they could, rather easily, have arranged for a more informative error message should an error occur.

Unfortunately, there's even worse to come. Through some inexplicable lapse of judgment, when the Thunderclock driver was placed in SOS.Driver on the demo disk, they also changed its name to *.clock*. There is no reason whatever for this inconsistency; it is simply an error. Since the machine now expects all references to the Thunderclock to use the name *.clock*, that is also the name used in the demo program. So the demo program will run only on the demo disk. If you have placed *.tclock* driver on some other Basic disk, then the demo program won't run on the second disk. There is a conflict of names.

Since, by Murphy's Law, the worst possible disaster is bound to occur, we have the most unfortunate victims of all. These are the people who have very carefully added *.tclock* driver to the SOS.Driver file on the demo disk. Now there is universal confusion of names and nothing runs except rampant frustration.

If you have successfully threaded your way through these mine fields your Thunderclock will work. Most of the time. There are, however, some rumors of erratic performance. And indeed, upon careful disassembly of the driver code file it is apparent that an error was missed by the writers. In one subroutine, instead of sending to the clock the hexadecimal value $00, the error sends the contents of zero-page location $20. Usually the contents of zero-page location $20 do happen to be $00, but presumably they are not always so. This could well be a source of intermittent and unpredictable failure. But we hasten to add that we have used our Thunderclock with that same driver for many months and have never, so far as we know, encountered any problem with this bug.

Using Thunderclock is somewhat different from using the system clock, and perhaps a bit easier. You just treat it like any other file. From Basic you use the same commands you would use for talking to a text file on disk. First the file must be opened. Information is then transferred by the usual print and input commands. The following program segment reads eleven ASCII digits from Thunderclock and places them in the string variable T$.

```
200  OPEN #1, ".TCLOCK"
210  INPUT #1 ; T$
```

T$ now contains Thunderclock's time stamp in the format mmddwhhnnss. Note that Thunderclock neither stores nor supplies a value for the year.

Setting Thunderclock is equally easy. You put the information in T$, in the same format, and execute:

```
300  PRINT #1 ; T$
```

Of course the set/protect switch on the Thunderclock card must be in the set position or nothing will happen.

**The Good Old Days.** The only snag is that with Thunderclock you are expected to provide an accurate value for w, the day of the week. This calculation requires a figure for the year, which Thunderclock doesn't store. To perform the calculation you can either lift Zeller's Congruence from the Basic program *Timeset*, or you can simply run the date information into the system clock and then read it back out again. SOS will have calculated w for you. Note that in Apple III the Thunderclock value for w ranges from 1 to 7, with Sunday being 1, just as it does in the system clock. This is different from Thunderclock's output in Apple II, where w ranges from 0 to 6, with Sunday being 0.

Pascal also uses standard file commands to talk with Thunderclock. The input file should be of the type "interactive." After appropriate variable declarations execute:

```
reset (clockfile, ".TCLOCK");
readln (clockfile, timestring);
```

The string variable Timestring now contains Thunderclock's time stamp. Setting Thunderclock is done with:

```
writeln (clockfile, timestring);
```

which writes the contents of Timestring back out to Thunderclock.

If you have ever used Thunderclock in Apple II you will find many differences when you use it in Apple III. The Thunderclock has a rather complicated Apple II program stored in a ROM chip right on the Thunderclock card. When you talk to Thunderclock from Apple II you get the full benefit of information processing provided by this private program. So you can do a lot of things. Unfortunately, the ROM chip subroutines are specific for Apple II. They send output data directly to Apple II's input buffer at memory location $0200. There is no such animal in Apple III. So few, if any, of the ROM chip subroutines can be used in Apple III. All the tricky things that the ROM chip is supposed to do must instead be done by the *.tclock* driver.

Well, some of Thunderclock's abilities were put in the driver, and some were not. It's largely a question of economics and deadlines. The Apple III market at that time wasn't nearly as big as it is now, and writing a more complicated driver would have been expensive. There was nobody to pay for it. There were not enough Apple IIIs around to anticipate a huge sales volume, and there was no additional charge for the Apple III driver program when you bought a Thunderclock. Furthermore, a more complicated driver would take longer to write and debug, with resulting delays in distribution to the public. So they opted for a minimal driver. You can read Thunderclock and write to it. And that's about all. There is only one format for the time stamp instead of the variable format you get in Apple II. And you cannot use the BSR/X-10 interface. Nor can you use any of Thunderclock's interrupt facilities. None of these things is supported by the current driver. It is not the fault of Apple III. The limitations are in the driver program. To our certain knowledge, at least two programmers have written drivers for their own private use that fully support Thunderclock's interrupts. So it can be done.

Perhaps the worst problem with this minimal driver is the lack of adequate documentation for its use. The three pages of mimeographed material that are available presuppose considerable familiarity with the Apple III operating system. They are probably incomprehensible to most Apple III novices. And with Apple III novices we must include the Apple dealers. Until very recently, few dealers have possessed sufficient depth of knowledge about Apple III to cope with previously undocumented problems. Unfortunately, the setup problems and driver limitations have generated so many hostile letters that the folks at Thunderware appear to have lost all enthusiasm for another effort. It's a pity.

According to a spokesman for Thunderware, the adaptation of Thunderclock to the Apple III was actually instigated and carried out by Apple. Apple apparently wrote and distributed the driver to assist Apple III owners who were devastated by the missing system clock. The ensuing problems, of course, all descended on the people at Thunderware, who think of themselves as innocent bystanders.

A word about Apple II emulation mode. There is not, hiding within Apple III, a miniature Apple II just waiting to emerge. The machines are totally different. What the emulator does is create, within Apple III, a virtual Apple II that exists only in a shadowland of swirling electrons. So it is absolutely uncanny how exactly the image is drawn. For most purposes the machine becomes an Apple II, right down to the arrangement of the keyboard. But it's not perfect. There are limitations, and one of those limitations is no interrupts. In emulation mode the Thunderclock is back in its natural home. It works perfectly, until you try to use interrupts. The Thunderclock sends its interrupt, but the computer isn't listening. This is not true in Apple III native mode where interrupts from the slots are fully supported, provided you have an appropriately written driver.

If you have a Thunderclock for your Apple III, what can you do with it? Well, you can read it and use the information in any program you may write. It does have a battery back-up and it does keep modestly accurate time. You can use it in any programs that require the time or date, such as a check writer. But perhaps the most important use for the Thun-


O.K. APPLE III
we've got you covered.
Service
File up to 16 mega bytes.
Over 32,000 records per file.
As many fields in a record as you want.
254 characters in each field.
Custom design screen input.
Custom design screen output.
Custom design print output to include any combination of mathematics.
Use all the extended capacity of Apple III with
DATA MANAGER III
The Apple computer people designed tremendous new capabilities into the Apple III, and we produced Data Manager III to make available full usage of these new capacities.
It is the big brother of Data Factory that has been so successful with Apple II. It will handle as many records as the storage media can hold with total flexibility. The search routine will select records by any criteria you wish. It is compatible with Applewriter III. Can be used with floppies or hard disc drive and the first year extended warranty is included at no additional cost. See it at your dealer today.
Another USER/FRIENDLY software package from the people that brought you Data Factory...Invoice Factory...Wall Streeter
Software that prompts you throughout the program.
micro lab
systems that work
312-433-7550
2310 Skokie Valley Road Highland Park, Illinois 60035

# the best of
# st micro.

Instead of just poring over a manual, run it and make certain that dBASE II does what you need done.

Then if you find it isn't right for you, send it back and we'll return your money, no questions asked.

But we know that you'll probably keep it.

Because having dBASE II is like having a black belt in micros.

Call (213) 204-5570 today or drop by your local computer store for the rest of the story.

Ashton-Tate, 9929 Jefferson Blvd., Culver City, CA 90230.

Ashton-Tate

derclock is setting the Apple III system clock. Then you have a proper time stamp on all your files, and that really is nice.

If you have the MM58167A chip in your Apple III, but no battery, then the Thunderclock is still a handy time source each time you power-up your computer. Once properly set, the system clock chip will, of course, go on keeping time as long as power is applied.

On this line of reasoning it would seem that once you have acquired both the MM58167A chip and a battery back-up for Apple III, you would then have no further use for Thunderclock. This may not be entirely true. For one thing, you might reasonably hope that someday you will get a driver that supports interrupts, if you have a particular passion for interrupts. Perhaps you may wish to write one yourself. It's a pleasant challenge of rational proportions.

**Where Does the Time Go?** More important, there is the question of timekeeping accuracy. Neither Thunderclock nor the MM58167A chip is likely to win any awards for surpassing accuracy. The Thunderclock is susceptible to changes in environmental temperature, although this has not been a problem for us. Our Apple III seldom gets turned off! Even so, it is nothing like a quartz watch. We have not yet succeeded in obtaining accuracy specifications for the MM58167A chip; nor do we know what nefarious influences may prey upon it. But we do know that the two MM58167A chips we have tried each lose several seconds a day in Apple III.

It would seem then that you aren't going to get a terribly accurate clock either way. But at least the Thunderclock is adjustable. You can jimmy the rate control occasionally and, with some patience, get it timed fairly well. So you may just prefer to continue setting your MM58167A chip from the Thunderclock, from time to time, rather than soldering that battery into the Apple III motherboard.

The accompanying program is designed to do just that. It sets the system clock from the Thunderclock every time you reboot. It is, however, a strange sort of program. It is a driver. A single-purpose driver called *.update* (unless, of course, you rename it). It works as follows: Using the *System Configuration Program* on the Utilities disk you place Update in the SOS.Driver file of any or all of your boot disks. Any, that is, that also contain *.tclock* driver. When you boot the disk the drivers are read into operating system memory. During boot, SOS sends each driver an initialization command. This gives each driver an opportunity to set up its device, load arrays, and generally do whatever needs to be done before it is ready to face the big world and your programs.

When Update gets its initialization command it *queues* an *event*. This means that it notifies SOS that there is a program segment to be run at the end of boot, before SOS turns control over to the user's programs. So when SOS has finished the boot it checks the Event Queue and finds Update's message. It then branches to the Event Handler which, of course, is another little routine embedded in the Update driver code file. The Event Handler opens the file *.tclock*, reads the Thunderclock, and immediately sends that information to the system clock with a *set time* SOS call. Then it tidies up its open file and exits. The deed is done. SOS turns control over to the interpreter and you are in business. Sounds complicated? Maybe so, but it's quick. The system clock is set in a fraction of a second during an otherwise normal boot, so you never have to think about it. Update driver will work in any boot disk and is completely independent of the language or application being booted.

**System Prerequisites.** Update driver does make some assumptions. It assumes, of course, that you actually have a functioning Thunderclock in your Apple III and also that you have installed the Thunderclock driver (properly configured) in the SOS.Driver file on your boot disk. The Thunderclock driver and Update driver must both be there. Finally, it assumes that the name assigned to your Thunderclock driver is *.tclock*. Update seeks the Thunderclock driver by name, and the only name it knows is *.tclock*. If these conditions are not met, or if any other error occurs, Update simply gives up the attempt and does nothing. An otherwise normal boot still occurs, but the system clock will not have been set.

Since the system clock requires the year, and Thunderclock doesn't have the year, that bit of data is provided by the Update driver. The year is stored in Update's configuration block data. It can be changed, as required, using the *System Configuration Program*. Instructions for editing configuration block data may be found on page 16 of the *Standard Device Drivers Manual*. Only two digits of the year may be changed. SOS always requires that the century be 19.

You don't need to know anything whatever about programming to use the Update driver. In fact, you don't actually do anything. Its function is entirely automatic. All you have to know is how to get it into SOS.Driver using the *System Configuration Program*, and most dealers ought to be able to help you with that. The program is fairly short, particularly if you leave out all the remarks. But you do need Apple III Pascal in order to assemble it, unless you know somebody who could be persuaded to assemble it for you. It shouldn't take long.

```
;*************************************************************
;
;            .UPDATE Driver (for use with Thunderclock)
;
;*************************************************************
;
            .proc update
;
; equates
;
queevent      .equ       191F       ; queevent entry
Breg          .equ       0FFEF      ; bank register
reqcode       .equ       0C0        ; request code
;
; device info block
;
idblock   .word  0000              ; link to next driver
          .word  start             ; entry point
          .byte  07                ; length of name including period
          .ascii ".UPDATE"         '' ; must be 15 characters including
                                      blanks
          .byte  80, 00, 00        ; device, slot, unit #
          .byte  60                ; device type - character device
          .byte  00                ; subtype
          .byte  00                ; filler byte
          .word  0000              ; blocks
          .word  0000              ; manufacturer - unknown
```

```
         .word  0000              ; version number
         .word  01                ; configuration block length = 1
                                    byte
;
year     .byte  82                ; the year as binary coded
                                    decimal
;
; event parameter array - used by driver
;
param    .equ   *                 ; parameter array
         .byte  0FF               ; priority = highest
         .byte  00                ; ID byte
         .word  handler           ; pointer to event handler
pBank    .byte  00                ; handler Bank (to be filled in)
;
pPtr     .word  param             ; pointer to parameter array
;
; entry point for driver
;
start    lda    reqcode           ; load request code
         cmp    #8
         beq    DInit             ; if not DInit then just return
         rts
;
DInit    .equ   *
         lda    Breg
         sta    pBank             ; driver bank to parameter array
         ldx    pPtr              ; parameter array location in X
                                  and Y
         ldy    pPtr+1
         jsr    queevent          ; queue the event
         rts
;
;*****************************************************************
;
;              event handler (embedded in driver)
;
;*****************************************************************
;
handler  .equ   *
;
         jmp    hStart            ; skip over buffers and arrays

;
; pathname string for SOS call "OPEN"
; ----- change this if your Thunderclock driver is not called ".TCLOCK"
;
tclock   .byte  07                ; name length including the
                                    period
         .ascii ".TCLOCK"         ; driver pathname - must be
                                    capitals
;
; time stamp buffer "sysbuf" - 18 bytes long
;        system clock uses all 18 bytes
;        tclock uses 12 bytes beginning at "tcBuf"
; initialize bytes 2-17 to $30 = ASCII 0
;
sysbuf   .word  3931              ; the century = 31 39 = ASCII 19
bufYear  .word  3030              ; year will go here
tcBuf    .block 0B,30             ; time stamp from tclock goes
                                    here
cr       .byte  30                ; this byte will receive $0D
                                  (carriage return)
         .word  3030              ; two more bytes in sysbuf
;
; parameter lists for SOS calls
;
oList    .byte  04                   ;param list for OPEN
         .word  tclock               ;pointer to pathname
refnum   .byte  00                   ;reference number result
         .word  0000
         .byte  00
;
rList    .byte  04                   ;param list for READ
rRefnum  .byte  00                   ;reference number from
                                      OPEN
         .word  tcBuf                ;pointer to time stamp buffer
         .word  000C                 ;transfer 12 bytes
         .word  0000                 ;result = number bytes
                                      transferred
;
tList    .byte  01                   ;param list for SET TIME
         .word  sysbuf               ;pointer to buffer for system
                                      time
;
cList    .byte  01                   ;param list for CLOSE
cRefnum  .byte  00                   ;reference number from
                                      OPEN
;
; handler entry point
;
hStart   .equ   *
;
; load YEAR from: configuration block (2 nibbles of binary coded decimal)
;          into: sysbuf (2 bytes of ASCII digits)
;
         lda    year
         and    #0F               ; mask off high nibble
         ora    #30               ; convert to ASCII
         sta    bufYear+1         ; low digit
         lda    year
         lsr    A                 ; shift high nibble -- >low nibble
         lsr    A
         lsr    A
         lsr    A
         ora    #30               ; convert to ASCII
         sta    bufYear           ; high digit
;
; open .TCLOCK as a file
;
         brk                         ;SOS call
         .byte  0C8                  ;OPEN call number
         .word  oList                ;parameter list for OPEN
         bne    return               ;return immediately on error
;
; read tclock time stamp into buffer
;
         lda    refnum               ;reference number from
                                      OPEN
         sta    rRefnum
         brk                         ;SOS call
         .byte  0CA                  ;READ call number
         .word  rList                ;parameter list for READ
         bne    return               ;return immediately on error
;
; TClock has returned newline character ($0D) as 12th byte of timestamp
; change it to $30 = ASCII 0 (occasionally Pascal objects to $0D here)
;
         lda    #30
         sta    cr
;
; validate all eleven bytes from TClock - if slot empty or not Thunderclock
;        then (probably) not all will be ASCII digits
;
         ldx    #0B
next     lda    tcBuf,x
         cmp    #30
         bcc    return
         cmp    #3A
         bcs    return
         dex
         bpl    next
;
; set system clock
;
         brk                         ;SOS call
         .byte  62                   ;SET TIME call number
         .word  tList                ;parameter list for SET TIME
;
;        (incidental info: GET TIME is identical - call number 63)
;
; close tclock
;
         lda    refnum               ;reference number from
                                      OPEN
         sta    cRefnum
         brk                         ;SOS call
         .byte  0CC                  ;CLOSE call number
         .word  cList                ;parameter list for CLOSE
;
return   rts
;
         .end
```

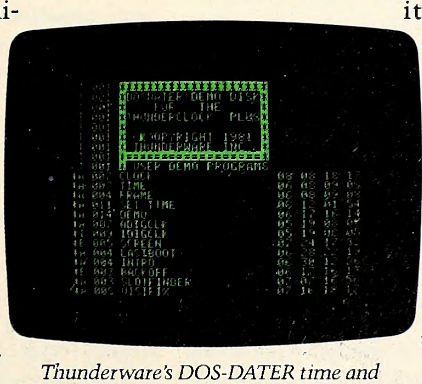
# HAVEN'T YOU HEARD OF THUNDERCLOCK PLUS™?


If you want to put your Apple® to work—around the clock—Thunderclock Plus is the solution. Just plug it in and your programs can read the month, date, day of week and time—down to the second—in any of Apple's languages. So your Apple can do any number of tasks for you automatically. In the office, the lab or at home.

Most good software packages for business, data base management, communications and time management are made to read Thunderclock Plus. (It's compatible with DB Master,* Micro-Courier** and VisiDex†, to name a few). So no matter how you use your Apple now, Thunderclock Plus can make it a more versatile and efficient tool.

For example, with business or communications software, your Apple can automatically access a data base or send electronic mail when the rates are lowest.

In addition, Thunderclock Plus can organize your disk files. Our optional DOS-DATER™ software upgrades the regular DOS on your disks. So every time a program is saved or a file is modified, the time and date, *to the minute,* are stored in the CATALOG with the file name. Now you can instantly know exactly when your files were last updated.



*Thunderware's DOS-DATER time and date stamps your disk files to the minute.*

Thunderclock Plus can even give you a sense of security. Or just make your life a little easier. With our X-10 interface option and a BSR X-10‡ Home Control System, your Apple can turn on your lights, water your lawn...whatever you desire, according to schedules you create. It comes with our menu-driven SCHEDULER software. So it's easy to design and modify schedules that can run in the "background" while you have "hands-on" use of your Apple.

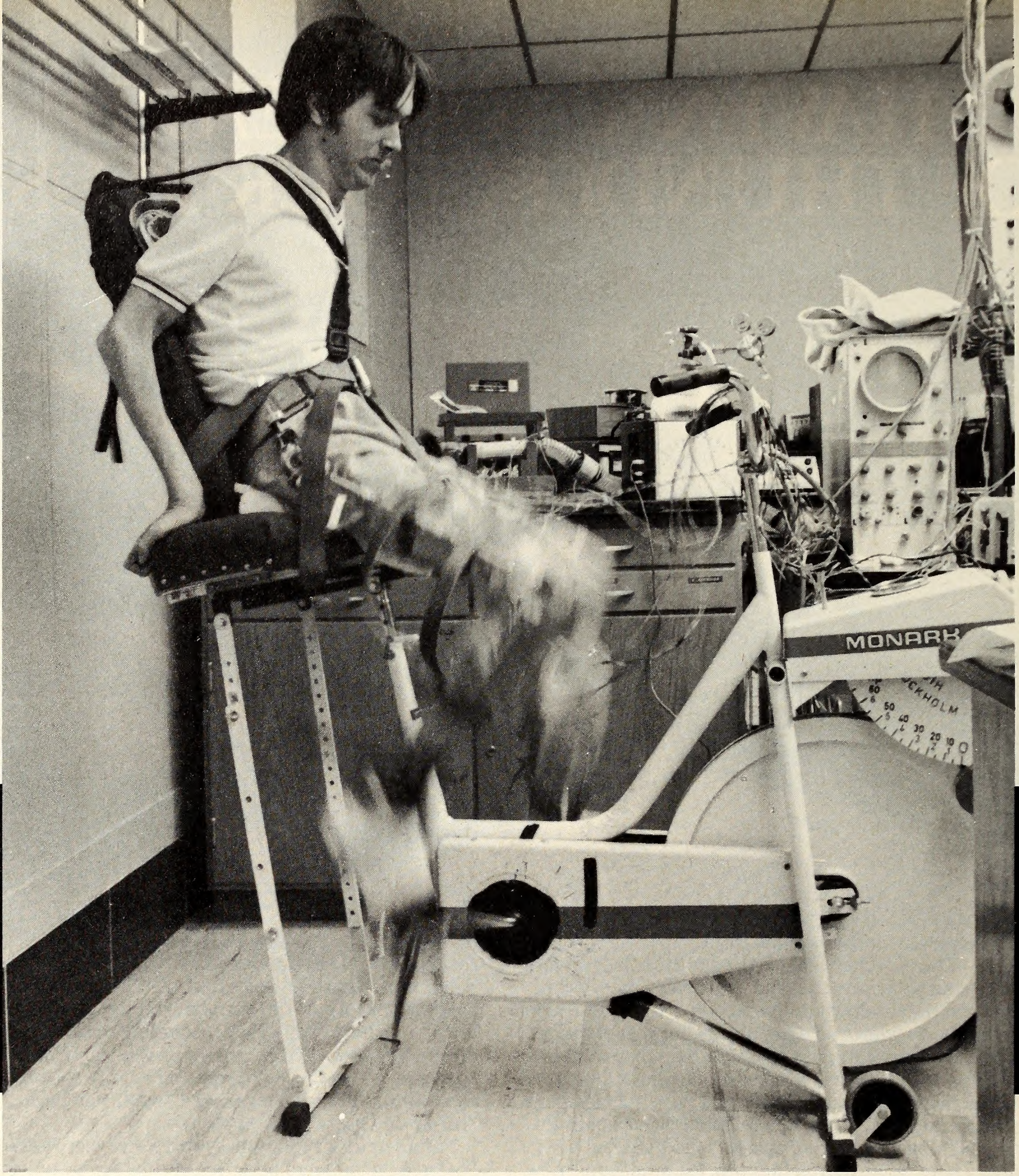
Thunderclock Plus comes with a one-year warranty. Powered by on-board batteries, it runs accurately for up to four years without battery replacement.

So now that you've heard of Thunderclock Plus, isn't it time you put your Apple to work—around the clock? See your dealer for a demonstration or contact us.

| | |
|---|---|
| THUNDERCLOCK PLUS and BASIC software | $150 |
| DOS-DATER/DEMO disk | $ 29 |
| X-10 Interface option | $ 49 |
| PASCAL software disk | $ 29 |

®Apple is a registered trademark of Apple Computer, Inc.
*DB Master is a registered trademark of Stoneware, Inc.
**Micro-Courier is a registered trademark of Microcom.
†VisiDex is a registered trademark of VisiCorp.
‡BSR X-10 is a registered trademark of BSR (USA) Ltd.

**THUNDERWARE, INC.**
44 Hermosa Ave., Oakland, CA 94618 (415) 652-1737



BY LAN BARNES

Wright State University seems an unlikely place for Apple computers to be making their mark. Named for Orville and Wilbur Wright, nearby Dayton's two most famous native sons, the semirural Wright State campus sits quietly in the middle of central Ohio's thunderstorm country, like a twentieth century Currier and Ives.

But appearances are deceiving. Underneath it all, Wright State is the site of one of the most exciting and advanced computer-aided bioengineering projects underway today, and Apples are in the thick of it. Here Dr. Jerrold S. Petrofsky, associate professor of biomedical engineering and physiology, is using computerized muscle stimulation to make it possible for people paralyzed by spinal cord injury to ride bicycles and, eventually, to walk again.

The study of physical handicaps isn't new here. The college has one of the most modern campuses in the Ohio State University system, and wheelchair access is designed into the plant. During the winter, an underground tunnel system makes it possible for wheelchair-bound students to avoid slush altogether, all of which has attracted to Wright State a large population of physically limited students.

Petrofsky's story goes back more than fifteen years to his undergraduate days at Washington University in Saint Louis, where he transferred after a year at Missouri School of Mining and Metallurgy. It was at Washington University that Petrofsky developed his passion for computers.

"While I was getting my undergraduate degree, I needed money, so in my junior and senior years I worked part-time as the electronics technician for Saint Louis University School of Medicine's department of physiology. And two days a week I worked on their computer system, adding some core memory to it."

That was a LINC (Laboratory Instrument Computer) system, Petrofsky recalls, one of the first minicomputers made for medical use, in 1959, with 1K of internal memory and magnetic tape with 250K on each dual tape drive.

"If they'd had any sense, they'd have hired somebody who knew what he was doing," Petrofsky chuckles. "But in 1967, who knew what he was doing? There just wasn't such a thing as a computer engineering degree or a computer engineer—it was all people from other fields trying

# A Dream of Walking

Opposite page: With an Apple acting as the motor cortex for his legs, Wright State University student Jeff Housch pedals an indoor stationary training cycle. Housch was the first subject in Petrofsky's earliest computer-controlled leg extension tests.

to make computers work. Microprocessors weren't even invented for another five years."

He stayed at Saint Louis University to get his Ph.D. in physiology with a minor in biomedical engineering at Washington University by special arrangement between the two schools. Then, while he was doing his postdoctoral work at Saint Louis University, he went to night school at Washington University in computer engineering (last May he finally finished enough credits to get a bachelor of technology in computer engineering).

**Setting His Goals High.** About 1969, the young physiologist/computer whiz began to toy with an idea suitable for a science-fiction yarn. Since electrodes can cause muscle contractions in people with spinal cord injury, Petrofsky reasoned, and since computers can simulate central nervous system activity, then it should be possible to use computers to bypass the spinal injury and enable paraplegics and quadriplegics to use their muscles again. And Petrofsky isn't timid: the goal was walking.

"I looked at the basic problem logically. The muscles are like a motor in your body, and when you've been paralyzed, let's say, from the neck down, your brain keeps thinking 'move,' but the messages can't make it down to the rest of your body. You have a motor down there that's healthy, a perfectly functional system. It's just turned off. So how do you bring that system back up again without blowing it apart?"

In 1969, two things were still missing that Petrofsky needed to realize his goal. The first was microcomputers, although even then it was obvious to anybody working in the field that miniaturization was the wave of the future. The second thing missing, however, was something Petrofsky could do something about. There was a void in fundamental knowlege of using outside electrical stimulation to control coordinated movement.

Thus began a long period of experiments with cats, first to determine how a cat wills itself to walk—the algorithms of cat walking—and then to see if a computer could take over the direction of a cat's hind leg. Petrofsky trained cats to walk on a treadmill in a special leg brace that limited them to a simple, manageable range of motion. Electrodes on a cat's muscles recorded the impulses from the cat's brain to the leg. Later, Petrofsky would play back the stimulation pattern on a denervated or anesthetized cat to see if the computer could reproduce a walking motion.

Progress was slow, but the basic research paid off with answers. Still, Petrofsky knew that there was a limit to how far animal experiments could take him.

In 1979 he came to Wright State, primarily because of the University's commitment to paralysis research. There his work benefited from association with Chandler Phillips, M.D., associate professor of engineering and physiology and a mathematical modeler, and Roger Glaser, Ph.D., a professor of physiology and an expert on wheelchair design and exercise physiology in paralytics.

Then in September 1981, a casual remark of a colleague changed the course of Petrofsky's research. At a think tank of paralysis researchers, another worker in the field said to Petrofsky that perhaps his research had already produced valuable results. Even though he couldn't make paraplegics walk yet, he could help them exercise their atrophied muscles with electrical stimulation.

Petrofsky immediately saw the advantages. "There were things we couldn't get out of animals. Animals are safe. If you screw up an experiment, you feel bad about it, but with people you've got to be God-awful careful, because you sure don't want to injure anybody," Petrofsky says.

"But there were things we needed to know. We know for a fact that if you take a muscle and paralyze it for three months and stimulate it, it'll come back, but who knows after ten years or twenty years? Cats don't live that long, and I don't have ten years to wait. Nobody else does either."

He realized that he needed a population that had been injured for a long time to test muscles' ability to come back again.

"Obviously, what we're trying to do is to get somebody to walk. That's an absolute. But that's really a complex problem. There is an awesome distance between stimulating gross muscle movement in a paralyzed leg and the fine muscle coordination needed for a movement such as walking or pedaling," Petrofsky says.

"One of the things you have to find out is how badly paralyzed is paralyzed muscle? If a guy's been in a wheelchair for ten years, can that muscle come back to normal strength and endurance? Well, Mike, who's on the indoor bike right now, has been paralyzed for eight and a half years, and his legs are back to normal strength and size. So apparently the answer to that is yes. Nobody knew, because nobody had ever tried to train paralyzed muscle to bring it back again. The assumption was, 'Once in a wheelchair, always in a wheelchair. Why bother?' "

Nan Davis provides the motive power for the outdoor tricycle. The more complicated movements required to ride the "Zap Bike," as it is affectionately called, are controlled by a Z-80 microprocessor running an assembly language program developed by Petrofsky on the Apple.

**The Questioning Begins To Pay Off.** As a physiologist, Petrofsky saw other questions that had to be answered with human test subjects. "Another thing was, even if the muscle's fine, how about the bones? Also, you're not using your heart that much when you sit in a wheelchair for ten years. It's just idling, especially for a quadriplegic. If you put it under the stress of walking again, it could cause a heart attack.

"And we had to find the best way to train muscle to try to get muscle to come back. We know for the average guy, if he wants to become a weightlifter, the best way to build up strength is to go into a gym and lift weights at about 60 percent of his maximum strength, very slowly, doing isokinetics. But who knows with muscle that's been sitting in a wheelchair for ten years? Maybe 10 percent of their strength is best; maybe bicycling is the fastest way. So what's the best mix to bring the muscle back, maximizing strength and endurance?"

In February 1982, Petrofsky was ready to start tests on human subjects, and the miracles began. Paralyzed student volunteers ("We'll take anybody we can get," Petrofsky says with a grin) began to make regular trips to the noisy, crowded labs on the second floor of Allyn Hall, where Petrofsky and his co-workers churn around in chaotic good humor. The place is jammed with micros: Heathkits, Intertecs, TRS-80s, and Apples, Apples, Apples.

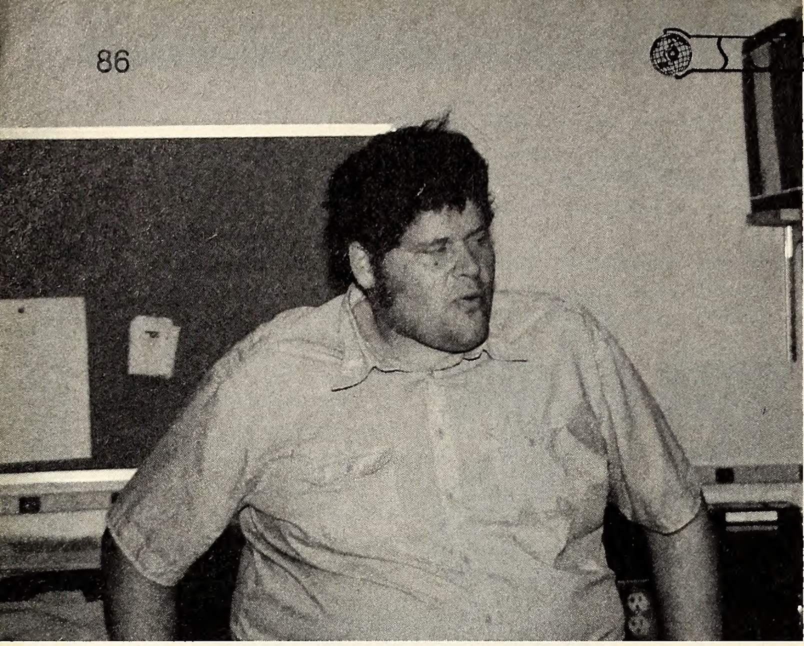
A space was cleared in the clutter, and Petrofsky and Harry Heaton, his technician, built the first strength-training machine ever designed for paraplegics—a leg extensor chair. When it came to choosing the computer that would act as the exerciser's brain, it was no contest.

"It had to be an Apple II. As far as I'm concerned it's the best value for the money that there is on the market," Petrofsky says. "It's an extremely fast, sophisticated little machine. I can go to some of the bigger systems, the S-100 systems, that cost four to eight times the price, and I'll get a little bit higher speed, but that's not worth the price.

"The Apple's also got a beautiful high-resolution graphics display. The other brands don't give as good graphics—block graphics instead of single dots. You don't get the resolution that you need.

"Best of all, it has a built-in bus so I can plug in boards, which I can't do with Radio Shack or Pet or some of the other machines. In this project, I've had to build components on cards, and I need places to plug them in. I've got direct access to the rest of the bus, and the cards are predecoded, which is great because I don't have to waste chips on the boards decoding the slot numbers since they have device-select lines set up right on the boards."

But computerniks aren't always totally logical, and Petrofsky's dedication to the Apple II has an irrational component. "We've got a little TRS-80 in the other lab, and we have chess on both," he says. "One time



Physiologist and computer whiz Dr. Jerrold S. Petrofsky.

Chan Phillips and I took the outputs and hooked the two computers together, and the Apple beat the tar out of the Radio Shack playing chess."

**A New Training Season.** With the leg extensor in place and the Apple ready to act as a motor cortex, the moment of truth had arrived. The first subject was Jeff Housch, a Wright State junior (now a senior) majoring in economics and philosophy. Electrodes were fixed over the stimulus points on Housch's quadriceps, and after Petrofsky fine-tuned the delays in the computer program, Housch watched with as much surprise as the others while his leg extended itself and then relaxed.

For Housch and the other volunteers, a period of physical training started, for now they had become athletes of a new kind. Getting them walking remained Petrofsky's ultimate goal, but more realizable intermediate goals were identified first—physical training on the leg extender, pedaling on an indoor, stationary bicycle, and developing a muscle-powered outdoor bike for paraplegics. And each intermediate goal had its own special programming challenges for Petrofsky to solve.

Where programs are just straight loop programming, like the leg trainer, Petrofsky uses Applesoft floating-point Basic ("Although it's in ROM now, so it's not really a 'soft' anymore," says Petrofsky the purist). "And that's what most of the other programs run in, too. Just floating-point Basic. The leg trainer runs in Basic because the Apple's fast enough to do what I need to do," he says.

"I use peek and poke statements to look at the machine code memory locations where I've stored A to D [analog to digital] converters. I do my A to D in memory locations off of the peripheral slot, and then there's some other circuitry built into the cards for the control systems, but aside from that I'm just running it in Basic. It only takes ten to fifteen milliseconds on the outside to code an instruction, and that's plenty quick for my purposes."

With the success of the leg trainer, things were moving quickly, but the smooth motion of bicycling is far more complex than simple leg straightening. Now the computer needed to stimulate four major muscle groups in a cyclic pattern, and it also needed to know the relative position of the feet, hips, and knees at all times.

"Originally, when the bike was put together, it got its position readings out of knee and ankle sensors, but then we found that it really wasn't necessary. The only thing you need to do is get the position of the feet," Petrofsky says.

"We started with an optical sensor on the bicycle pedals, and it just counted spokes when they broke the beam of light. It made a nice little sensor.

"The first time we got it to work just fine, and the university wanted to bring a photographer up to take some pictures, because it was the first time a quadriplegic had ever pedaled a bike, and everybody was excited. We had the guy pedaling and they shot a picture—***foomp***—goodbye optical sensor. It bombed out and screwed the whole computer program up, so we dropped optical sensors right there.

"Now we're using a 360-degree potentiometer, which I think will probably be the final way we'll do it. It's hitched to the sprocket wheel so that with each rotation you get a rotation of the potentiometer. That's all you need; that gives you position, and then the computer program handles the rest."

**Growing at a Geometric Rate.** Not all Petrofsky's concerns were programming problems. "If you take a look at an X ray of someone who's been in a wheelchair, the bones are like paper—like somebody who's a hundred years old. People in wheelchairs often break their legs if they bang them into things," he says. "But we're finding that the bone is coming back fairly quickly; still, fairly quickly for the bones means about six months, whereas it takes about two months for the muscles to come back.

"So we've got a flag of caution that's thrown up: 'Be very careful when you bring the muscles back, because if you bring them back too fast, you'll snap the bone.' The muscle comes back two to four times faster than the bone."

Even with all that caution, the group had a protocol-related hairline fracture that Petrofsky still finds very strange.

"We haven't figured it out yet. A normal person's leg extension strength might be about seventy-five pounds. Well, we had one of our subjects start out with about twelve pounds of leg extension strength, and after about two and a half months he was up to ninety pounds. We figured, 'Fine, he's going to level off.' Because that's at least a little bit above normal strength.

"Every two weeks we test the subject's strength, so two weeks later we tested him, and he broke the meter at one hundred thirty pounds. He was just training for thirty minutes a day, three days a week, and with that type of workout, he doubled the strength of the average man. His bones just weren't ready for that kind of stress, and he got a hairline fracture.

"So, one of the things I'm doing this week is modifying the equipment so that if strength exceeds a certain limit mechanically the system shuts down."


## THE SECOND SCENARIO

**The Wizardry Adventure Continues!** Barbarians gather, preparing to sack the city of Llylgamyn. News that the mighty staff of Gnilda has been reclaimed by its earthgod creator has travelled fast. Protected for a thousand years by the power of the wonderous artifact, and now defenseless, Llylgamyn is doomed, unless you help!

What have people said about the first scenario?

*. . . [Wizardry] has the potential to become a classic.*
David Lubar, *Creative Computing*

*The most eagerly awaited adventure . . .*
*Computer Merchandising*

*. . . perhaps the most advanced adventure program on the market.*
Forest Johnson, *The Space Gamer*

*The amount of detail is fantastic.*
Neil Shapiro, *Popular Mechanics*

*Knight of Diamonds*, a challenging scenario for experienced *Wizardry* players (13th level characters will barely survive!).


*Proving Grounds of the Mad Overlord* (shown at right) and *Knight of Diamonds* operate on **any** Apple Computer* with at least 48K, DOS 3.3, and 1 disk drive.

*Proving Grounds of the Mad Overlord*—**$49.95**
*Knight of Diamonds*—**$34.95**

Join over ten thousand adventurers exploring the ultimate fantasy experience.

MasterCard VISA C.O.D.

**Shipping $2.00 in U.S.; $5.00 outside U.S.**

***NOW AVAILABLE AT YOUR LOCAL COMPUTER AND HOBBY STORE***

*APPLE IS A REGISTERED TRADEMARK OF APPLE COMPUTER, INC.

Sir-tech SOFTWARE, INC.
6 MAIN STREET, OGDENSBURG, N.Y. 13669
(315) 393-6633

Despite this setback, the outdoor bike project went ahead. But in this endeavor, Petrofsky tempered his admiration of the Apple II with a little of his old love of 8080 assembly language.

"The Apple is a nice machine. For most of our applications where we have to run something in Basic, the Apple does just fine, although it just about gets overrun by controlling four muscles simultaneously.

"So, for the outdoor bike, I used Z-80 assembler because I'm very familiar with it and with CP/M. I've got the *Lisa* assembler for the Apple, but I'm less familiar with *Lisa*."

But the outdoor bike's central controller was also developed on the Apple as an indoor simulator with the Z-80 card.

"That whole system is handmade. I use the Z-80 board on the Apple in the far lab, which is set up to run CP/M. Then I run a cable from a plug-in Apple port to the I/O to the outdoor bike. Then there are wheel stands under the outdoor bike to prop it up, so when we have somebody pedaling the bike, it runs the output through the Apple. That way, if I have any software errors on the on-board Z-80 computer, I can find them on the Apple.

"The Apple ports are set to be identical to the ports on the little computer that I built with the hardware Z-80, so when everything looks good, I get the program in machine code and store the assembled binary version in high memory. I have a program for burning PROMs on disk in CP/M Basic, and I call that program in, go into high memory, and burn a PROM off.

"That Apple is strictly for testing the outdoor bike, modifying it, and burning PROMs. Actually, I was happy with the second one I burned. I also have a couple of PROMs I blew that check the system out. They generate sine waves and triangular waves in different channels and check the different inputs to see if the system's running right."

Now that true mobility was the issue, other conditions had to be considered, and Phillips's mathematical models became more important in Petrofsky's programming. Away from the controlled conditions in the laboratory, the programs had to handle the effects of ambient temperature on muscle performance and each subject's strength and work capacity. And, of course, there are always hardware glitches.

"When we got the outdoor bike running, I didn't like one of the parameters. It was okay when we were using it on the Apple as a simulator, but when we actually got it in the field and had the bike going down the street instead of sitting on a jack stand, I didn't like one of the timing parameters, so I blew a second PROM," Petrofsky explains.

Then there was throttle control. "I built a throttle control that was like a motorcycle throttle, and Harry didn't like it," Petrofsky says. "Then he built one that was nicer looking, but it started causing crosstalk with one of the other A to D channels and was screwing up the out-

# The Algorithms of Movement

All of Petrofsky's control programs share a similar design. Everything is done in "closed loop" programs, simply meaning that the computer sends a stimulus; then monitors a sensor to see what the leg did; and then goes through a logic tree, deciding whether to increase stimulation (or decrease it or stay the same) depending on what happened.

On the single leg trainer, the program generates a synthetic saw-tooth wave with a period of three seconds up, three seconds down, and then adjusts the output current to the electrodes until the leg sensor has the same output voltage as the wave. If the leg is too low, it increases the stimulation; if the leg is too high, it decreases the stimulation.

Obviously, Petrofsky says, the trick on something like that is real-time constants. "You've got to recognize that this is not a computer," he says, slapping his leg. "It's not going to respond in a microsecond. So, if the Apple gives the leg an additional increment voltage, it's got to wait the right amount of time before it tries to give it more voltage to get a response.

"In fact, one of the really nice things about the Apple is, like all microprocessors, it's a little bit slow, which gives me the time delays I have to have to allow the body to respond. So rather than being a detriment, the slower speed of running Basic helps the program."

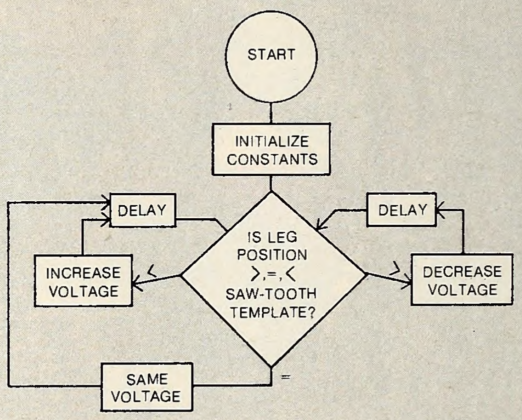
But Petrofsky needed more speed for the bicycle programs. "I had to do some programming tricks in Basic to make the thing run. In order to get the speed up, I spend the first few lines in the program defining all the variables that are stored in floating point. And then I concatenate all the statements, so I only have a few program lines, with everything in variables; no hard numbers. That way it runs faster than all get-out."

However, Basic gets left behind with walking. "The trainer was simple," Petrofsky says. "The indoor and outdoor bikes, not too bad. But the standing algorithm is a lulu, and the walking algorithms are even worse, because suddenly you've got pressure sensors on the front and back of the foot and on three joints on each leg, then a gyroscope showing equilibrium, and you've got to test and coordinate every sensor constantly.

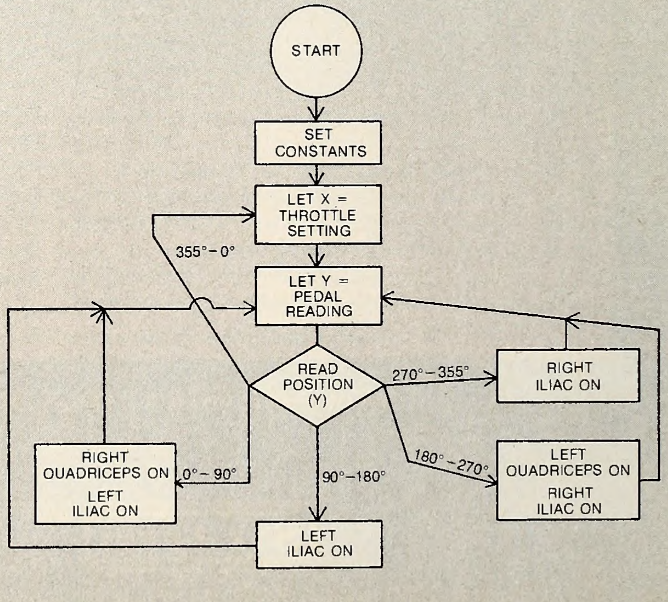
"You've got say, ten sensors, and with every individual change of each sensor you want a whole different action depending on the other readings. '*If* this is here; *and* that is there; *and* that is going there; *xor* something else is going somewhere else; *then* do this; *else* do that'—a couple of thousand permutations in the control program.

"And that's just for smooth ground. If you were to go up steps or uphill or downhill, all hell breaks loose. If you try to sit up out of a chair, it gets even worse. And if you're on ice or grass, it gets really nasty."

LEG TRAINER ALGORITHM




BIKE ALGORITHM




STAR MAZE
Star Maze, a tour-de-force of hi-res animation and graphics. Yours is a desperate mission—to find and recover the power jewels hidden in the maze before your fuel runs out.
Star Maze features animation effects never seen before on an Apple as well as the fine touches that mark it as a Sir-tech game.
• huge 18 color multi-directional maze
• 16 maze levels
• playable by keyboard, paddles, joystick or joyport
• high scores saved to disk
• and more . . . all for $34.95
Programmed by Gordon Eastman from an original design by Robert Woodhead
NOW AVAILABLE AT YOUR LOCAL COMPUTER AND HOBBY STORE
MasterCard VISA C.O.D.
Shipping $2.00 in U.S.; $5.00 outside U.S.
SOFTWARE THAT COMPLIMENTS YOUR INTELLIGENCE
SIR-TECH SOFTWARE INC.
6 MAIN STREET
OGDENSBURG, N.Y. 13669
(315) 393-6633

put. So I dumped it and went to a game paddle. Now we want a throttle that doesn't crosstalk and doesn't look like the guy's playing *Pac-Man.*"

**The Apple Gets Zapped.** Eventually it worked. The "Zap Bike"—actually a large, adult-style tricycle that eliminates the problem of balance—took a tour of the Wright State campus with Nan Davis providing the motive power and a home-built Z-80–based microcomputer comprising a half-dozen chips providing the coordination for her movements.

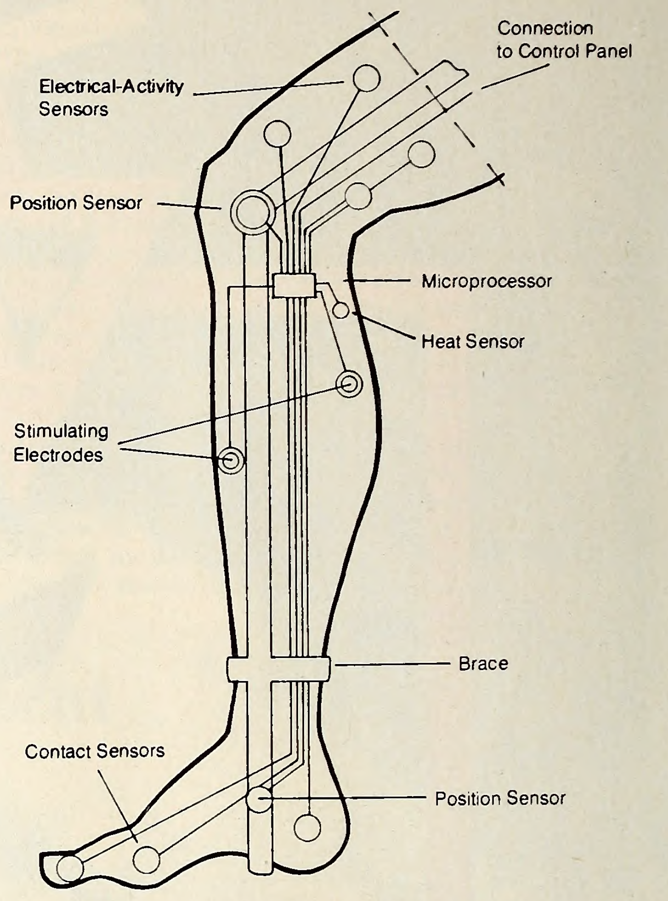
Petrofsky, however, is ambivalent about the success of the Zap Bike. He sees it as having contributed two things to their overall progress. First, he appreciates the practical experience for everyone on the team, especially Phillips, whose mathematical modeling benefited. "I write the programs, but he does the equations," Petrofsky says. "It was a chance for him to look at his equation and for me to check out control programs."

The second benefit was that it gave additional training to the test subjects. Instead of one muscle moving, now whole legs were moving, putting even greater stress on the heart and lungs and pushing training one step further.

"I've got mixed emotions about the outdoor bike," Petrofsky confesses. "In terms of an engineering design, it was well worth the effort, because it's designed up from scratch so a paraplegic or quadriplegic can head out on a pretty day and pedal all over the park with it. It makes them mobile again, and they don't have to sit in a wheelchair.

"And so it's very easy for them to hop in the chair or get helped into it and off they go, pedaling around the neighborhood. In fact, we're working on putting carbon electrodes in some slacks, so they can pull the slacks on and all the electrodes are automatically on. That's practical.

"In terms of physiology and the whole project, it didn't do a lot. Aside from the fact that it was a fun piece of engineering, and also it was the first time paralyzed muscles had ever been used to get somebody to move again, to propel themselves, it was a God-awful waste of time. One month down the tubes on that stupid computer, getting out the crosstalk between the different circuits—that computer has only got about eight chips on it, so I had a lot of problems with software and hardware. It went through a couple of drafts and a lot of broken boards before I got it up the way I wanted it, all battery operated and independent.




Blueprint for mobility: The microprocessor just below the knee accepts feedback input from contact, heat, electrical activity, and position sensors placed at strategic locations on the leg brace and the leg itself. The microprocessor stimulates muscles of the lower leg through electrodes.

"In that respect, we could have had somebody walking by now if we hadn't been messing with that bike," Petrofsky laughs. "I've got a lot of nights tied up in that bike."

So the target is, and will always remain, walking. The latest addition to the lab is a walking support consisting of an I-beam supported on a two-inch by four-inch frame. The researchers have suspended a parachute harness from a free-running wheel on the beam, so that their paraplegic volunteers can practice computer-controlled standing and walking without fear of falling.

With walking, however, the project leaves the venerable Apple II behind. "The Apple couldn't keep up with it really," Petrofsky says. "I need to control four times as many parameters ten times as fast, and the Apple was a little bit marginal as it was for the bicycle. So I'm using a Z-80 S-100 system for the walking experiments."

How can all this be measured in human terms? The paralyzed volunteers make it clear that more is involved than just mobility (all of them are already equipped with motorized wheelchairs). Kevin Leonard, a Wright State sophomore who has been with the project since May, says that since he started exercising his legs, the involuntary muscle cramps that used to wrack his legs have all but disappeared.

And although Leonard hasn't ridden the outdoors bike yet, he's looking forward to it. "Given a choice between the Zap Bike and my wheelchair, I'd take the bike," he says. "I'd be getting more exercise that way—and there's something about using your own muscles to move."

Housch agrees with Leonard on that, but, like Petrofsky, Housch is looking to the future. "I'd like to see us walking," he says. "I don't see how Jerry's going to do it, but he says he will, and so far he's done everything he said he was going to do." ∎


*Galactic Attack.* The first sophisticated home computer game with action-packed tactics *and* challenging strategy. Easy to play, *Galactic Attack* takes skill to master.

Outnumbered by savage invaders, you must outsmart as well as outfight your deadly opponents. Your mission is to defeat enemy ships in space with Photon Torpedos and Phasers, while recapturing planets by bombardment and landing your ground forces. You control weapons, damage control, warp speed, sensor scans and protective shields. But take warning! *Galactic Attack* is a real-time game. A moment's indecision may cost you your ship.

The high resolution graphic display indicates the constant movement of ships, planets, and torpedoes. Continuous readout gives your speed, heading, fuel level, shield strength, damage report and status of ground forces.

With a variety of skill levels and situations to choose from, *you* determine the complexity and duration of each game.

You need only DOS 3.3, 48K Apple II/II+ with 1 disk drive—**$29.95.**

**NOW AVAILABLE AT YOUR LOCAL COMPUTER AND HOBBY STORE**

***'SOFTWARE THAT COMPLIMENTS YOUR INTELLIGENCE'***

**SIR-TECH SOFTWARE INC.**

***6 MAIN STREET***
***OGDENSBURG, NEW YORK 13669***
***315-393-6633***

MasterCard VISA C.O.D.

**Shipping $2.00 in U.S.; $5.00 outside U.S.**

# Announcing an Exciting New Way to Make Effective Presentations


Now you can produce professional-looking slide shows—complete with graphs, charts, and special effects—quickly, easily, and economically. All you need is an Apple Computer, a little imagination, and the Executive Briefing System.

The Executive Briefing System is a powerful new software program that transforms your Apple computer into a tool for creating full-color graphic presentations—Dynamic presentations that will make your point or sell your product more effectively.

To see what this exciting new product can do for you, visit your local Apple software dealer today.

LOTUS

55 Wheeler Street, Cambridge, MA, 02138 (617) 492-7171

# Mind Your Business

BY PETER OLIVIERI

For many readers, it's nearly time to get out the long underwear, throw another log on the fire, and snuggle up with a good user manual. One thing about winter is certain . . . it makes you appreciate summer.

This month we'll be talking about a variety of things. We'll hear from a BUG (Business User Group) member, examine a business applications package, and begin our long-awaited look at the Apple III.

**Apple III Overview.** Most people have heard something or other about the Apple III. Indeed, there are a lot of impressions of this machine and its capabilities. To Apple Computer's credit, the "new" Apple III has none of the problems the earlier machines had. The Apple III is now exactly what it was intended to be originally: a quality machine designed specifically for the small business user.

This month we'll examine the hardware that might comprise a typical small business system. Next time we'll focus on the documentation, the systems software, and some of the applications software available for the Apple III. In future columns, there'll be a special section devoted exclusively to Apple III items.

**The Hardware.** An Apple III system designed for the typical small business user would probably consist of these basic components: the Apple III itself, with 128K of main memory (a 256K version is available) and a built-in 5¼-inch floppy disk drive; an Apple III monitor (or similar display device); a second disk drive, which could either be the Apple III 5¼-inch floppy drive or the ProFile hard disk; and a printer.

This represents a typical arrangement, but it's not the only way to put together an Apple III system. As your business grows and your needs change, you can upgrade and expand your system, adding equipment as you need to. You can, for example, connect as many as three floppy disk drives.

The machine itself is thoughtfully designed. The Apple III monitor fits well on top of the main machine and the built-in floppy disk drive is convenient and just as easy to use as the separate disk drive unit.

The keyboard on the Apple III differs considerably from the Apple II keyboard. It has a numeric keypad that facilitates the rapid entry of numbers. On the main keyboard, you'll find up and down arrow keys; a tab key (that actually works); some new symbols, such as a back slash, ellipsis points, and two sets of brackets; and two special programmable keys. The entire keyboard has a nice feel to it, and touch typists are likely to appreciate the raised dots that have been strategically placed on some of the keys.

The reset button is tucked behind the keyboard proper, well away from wandering fingers. It is evident that thought went into the design of this keyboard. In addition, there are a couple of different character fonts to choose from.

The back of the Apple III has the usual on-off switch and AC cord connection in addition to seven other plugs. These include an external disk drive connector (no more having to take off the cover), two joystick ports (one of which can also be used as a place to connect a Silentype printer), a color video port, a black and white video port, an audio port for sending audio to a speaker or a tape recorder, and a serial port that allows you to connect the Apple to all devices that use the RS-232-C standard communications format.

**Inside the Apple III.** The most important parts of the Apple III are on the inside. You won't need to do much on the inside of the machine, however, particularly given the arrangement whereby the printer and disk drives connect to ports at the back. Removing the cover entails turning two screws that are located underneath the machine itself. When you remove the cover, a small light warns you if the power is still on.

As you peer inside the machine, you'll notice the disk drive and power supply almost immediately. Between these two items are four long connector slots. It is here that the interface cards are placed. These cards might connect to a plotter, to test equipment, or to a modem for communicating with another computer.

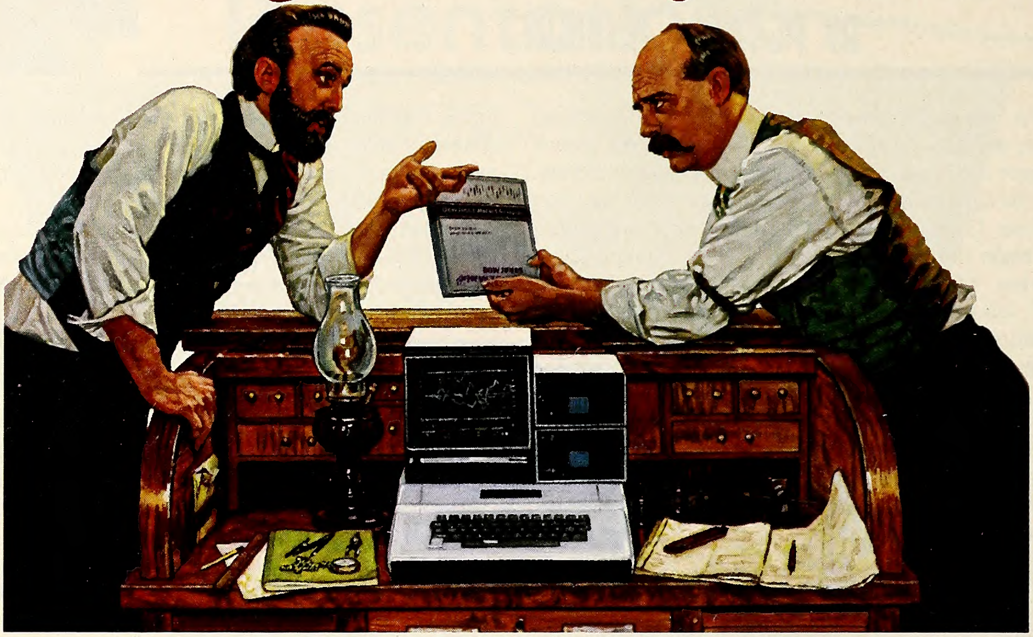
The most likely card to include is the one that connects to the ProFile hard disk system. Once this card is installed, the ProFile itself can be attached without removing the cover of the machine.

Rounding out the inside of the Apple III are an internal speaker and a security mount. This mount enables you to bolt the Apple to a table or shelf or secure it to a chain to discourage theft. The III also has a built-in fan.

**Profile of a ProFile.** The Apple III hard disk is called the ProFile. It's about the size of a shoe box (size 12 sneakers would fit in nicely). The ProFile consists of a Winchester hard disk drive (which essentially means that it is self-contained, environmentally sealed, and very fast) that can store five million characters, approximately equivalent to what can be stored on thirty-five floppy disks. To put it in even more familiar terms, the ProFile can store about the equivalent of twelve hundred typed pages of single-spaced text.

The ProFile is fast, quiet, and very easy to use. Graphics can be displayed at a speed that is certain to surprise you. In addition, the ProFile does an automatic surface analysis and error-detection routine each time it is started up—to ensure the integrity of the stored data.

# *Mr. Dow and Mr. Jones* *introduce* *Dow Jones Software™*



**Jones:** "Mr. Dow, look what they're selling in that new store down the street: Dow Jones Software. You haven't gotten us into ladies' fashions, have you?"

**Dow:** "No, Mr. Jones. That's a computer store, and our software products allow investors and business professionals to use a personal computer like this one here to easily manage financial information."

**Jones:** "But what about our reputation? We've been leaders in serving the business and financial community for over 100 years. Are you sure this new software will be as reliable as The Wall Street Journal and Barron's?"

**Dow:** "Of course, Jones. Our software is so reliable we back it up with a full-year warranty. People trust Dow Jones Software the same way they trust the Journal. And we have a toll-free Hotline number in case they want expert help."

**Jones:** "Couldn't that be a lot of phone calls? After all, we've got the Dow Jones Averages to get out every day."

**Dow:** "Don't worry, Jones. Our software is very easy to use, and we have a fully staffed Customer Service Department to respond to our dealers and customers."

**Jones:** "Just what can our software do?"

**Dow:** "In a nutshell, Jones, with a personal computer, a telephone, a modem and Dow Jones Software, you can easily perform complex analyses on the information available from our information service, Dow Jones News/Retrieval®."

**Jones:** "People really use our software to make decisions?"

**Dow:** "Absolutely. Once you've stored the information you want, our software does the rest. For instance, with one Dow Jones Software product you can follow indicators for stocks, sort, rank, screen and set critical points for buying and selling. With another, you can easily construct technical charts. Look at this beautiful graph."

**Jones:** "You mean all those calculations I've been doing by hand I could do in a fraction of the time with this software? That's great!"

**Dow:** "It is, Mr. Jones. Just like the Journal, Dow Jones Software is a resource you can bank on!"

---

**Dow Jones Market Analyzer™**
**A technical analysis product** that allows private and professional investors to automatically collect, store and update historical and daily market quotes, and to construct technical analysis charts at the touch of a key.

**Dow Jones Market Microscope™**
**A fundamental analysis product** that allows professional money managers to choose and follow indicators for extensive lists of stocks and industry groups, and to sort, rank, screen and set critical points for buying and selling. (Available 4th quarter 1982.)

**Dow Jones Market Manager™**
**A portfolio management product** for private or professional investors who desire immediate access to pricing and financial information, and who need an accounting and control system for their portfolios of securities. (Available 4th quarter 1982.)

**Dow Jones Connector™**
**A communications product** for the business or professional person who wants instant electronic access to news, facts and vital data at the home or office, via personal computer, simple terminal, communicating word processor or teletypewriter.

**For dealer information and a free brochure, call our Customer Service Hotline number:**
**1-800-257-5114**
**(in NJ 609-452-1511)**
**Sold through computer retailers.**

**DOW JONES SOFTWARE™**

***...Bank on it.***

The only drawback to the ProFile hard disk system is that there's no simple way to back up the information you have stored. You would have to back up to floppies, which, by the way, store 143,000 characters. Nevertheless, once you get used to using a hard disk, you may find it very difficult to adjust to using anything else.

**Pluses and Minuses—a Quick Summary.** Using the Apple III is easy and, in many ways, a real pleasure. Among the machine's major strengths are its large main memory (from 128K to 256K), the ease with which the system can be expanded, the microprocessor speed, and the comfortable keyboard with its new capabilities. The separate keypad is useful. It's also handy to be able to connect devices at the rear of the machine, instead of having to hook them up inside.

Both upper case and lower case characters can be generated, controlled by the shift key. Text can be displayed in either forty columns or eighty columns and in color if you have a color monitor. There are three graphics modes on the III: 280 by 192, 140 by 192, and 560 by 192.

Another plus is the Apple III operating system (SOS), which handles files easily and makes possible the interchange of data from different applications programs.

The Apple III is reliable, with none of the problems that had been associated with its introduction. Designed with both the user and the business in mind, the machine is attractive, functional, and human engineered.

What about the Apple III's disadvantages? Well, one weakness is that the operating system and the device controlling programs take up a fairly large chunk of memory (as much as 64K). Also, while the machine does operate faster than the Apple II, its actual processing speed is slowed down considerably by the number of SOS instructions needed. Some floppy disk space is taken up by device drive programs and such, which are stored on disk.

The machine itself is much heavier than the Apple II; consequently, it's not as easy to move the III from room to room or to take it home. And, as we discussed earlier, the ProFile hard disk drive has a cumbersome back-up procedure.

All things considered, the Apple III is a well-designed machine. Certainly, it is a leading candidate for small business users to consider. A machine must be more than attractive and reliable, however, if it is to become "the machine of choice" for people who are involved in small businesses. It must be supported by good documentation, good systems programs, and a wide variety of applications software.

Next month we'll see how the Apple III measures up in these areas. We'll look carefully at the documentation that comes with the III and at many of the currently available applications packages. We'll conclude with a comprehensive summary and critique of the Apple III's strengths and weaknesses as a business computer.

**Business User Group.** We're beginning to get responses to the request for information about your experiences with printers and word processing packages. Please keep them coming; other users will find your input helpful. Send your comments to Softalk BUG, Box 60, North Hollywod, CA 91603.

Meanwhile, Robert Jordan from Memphis, Tennessee, mentions that he is quite happy with his IDS 440G Paper Tiger Printer, for which he paid $900. He cites the strengths of the machine as its speed and ruggedness. Two of its drawbacks are the fact that its letters have no true descenders and that replacing the printhead is expensive. While Jordan likes the 440, he feels that the 460 and 560 may be better machines.

Our thanks to Bob Jordan for kicking off our BUG printer/word processing comments.

**On to Applications—Investigating Infotory.** One very common application in a great many businesses is inventory control. Good management of inventory can be essential to the success of an organization.

Because it involves maintaining item counts and doing calculations, inventory control is an ideal application for the computer. Oddly, however, there are precious few inventory systems available. Most database management systems now on the market can perform an inventory control function. And, in fact, the inventory control situation is sometimes cited as an example of a database application. Nonetheless, organizations that are well versed in the problems associated with the day-to-day control of inventory would probably welcome a package specifically designed for that purpose.

One such package is *Infotory*, an inventory management program by SSR Corporation.

Managing inventory is important to an organization. Members of a company's sales force want the inventory available so their customers can get the best service in the shortest time. The finance department wants to keep the inventory at a level that minimizes storage and carrying costs. The *Infotory* package is designed to help manage these sometimes competing objectives.

The system offers the following features:

1. It can keep track of and report on as many as fourteen hundred items.
2. Inventory is managed from its receipt to its sale.
3. Purchase order management information is provided.
4. New items can be added to the inventory.
5. Item inventories can be updated.
6. Inquiries can be made about any item.
7. Inventory reports, price lists, and sales/cost analysis reports can be printed on request.
8. A complete audit trail is printed for all transactions made. This is an unusual and most welcome feature.
9. A special feature, called Anyreport, makes possible the creation of customized reports.

The system requires an Apple II Plus with 48K and ROM Applesoft, a monitor, an eighty-column printer, DOS 3.3, and two disk drives.

To begin using the *Infotory* system, you must first prepare your current inventory information for entry into the computer. The initial chapter in the user guide explains what you must do to make this conversion successfully. The manual includes a data collection form to aid you in this process, and it's easy to make extra copies of this form to accommodate large inventories.

All terms that are used on the form (or anywhere in the package) are introduced and clearly defined. Following the explanation of how to convert your existing data so that it is ready for *Infotory*, you're presented with a glossary of terms, a definition of all data fields, and examples of all formulas that are used in calculations (for example, cost on hand, gross profit, margin, and so on).

The user guide is among the best available. It is clearly written and contains more than one hundred screen illustrations. Sometimes, what you get for a screen illustration is a blank screen. But then if that's what you're going to see when you start up the system, perhaps displaying it in the manual is not such a bad idea. The authors of the manual seem to have anticipated all the possible little mistakes new users might make.

When you start up the system, you can enter daily information (dealing with items received, items sold, items on order), look at or generate information and reports (choices include item information, inventory reports, price lists, sales and costs analyses, the Anyreport generator), or take care of data maintenance tasks (add or delete inventory items, change or update items, clear sales and cost data, initialize new disks). The user guide describes each of these menu items in detail.

Each inventory item has an item number, a description, a price, a reorder level, a location, and a vendor associated with it. As you use the system over time, additional information about the inventory becomes available. For example, you can get period sales, average costs, quantities on hand, quantities on order, profits, margins, order costs, and carrying costs.

The Anyreport feature, which is much like a typical database search, is one of the most useful features in the package. You can use the Anyreport option to search through the inventory and find all items currently in stock that cost more than $50.

The setup for making selections is particularly nice. First, a list of all


# WILL MAKE YOUR APPLE II THE JACK·OF·ALL·TRADES FOR $79.00

**Four applications in one.**
The Incredible Jack combines the convenience of a personal filer with the power of a calc package. It handles most word processing tasks with ease. Toss in the ability to sort and print mailing labels, and you have a totally integrated, suprisingly easy to use package that does most of what you bought your Apple for.

**Organize information your way.**
The Incredible Jack lets you arrange your information in "records" you design yourself using the computer display. Each record may be as little as a mailing label or as large as 60 sheets of legal sized paper.

**Word processing made easy.**
With a few simple commands you can master in minutes, you can create letters, memos and reports. To help you edit, Jack lets you insert, delete, and copy portions of text all with automatic word wrap and flush right margins. The Jack does away with the mind boggling control codes and formatting options of other word processing packages.

**Automatic decision making.**
Jack even allows you to build decison making logic into your file. With English language rules and a powerful IF THEN ELSE function, you can instantly calculate complex discount tables, commission plans, contract terms, or tax rates.

**See for yourself.**
Try Jack. (You'll need an Apple II with 16K memory extension and 2 disks.) If you're not convinced it's the best investment you've made since you bought your Apple, send Jack back in good condition within 2 weeks for a full refund.

To order call: **(800) 645-4513.** For information: (516) 269-1120. To order by mail, enclose $79.00 plus $5.00 shipping and handling. (charge card # and exp. date) MasterCard VISA

N.Y. residents add 7¼% tax, and send to: Business Solutions, Inc.
60 E. Main St., Kings Park, N.Y. 11754.

**THE INCREDIBLE JACK.** T.M. business solutions

WILL MAKE YOUR APPLE II THE JACK OF ALL TRADES the fields is displayed on screen and you're asked to type yes next to those you want printed up. Next, a list of all the fields is displayed once again and you're asked to type a field number, a comparison operator (a less than, more than, or equals symbol), and then another field number or some specified value. This option makes possible the printing of some pretty fancy reports.

Not much has been left out of this system. There are times when the extended costs (costs times quantity) must be calculated manually before entry, a task the system might have been designed to do. In addition, it might have been nice to have the capability of entering more information about a vendor, although this would have no doubt used up some of the storage space available for storing inventory data. Finally, no mention was made in the documentation about back-up copies of the program disks, although backing up one's data disks was discussed in detail. These are relatively minor complaints, however, considering the advantages the system offers.

Before we conclude our discussion, it might be helpful to look at some of the data limitations of the *Infotory* package. As mentioned earlier, the system accommodates fourteen hundred items. The allowable field lengths of the data items are shown in figure 1.

If these parameters are within the range of those you're working with in your inventory system, it would be worth your while to give serious

| | |
|---|---|
| Reorder level | 9,999 |
| Period cost | 999,999.99 |
| Period sales | 999,999.99 |
| Item number | 15 characters |
| Description | 20 characters |
| Location | 4 characters |
| Vendor name | 3 characters |
| Price | 9,999.99 |
| Average cost | 9,999.9999 |
| Quantity on hand | 9,999 |
| Quantity on order | 9,999 |
| Quantity sold | 9,999 |

Figure 1.

consideration to *Infotory*. The programs in this system are easy to use, error messages are clear and informative, and the documentation is unusually well done.

**So Long.** Well, that's it for another month. Next time we'll continue with our Apple III profile, look at some more business packages, check in with BUG, and discuss some preliminary Inquisition results. Have a great month.

*Apple Computer, 20525 Mariani Avenue, Cupertino, CA 95014; (408) 996-1010.*
*SSR Corporation, 1600 Lyell Avenue, Rochester, NY 14606; (716) 254-3200.*

# All About Applesoft

by Doug Carlston

One of the most fascinating and powerful capabilities of your Apple is its ability to manipulate strings. A string is a sequence of characters—any characters. This sentence is a string. The number "1000" is a string (four characters long); so is "one thousand" (12 characters long). You can define any variable ending with a dollar sign as a string, like this:

```
A$ = "GOOD MORNING"
```

Now, whenever you *print A$*, the phrase "Good Morning" will appear on the screen. Old hat, you say? Perhaps, but take a look at the list of new Applesoft commands that we are going to learn this time, most of them designed specifically to manipulate strings:

| | |
|---|---|
| STR$(x) | VAL(a$) |
| LEFT$(a$,x) | RIGHT$(a$,x) |
| INT(x) | MID$(a$,x,y) |
| STEP | PEEK (−16336) |

With these few additional commands, we will be able to manipulate strings in countless useful ways—but let's count them anyhow.

First, we will design a formatting routine that will take any number and turn it into dollar and cents format, with two digits to the right side of the decimal and commas dividing large numbers every three digits.

Second, we will design a string analyzer that can separate first and last names in a mailing list for proper use in a form letter. (Yes! You too can create junk mail with a personal touch!)

Third, we will design a routine that will take any message and scroll it across the screen Teletype-style. The usefulness of this routine is somewhat uncertain, but it is fun.

Fourth (and last), we will design a very efficient branching routine that will search keyboard input for a given list of control codes and then jump to the appropriate line in a program. The usefulness of this routine is more than sufficient to make up for number three above, although it doesn't have much pizzazz.

Now let's see if we can deliver on all those promises. The first routine should be able to take a number, such as 12345.6, and convert it to dollar format ($12,345.60) before displaying it on the screen. This will make numbers much more readable.

The way to do this is to take the numeric variable value and change it to a string, which we can then play with:

```
10  INPUT NUMBER
20  A$ = STR$(NUMBER)
```

Line 10 allows us to enter any number, which is stored as the numeric variable Number. Line 20 converts the numeric variable to a string of characters and stores the string in the string variable A$. For example, if we type in the digits 00123, the variable Number becomes the value 123. Then line 20 converts the number 123 to the string of three characters 1, 2, and 3, and stores that string in the variable A$.

To add the dollar sign to the front end of the string we just concatenate our string with another:

```
50  A$ = "$" + A$
```

It's a little more complicated to get the cents part right. If a number has too many digits to the right of the decimal point, it's pretty simple to round it down to two, using the INT function. INT takes the value of whatever numerical value it is given and turns it into an integer by throwing away that part to the right of the decimal. For example, if N = INT(3.14159), then N equals 3.

Using this to round our number off to two decimal places is simple—just multiply the number (*not* the string) by 100, INT it, and then divide the result by 100. Like this:

```
15  NUMBER = INT(NUMBER*100) / 100
```

This works fine as long as our number has too many digits to the right of the decimal—1.2345, for example. But what about numbers with too few digits? We have to pad them out with zeros so they come out even—1.1 becomes 1.10, and 5 becomes 5.00.

It's easier to do this by fiddling with the string A$. We know that if the number had at least two digits to the right of the decimal point, then line 15 will have handled it properly, and there should be a decimal point exactly three characters from the right end of the string. We can look for that decimal point by using the mid$ command.

Mid$, left$, and right$ are three commands that return part of a string to you. They work like this. Suppose that

```
A$ = "A123BCD"
```

and that

```
B$ = LEFT$(A$,3)
```

Then B$ will be equal to the left three digits of A$ or, in this case, "A12."

As you can see, left$ has two parameters. The first one is the string that we are going to take the left part of. The second tells how many characters to take. If we had set B$ equal to right$(A$,3), then B$ would have been equal to "BCD."

Here is an example. Can you figure out the value of Answer$ without running the program?

```
T1$ = "VERTICALLY": T2$ = "FOOD"
ANSWER$ = LEFT$(T1$,3) +          RIGHT$(T1$,1)
          + RIGHT$("B U G",2) + RIGHT$(T2$,3)
```

Mid$ is a little more complicated. As its name implies, it can be used to pick a string out of the middle of another string rather than just one end or the other. However, to do this it not only needs to know the source string and the length of the segment it will be copying, it also needs to know where to start copying from. For example, let's say you want to pick out two characters from a string, starting four characters from the left. Your command might look like this:

```
ANSWER$ = MID$("UNANOLOG",4,2)
```

Answer$ is equal to the substring "No." The first parameter in parentheses is the string to be examined. The second is the starting position in the string (counting from the left), and the third is the number of characters to be taken.

Let's get back to our problem now. We want to find that decimal point and, one way or another, make sure that there are two digits to the right of it. The first thing we can do is check to see if the decimal point is already in the right place. If it is, we're all set and don't have to worry about monkeying around with the string. What should our test look like?

```
25  IF MID$(A$,LEN(A$)-2,1) = "." THEN 40
```

What this says is find the last character in A$ and back up two characters. Then look and see if the third character from the end is a decimal point. If so, glory be, home free; skip this section and move straight on to line 40!

Before we go any further we should probably back up a little and make sure that all strings have a decimal point somewhere. The only numbers that don't are integers, and we can test for integers and add a decimal to the end of the string using our old friend INT once again:

```
22  IF INT(NUMBER) = NUMBER THEN A$ = A$ + "."
```

The reason for this will soon be clear. If A$ fails the test on line 25, it is because there are too few digits to the right of the decimal. So we can just add a zero to the end of the string and then run the test on line 25 again:

```
30  A$ = A$ + "0":GOTO 25
```

We're almost done. All that remains is to put commas in wherever they are needed. If the number is smaller than one thousand, there will be no commas:

```
40  IF NUMBER < 1000 THEN 50
```

Let's set up a new string equal to all of A$ above the hundreds place. Then let's eliminate all of A$ to the left of that first comma:

```
45  B$ = LEFT$(A$,LEN(A$) - 6): A$ = "," + RIGHT$(A$,6)
```

Now if B$ is less than four digits long we can just tack it onto the beginning of A$ and exit this part of the routine, having inserted (in line 45) the number's one and only comma:

```
46  IF LEN(B$) < 4 THEN A$ = B$ + A$:GOTO 50
```

However, if B$ is longer than that, we are going to have to add more commas:

```
47  A$ = "," + RIGHT$(B$,3) + A$: B$ =
    LEFT$(B$,LEN(B$)-3):GOTO 46
```

What we have done is to add another comma to A$ followed by the right three digits of B$. We then knocked three digits off the right end of B$ and returned to the test in line 46 to see if B$ was now less than four digits long. The program will stay in this loop, gradually taking digits off B$ and adding them back, with commas, to A$ until the full number has been reassembled as A$.

Finally, let's add two lines to permit us to see our results and do it again and again:

# The Program Innovators

## Games

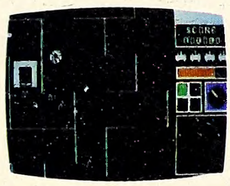
## Education


ODYSSEY

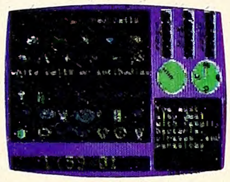

ATLANTIS


CRISIS MOUNTAIN


NIGHTMARE GALLERY



BOLO

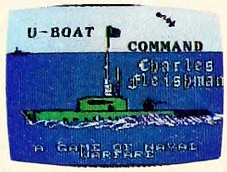


MICROBE


PROCYON WARRIOR


CAMPAIGN TRILOGY

ESCAPE FROM ARCTURUS



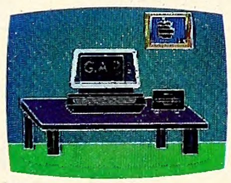

U-BOAT COMMAND

## Business

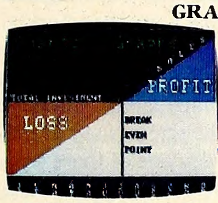
## Utility

THE LINGUIST

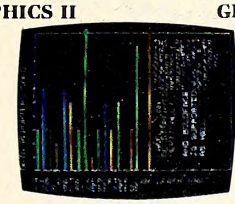
HIGHER GRAPHICS II



G.A.P. GRAPHICS




HIGHER TEXT II



DATA REPORTER

## # 1 Producer of Innovative, Ground Breaking Software

The people at Synergistic Software take pride in producing software with maximum resolution and versatility for business and entertainment. Adventure and arcade games offer technical innovation as well as imaginative plot lines for outstanding action and stimulating adventure. Our uniquely flexible business programs can maximize your productivity. Software is available for the Apple, Atari & IBM PC Computers. Find out why the August issue of Softalk Fastalk showed Synergistic Software as the No. 1 producer of "innovative, new ground" software for the Apple II.

Synergistic Software

Synergistic Software, 830 N. Riverside Drive, Suite 201, Renton, WA 98055, (206) 226-3216

```
60  PRINT A$
70  GOTO 10
```

And one final warning. Don't enter numbers over one billion. The Apple stores very large numbers in exponential form (and doesn't keep very good track of more than eight digits either), so you can expect some errors when using very large numbers.

**Getting Personal.** Some of that was pretty heavy sledding. The second project is a good deal easier. Perhaps you have suffered from the recent outpouring of computerized "personal" letters trying to sell you subscriptions to magazines other than *Softalk*, or real estate, or used cars. The thing that makes these letters particularly irritating is their familiar tone: the way they call you by your first name, talk about the benefits they are about to bestow on the entire "Carlston" family, and comment on how all your neighbors in "San Rafael" are buying their used cars by mail too.

Well, the only way to fight fire is with fire, so now we are about to impart to you the secret of how to personalize your own junk mail. There isn't much to it really. Suppose that you want to type in a list of names and addresses. For the envelope you need to use the full name of the party in question. In the greeting, you would like to use just the first name, and in the text you want to refer to the family, requiring the use of the last name.

Separating out the first and last names from a string containing a person's entire name is not particularly difficult. You merely have to search the string for spaces. You can usually assume that everything before the first space is a person's first name and everything after the last space is the last name (although even the best made plans sometimes go awry—that's why you occasionally see form letters addressed to "Mr. Jr.").

Here's a little routine that separates Fullname$ into First$ and Last$:

```
10  INPUT "ENTER FULL NAME: ";FULLNAME$:L =
    LEN(FULLNAME$)
20  FOR X = 1 TO L
30  IF MID$(FULLNAME$,X,1) = " " THEN Y = X: X = L
40  NEXT X
50  FIRST$ = LEFT$(FULLNAME$,Y-1)
60  FOR X = L TO 1 STEP -1
70  IF MID$(FULLNAME$,X,1) = " " THEN Y = X: X = 1
80  NEXT X
90  LAST$ = RIGHT$(FULLNAME$,L-Y)
100  PRINT "THE FIRST NAME IS ";FIRST$;" AND THE LAST
     NAME IS ";LAST$
110  GOTO 10
```

Take a close look at the use of the for-next loop in line 60 for a moment. Usually the value of the counter in a for-next loop increases by one each time you go through the loop. In other words, it steps the counter by one each time through. However, by addition of the *step* command at the end of the for command, one can change the step to any other value. In this case we wanted to work from the back of the string to the beginning. In other words, we wanted to count backward by one until we found our first space. Therefore, we used step −1.

**Scrolling Along.** Our third routine is fun. Load a message into the string A$, jump to this subroutine, and watch the message go rippling across the screen. Since scrolling messages are sometimes hard to read, this routine is designed so that pressing any key will stop the scroll. Another press will start it up again.

```
10  INPUT "MESSAGE: ";A$
20  GOSUB 100: GOTO 10
100  G$ = "          ":G$ = G$ + G$:G$ = G$ + G$ + A$:J =
     LEN (A$):A$ = " ": POKE - 16368,0: FOR P = 1 TO J: VTAB
     23: HTAB 1: PRINT MID$ (G$,P,40);:O = PEEK (-16336): IF
     PEEK (-16384) > 127 THEN POKE - 16368,0: GOSUB 110
101  NEXT :G$ = RIGHT$ (G$,40) + LEFT$ (G$,40): FOR P = 1 TO
     41: VTAB 23: HTAB 1: PRINT MID$ (G$,P,40);:O = PEEK
     ( - 16336): IF PEEK ( - 16384) > 127 THEN POKE
     - 16368,0: GOSUB 110
102  NEXT :G$ = " ": RETURN
110  IF PEEK ( - 16384) < 128 THEN 110
111  POKE - 16368,0: RETURN
```

That routine was not exactly written for maximum readability. However, this is more or less how it works. The string G$ is the display string. At first it is filled with ten blanks. Then A$ is tacked on to the end of it. Forty-character-long chunks of G$ are then displayed on the screen. Each time through the for-next loop, the starting point of the forty characters is incremented by one. In other words, the first time through, characters one through forty are displayed. The second time through, characters two through forty-one are displayed, and so forth. The peek (−16336) is just a reference to the speaker location in the Apple. Peeking it causes a clicking sound (peeking it very quickly can generate music, but that's another story).

**Branching Out.** Our final project is much less amusing but very useful. Think back to last month's project. We wrote a program to create shape tables. There were many keys we used to move our cursor about the screen, load in shapes, or save them to disk or cassette. We tested for each one of those keys with an if statement. Simple and effective but not very economical—and very wasteful indeed of computer memory.

Suppose that we had created a string of all of the keys we used to control the program, like this: *A$ = "IJKMCSRXY"* (you can even include control characters in such a list, although you won't be able to see them). Then the use of a command called the *computed goto* statement permits a very elegant branch. It looks like this:

```
100  GET A$: FOR I = 1 TO 9: IF A$ = MID$("IJKMCSRXY",I,1)
     THEN J = I:I = 9: NEXT : ON J GOTO
     1000,2000,3000,4000,5000,6000,7000,8000,9000
110  NEXT : GOTO 100
```

If J equals one, then the program branches to the first line number listed in the list after the goto statement. If J is two, then program control shifts to the second line number listed (2000 in the example above) and so forth. And the best thing is that, if the key pressed is not one of those in the string, the program immediately returns to line 100 and waits for you to try another key.

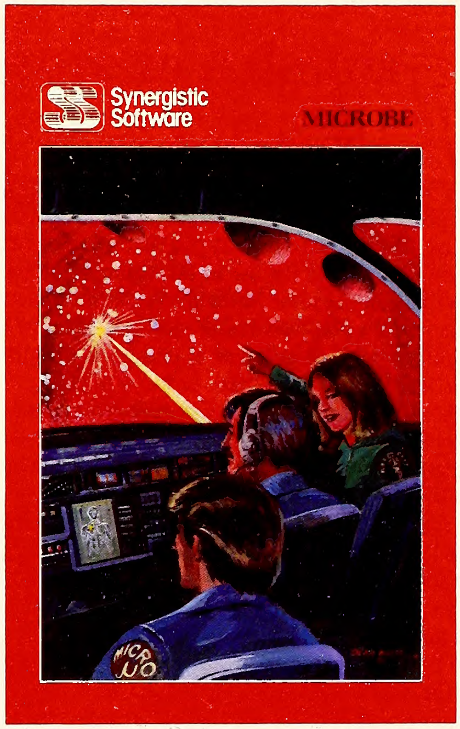
That's enough for this month. Your homework assignment is to use the commands illustrated in this month's column to design a routine that creates random but pronounceable names. Another truly useful routine for your library of tools.

# An Innovative Breakthrough

# MICROBE:
## The Anatomical Adventure
By Robert Clardy

## Your Fragile Submarine, Lost in an Awesomely Beautiful, Yet Terrifying World.
## THE HUMAN BODY!






Disguised as a challenging and exciting adventure/arcade game, Microbe is, in reality, a gold mine of instructional material covering:

- HEALTH AND SAFETY TOPICS
- HUMAN ANATOMY AND IMMUNOLOGY
- DISEASE AND DIAGNOSTICS
- DRUG AND MEDICAL PROCEDURES
- AND MUCH MORE

Microbe is the first of a new breed of games, designed and produced by a team of professional designers, physicians, educators and programmers. This diverse group has produced a unique product that presents an immense amount of information in a thoroughly entertaining format suitable for individual or group use, for children, pre-meds or physicians.

Your challenge begins with you in command of the crew of the submarine, Microbe, which has been shrunk to microscopic size and injected into the human patient. You must use your cunning, knowledge and dexterity to save your patient.

Perhaps a blood clot in the brain needs to be removed. You navigate quickly but cautiously toward the brain as you fend off attacking viruses, white blood cells and other body defenses. Escape the labyrinth of passages that make up the lung or use your medical knowledge to tame the powerful human heart.

Always different, always a challenge! Microbe uses the full capabilities of the Apple to provide the best adventure, arcade, strategic, and simulation game on the market. Game variation options change the skill level (speed of play, number and severity of attacks, etc.). The background (game players through physicians), the number of players (1, 2 or group play), and the emphasis (health/safety vs. medicine). Microbe even talks with VOTRAX™ and ECHO™ speech boards.

Enjoy your next viral infection, with Microbe: The Anatomical Adventure!

Requires a 48K Apple II or II plus DOS 3.3 . **$44.95**

Synergistic Software

Synergistic Software
830 N. Riverside Drive
Suite 201, Renton, WA 98055
(206) 226-3216

**ORDER ONLY 1-800-426-6505**

WE ACCEPT CHECKS, MASTERCHARGE, VISA AND C.O.D. ORDERS. $2.00 HANDLING FEE EXCEPT ON PREPAIDS.

# GRAPHICALLY SPEAKING

## by Mark Pelczarski

First, just a note to remind you that we are looking for graphics creations on your Apple for use in this column. Along with the usual how-to's, we'll be doing some issues about all the different ways people are using the Apple's graphics. Please send any disks, printouts of artwork, or descriptions of your graphics applications to Softalk Gallery, Box 60, North Hollywood, CA 91603.

We've talked about using graphics from Basic. However, the speed limit on Applesoft requires the use of machine language for animated graphics. The most difficult part about machine language is learning to think in hexadecimal, or base sixteen. Base sixteen is a kind of shorthand for binary, the number system that your computer actually "understands." The best way to handle hexadecimal is *not* to convert to base ten and back unless it's really necessary. Most of the time it's not. The only time you'll have to convert is when you are trying to reference your machine language routines or addresses from Basic.

In base ten (decimal), you count 1,2,3,4,5,6,7,8,9,10,11,.... In base sixteen (hexadecimal), you count 1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,10,11,.... The numeral A is really what we call ten, B is eleven, C is twelve, and so on. F is fifteen, after which comes 10, or sixteen (it's base sixteen, remember?).

To keep things straight, when referencing hexadecimal numbers, we'll precede the number with a dollar sign (Why? Because everyone else does). So 2000 is the good old number two thousand you all know, while $2000 does not mean two thousand dollars (drat); it means two thousand hexadecimal, which is really 8192. "Slow down!" you say. . . . Okay. Look at table 1. It shows numbers in binary, hexadecimal, and decimal. Notice how conveniently the numbers $0 through $F correspond to all combinations of binary numbers that can fit in four bits. Remember also how conveniently a byte, the basic storage unit, has eight bits. That means that two hexadecimal digits can give you all the possible values for one byte ($00 through $FF). That's why they picked base sixteen as such a convenient shorthand.

The largest hexadecimal number you'll be dealing with is $FFFF. It's a four digit number because the Apple and other similar computers use two bytes to store addresses, $FF and $FF, in this case. The number $FFFF in decimal is 65535, which gives the 64K that you hear about as the Apple's maximum memory size. One K is approximately 1,000 bytes; it's actually 1,024, or two to the tenth power. To understand the process of converting hexadecimal numbers to decimal, see figure 1 as an example. The rightmost of the four places is the ones column, the next is the sixteens column, the next column over is for sixteen to the second power, or 256, and the leftmost of the four places is for sixteen to the third, or 4,096.

| Binary | Hexadecimal | Decimal |
|---|---|---|
| 0000 | $0 | 0 |
| 0001 | $1 | 1 |
| 0010 | $2 | 2 |
| 0011 | $3 | 3 |
| 0100 | $4 | 4 |
| 0101 | $5 | 5 |
| 0110 | $6 | 6 |
| 0111 | $7 | 7 |
| 1000 | $8 | 8 |
| 1001 | $9 | 9 |
| 1010 | $A | 10 |
| 1011 | $B | 11 |
| 1100 | $C | 12 |
| 1101 | $D | 13 |
| 1110 | $E | 14 |
| 1111 | $F | 15 |
| 10000 | $10 | 16 |

Table 1.

```
$25C7 = $2*4096 + $5*256 + $C*16 + $7*1
      = 2*4096 + 5*256 + 12*16 + 7*1
      = 8192 + 1280 + 192 + 7
      = 9671
```

Figure 1. Converting hexadecimal to decimal.

**Machine Language Coding.** The first thing we'll do with machine language is to take the program in listing 4 from last month and convert the graphics part. In case you don't have it on hand, it's included this month as listing 1. The program first creates a Y look-up table (lines 150–250), then cycles through a loop, putting bytes with all the dots set onto the screen.

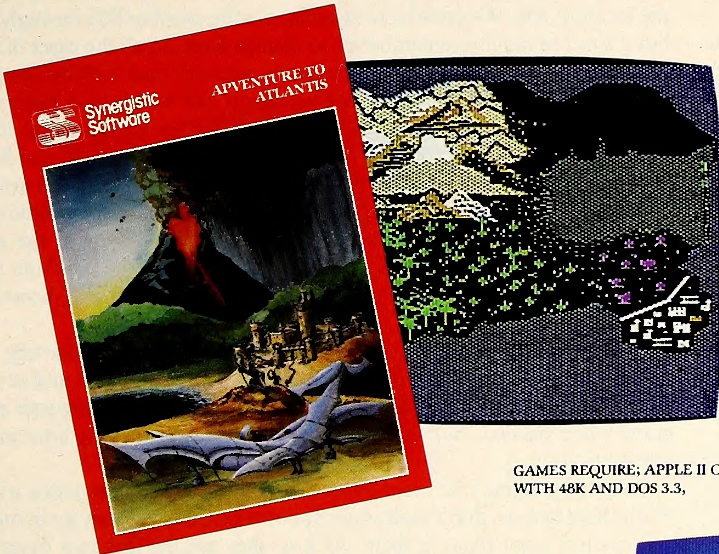
Our first task will be to create a permanent Y look-up table for our machine language routines. There's no need to recompute it all the time; once we have it we can bload it into any program that needs it. To make it easy, we'll use a Basic program to do the computations and poke the values we want into memory. Then we'll save that portion of memory to disk, and we've got it whenever it's needed.

We'll put the look-up table just above the hi-res page 1 screen memory. Page 1 is located from addresses 8192 ($2000) to 16383 ($3FFF). That means our table should start at 16384, or $4000. For reasons that will become apparent at some later date, we'll store the table in two sections; first the high bytes of all the addresses, then the low bytes. By high

```
 10  HGR
 12  GOSUB 150
 15  X = 0:Y = 0:XC = 1:YC = 1
 20  L = YT(Y) + X
 60  POKE L,255
 70  X = X + XC: IF X < 1 OR X > 38 THEN XC = - XC
 80  Y = Y + YC: IF Y < 1 OR Y > 190 THEN YC = - YC
 90  GOTO 20
140  REM THIS SUBROUTINE CREATES A Y-LOOKUP TABLE, YT.
150  DIM YT(191)
160  FOR Y = 0 TO 191
200  Y1 = INT (Y / 8):YR = Y - Y1 * 8
210  Y2 = INT (Y1 / 8):YS = Y1 - Y2 * 8
220  YL = 8192 +Y2 * 40 + YS * 128 + YR * 1024
230  YT(Y) = YL
240  NEXT Y
250  RETURN
```

Listing 1.

# Adventure Classics . . .




## Apventure to Atlantis

The forces of sorcery are gathering to repulse the marauding scientific Atlantean kingdom. The epic battle can end either in the enslavement of the world or the absolute destruction of Atlantis and the evil it represents. Combines hundreds of high-res color pictures with arcade-like animation and sound effects. Apventure to Atlantis has the best of arcade and adventure games combined into the ultimate apventure . . . . . . . . . . . . . . . . . . **$39.95**

GAMES REQUIRE; APPLE II OR II PLUS, WITH 48K AND DOS 3.3,


## Odyssey: The Compleat Apventure*

Embark on a heroic quest across the dreaded Sargalo Sea to explore islands, caverns and castles. Gather the forces and weapons needed to defeat the cruel Caliph of Lapour. Three interlocking programs using several hi-res and lo-res maps provide an unending variety of opponents and challenges. Specify Integer or Applesoft . . **$29.95**


## Campaign Trilogy

Three challenging adventure games on one disk.
**Dungeon Campaign:** explore complex maze-like dungeons in search of gold, jewels, and artifacts.
**Wilderness Campaign:** *A high-res crusade, you equip an army and challenge the evil necromancer.
**Sorcerer's Challenge:** A two player competition between two powerful magicians . . . . . . . **$29.95**

*Designated "Classics" by Softalk Magazine

## THE PROGRAM INNOVATORS

Synergistic Software

Synergistic Software
830 N. Riverside Drive
Suite 201, Renton, WA 98055

**ORDER ONLY 1-800-426-6505**

WE ACCEPT CHECKS, MASTERCHARGE, VISA AND C.O.D. ORDERS. $2.00 HANDLING FEE EXCEPT ON PREPAIDS.

Apple and Apple Dos are trademarks of Apple Computer.

```
160   FOR Y = 0 TO 191
200   Y1 = INT (Y / 8):YR = Y - Y1 * 8
210   Y2 = INT (Y1 / 8):YS = Y1 - Y2 * 8
220   YL = 8192 + Y2 * 40 + YS * 128 + YR * 1024
230   POKE 16384 + Y, INT (YL / 256)
235   POKE 16576 + Y,YL - INT (YL / 256) * 256
236   REM THE RIGHT HALF OF LINE 235 HAS THE FORMULA FOR
      FINDING THE REMAINDER OF THE DIVISION YL/256
240   NEXT Y
```

Listing 2.

and low bytes, we mean that for an address such as $4F8A, the high byte is the left half, or $4F, and the low byte is $8A, the right half. Since there are 192 Y values on the screen, the first part of the table, the high bytes, will take 192 bytes, from 16384 to 16575 ($4000–$40BF). The second half, the low bytes of the 192 addresses, will be put in 16576 to 16767 ($40C0–$417F).

The high byte of a value is computed from decimal by dividing by 256 and chopping off the remainder. The low byte is the remainder of that same division. The program in listing 2 does all the same computations for the look-up table that we did last month, but then splits the address into high and low bytes and pokes those into memory. Type it in, then run it.

After you run the program, your look-up table is in memory. Now you want to save it to disk. Type:

```
BSAVE LOOKUP,A16384,L384
```

or, alternately,

```
BSAVE LOOKUP,A$4000,L$180
```

The A means starting at which address, and the L means length. As shown, you can define these parameters with either decimal or hexadecimal numbers. Bsave means binary save.

You may want to bsave the file the same way on a few disks. We'll be using this look-up table a lot, and if you wind up doing much graphics programming in machine language, you will probably use it frequently.

Now for a short introduction to assembly language. Did we say machine language before? Well, assembly language is almost the same as machine language, except that machine language is just numbers (hexadecimal ones, at that). Assembly language corresponds one-to-one with those machine language numbers, but its commands are mnemonics instead. In other words, you use letters that mean something to you instead of the numbers that the machine understands. To convert your assembly language mnemonics to machine language numbers, you need an assembler. An assembler is a program that interprets the assembly language instructions you write and pokes the corresponding numbers into memory for the machine to read. Two assemblers that we know will work with the listings we've included are the *DOS Tool Kit* assembler from Apple and *Merlin* from Southwestern Data Systems. Some of the necessary mnemonics may be different on other assemblers. Of course, if you don't have an assembler and want to use the routines in these articles, you could just enter the finished numbers in the listings, but that's dull and boring and you really wouldn't learn much about writing in assembly/machine language.

This first little machine language routine (listing 3) takes an X and Y

```
 1            ORG   $6000
 2  TEMPLO    EQU   $06
 3  TEMPHI    EQU   $07
 4  LOOKHI    EQU   $4000
 5  LOOKLO    EQU   $40C0
 6  XVALUE    DFB   0
 7  YVALUE    DFB   0
 8  START     LDY   YVALUE
 9            LDA   LOOKLO,Y
10            STA   TEMPLO
11            LDA   LOOKHI,Y
12            STA   TEMPHI
13            LDA   #$FF
14            LDY   XVALUE
15            STA   (TEMPLO),Y
16            RTS
```

Listing 3.

value that you give it, finds the Y look-up value in the look-up table, then puts the value $FF (255) at the corresponding byte on the screen. Not much, but it does illustrate a few of the instructions and addressing methods used in machine language. Going through listing 3 line by line:

The first line (ORG $6000) says that the machine language routine will originate (ORG) at address $6000 (24576). The assembler will start putting our instructions at that address in memory.

The next four lines are equate commands (EQU), or label definitions. The first says that whenever we use TEMPLO, we'll really mean the location $06. We could just as easily use the number $06 throughout, but it's not as easy to remember or to change later. In all the cases in this example, the equates refer to addresses in memory that we'll be using in our program for storing things. We use them like variables in Basic, except we tell the computer exactly where in memory these storage locations should be. TEMPLO and TEMPHI refer to addresses $6 and $7 in your computer. The first 256 bytes (addresses $0000 to $00FF) are referred to as zero page, and can be accessed much faster and do some special things that other memory addresses cannot do. Most of the zero page is used by Applesoft and DOS (both of which are machine language programs, in reality). Addresses $6 through $9 are free, however, so we'll use two of those.

LOOKHI and LOOKLO are also equated; they are the starting addresses of the two parts of our look-up table that we created and saved. Again, we could just use the addresses in our assembly language program when needed, but the labels allow us to give them a little more meaning.

Next, we define two more bytes of storage (DFB means define byte). With these two we don't really care about where exactly they go in memory; we just want them in there. As it is, they are the first two bytes actually set aside by our assembly language program, so they'll be put at $6000 and $6001 (the ORG told the assembler to start at $6000, and the equates don't really count, since we were just telling it that a label means a particular number). We defined the bytes as having values of zero, although that doesn't matter, because it's in these locations that we'll poke our X and Y values.

Now the program starts. The first instruction (LDY YVALUE) is to load the Y register with the number in YVALUE. The Apple has three main registers, which are single bytes set aside inside the 6502 microprocessor. These are accessed very quickly, and most instructions center around the use of these bytes (and one other). The registers are labeled A, X, and Y. A is the accumulator, where most everything happens, including all mathematics and logic operations. The X and Y registers are used mostly as pointers, offsets, and counters. LDY YVALUE means to load the Y register with the number in the memory location represented by YVALUE. We'll be using this number as an offset, much as you would use the parameters of an array in Basic. In fact, we'll use it as an offset in our look-up table, just as we did in the Basic examples with look-up arrays.

The next line says to load the accumulator with the value in the address LOOKLO, offset by Y. Given the equate command in line 5, that means that it will load the accumulator with the value in address $40C0+Y. Exactly like using an array in Basic!

That is followed by STA TEMPLO, or store A in TEMPLO. This instruction takes the value that we just loaded into the accumulator from the look-up table and puts it in address TEMPLO ($06). Note that in assembly/machine language, you can't really say "take something from here in memory and put it over there" in one command. You have to load it into one of the registers from a memory location, then store the contents of that register in another memory location.

The next two instructions duplicate the load and store commands for the high byte from the table. That was easy. . . .

Next, we load the accumulator (LDA) with the number $FF (255). Note that the # sign means to use the number following it. If the # were left off, the instruction would mean "load A with whatever is in address $FF." A frequent mistake of beginners and experts alike is to leave off the # and wonder why the program does such strange things. Just when you're sure that your computer's broken or that you've found a bug in the ROM, you discover the missing # that changed the entire meaning of things. Grrrr. . . . You have to tell it everything!

Now for some tricky maneuvering. As we mentioned, zero-page ad-

```
SOURCE FILE: ML#1
----- NEXT OBJECT FILE NAME IS ML1
6000:                      ORG   $6000
0006:             TEMPLO   EQU   $06
0007:             TEMPHI   EQU   $07
4000:             LOOKHI   EQU   $4000
40C0:             LOOKLO   EQU   $40C0
6000:00           XVALUE   DFB   0
6001:00           YVALUE   DFB   0
6002:AC 01 60     START    LDY   YVALUE
6005:B9 C0 40              LDA   LOOKLO,Y
6008:85 06                 STA   TEMPLO
600A:B9 00 40              LDA   LOOKHI,Y
600D:85 07                 STA   TEMPHI
600F:A9 FF                 LDA   #$FF
6011:AC 00 60              LDY   XVALUE
6014:91 06                 STA   (TEMPLO),Y
6016:60                    RTS
*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

Listing 4.

dresses have some special functions that can do things other addresses can't. One is called indirect addressing. The next two instructions, LDY XVALUE (put XVALUE in the Y register) and STA (TEMPLO),Y, put the value $FF from the accumulator onto the hi-res screen at the address we want. The previous loads and stores put the base address of the screen line from our look-up table into TEMPLO and TEMPHI. Note that the low byte of the address went in the first location. STA (TEMPLO),Y says to store the contents of A in the address contained in TEMPLO and its following byte (TEMPHI), offset by Y. In other words, take the address stored in TEMPLO and TEMPHI, add the value in Y, and store the contents of the accumulator in the resulting location. In our particular application, this takes the base address of the line that we stored in TEMPLO and TEMPHI, adds the X value, which was loaded into the Y register, and stores the number $FF in the resulting location.

Confusing perhaps, but that's about as tricky as machine language addressing gets. If you can handle that, you've got a good start on understanding how machine language works.

The last line is a return from subroutine (RTS), which is the equivalent of a Basic return statement. It means to go back to the instruction from whence it was called.

Listing 4 shows the same program after it has gone through the assembly process. After assembling, you are shown the addresses of each assembled instruction and the actual hexadecimal values to which those instructions were converted. If you don't have an assembler, you can enter this short routine by hand with the commands in listing 5.

Finally, listing 6 is a Basic program that bloads the look-up table and machine language program, then loops through, pokes the X and Y values into locations 24576 ($6000) and 24577 ($6001), and calls the subroutine at 24578 ($6002). Notice the similarities and differences between listing 6 and 1. The new version may not seem much faster than the old, since most of the work is still being done in Basic, and the machine language routine is very short with no repetition. But as the tasks become slightly more complex, the speed differences in machine language are awesome, as you'll soon find out for yourself.

The bracket ( ] ) and asterisk (*) characters at the beginning of each line are prompts. You type the rest. After each line, press return.

```
]CALL -151
*6000:00  00  AC  01  60  B9  C0  40
*6008:85  06  B9  00  40  85  07  A9
*6010:FF  AC  00  60  91  06  60
*3D0G
]BSAVE PLOT,A24576,L23
```

Listing 5. Entering the machine language routine directly.

```
5   PRINT CHR$ (4);"BLOAD LOOKUP"
6   PRINT CHR$ (4);"BLOAD PLOT"
10  HGR
15  X = 0:Y = 0:XC = 1:YC = 1
20  POKE 24576,X: POKE 24577,Y: CALL 24578
70  X = X + XC: IF X  <1 OR X > 38 THEN XC = - XC
80  Y = Y + YC: IF Y  < 1 OR Y > 190 THEN YC = - YC
90  GOTO 20
```

Listing 6.


PLATO
WIDENS
YOUR
CHILD'S
WORLD.
In public school systems across the United States, PLATO® educational software has proved that it stimulates and speeds learning. Now, for the first time, Control Data has put PLATO lessons on microcomputer disks, so your child can learn at home—and enjoy it.
For information on PLATO educational software, or to order, call toll free 800/233-3784. In California, call 800/233-3785, or write Control Data Publishing Co., P.O. Box 261127, San Diego, CA 92126.

# WIDEN YOUR CHILD'S WORLD WITH THIS INTRODUCTORY OFFERING.

## 10 DAY MONEY BACK GUARANTEE.

Now, until May 31, 1983, when ordering a lesson at $45.00, you may order additional lessons for just $35.00 each. (Back-up disk is included with each lesson at no additional charge.)

Mail this form, or call toll-free 800/233-3784. (In California, call 800/233-3785.)

**BEFORE ORDERING: CHECK TO MAKE SURE YOUR EQUIPMENT MEETS THESE REQUIREMENTS.**

| | Apple II PLUS | Atari 800 | TI 99/4A |
|---|---|---|---|
| Memory: | 48K | 48K | 32K Memory Expansion |
| No. of drives required: | 1 disk and controller | 1 disk and controller | 1 disk and controller |
| Operating System: | D.O.S. 3.3 | D.O.S. 2 | PLATO Interpreter Cartridge* |

Will display on any color or b/w monitor or TV compatible with your microcomputer listed above.

*PLATO Interpreter Cartridge may be ordered directly with the order form below. Only one per TI 99/4A machine is needed.

CONTROL DATA PUBLISHING CO., P.O. Box 261127, San Diego, CA 92126

SHIP TO: (Please Print)

Name ______________________________

Address ______________________________

City ______________ State ________ Zip ________

☐ Payment enclosed (Check or money order only. Make payable to Control Data Publishing Co.).

☐ Mastercharge ☐ VISA ☐ American Express ☐ Diners Club

Card Number |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| Expiration Date ________

Your signature ______________________________

Telephone No. (        ) ______________________________

**Please complete the following:**

| Quantity | Title |
|---|---|
| | Basic Number Facts |
| | Whole Numbers |
| | Decimals |
| | Fractions |
| | Physics: Elementary Mechanics |
| | French Vocabulary Builder |
| | German Vocabulary Builder |
| | Spanish Vocabulary Builder |
| | Computer Literacy: Introduction |

Apple® is a trademark of Apple Computer, Inc. Atari® is a registered trademark of Atari, Inc. TI® is a registered trademark of Texas Instruments, Inc.

Warranty available free from the Control Data Publishing Co., 4455 Eastgate Mall, San Diego, CA 92121.

**For use with:** (check one)

☐ **Apple II Plus** Available: Nov. '82

☐ **Atari 800** Available: Dec. '82

☐ **TI 99/4A** Available: Jan. '83

Allow 3-5 weeks for delivery.

**INTRODUCTORY PRICES:**

One lesson $45.00. Each additional lesson $35.00.

☐ Include PLATO Interpreter Cartridge for my TI 99/4A. Price $50.00.

Price for above order: . . . . . $________

Calif. Residents: add 6% sales tax . . . . . . . . . . . . . . . $________

Add $2 shipping and handling . . . . . . . . . . . . . . . $ 2.00

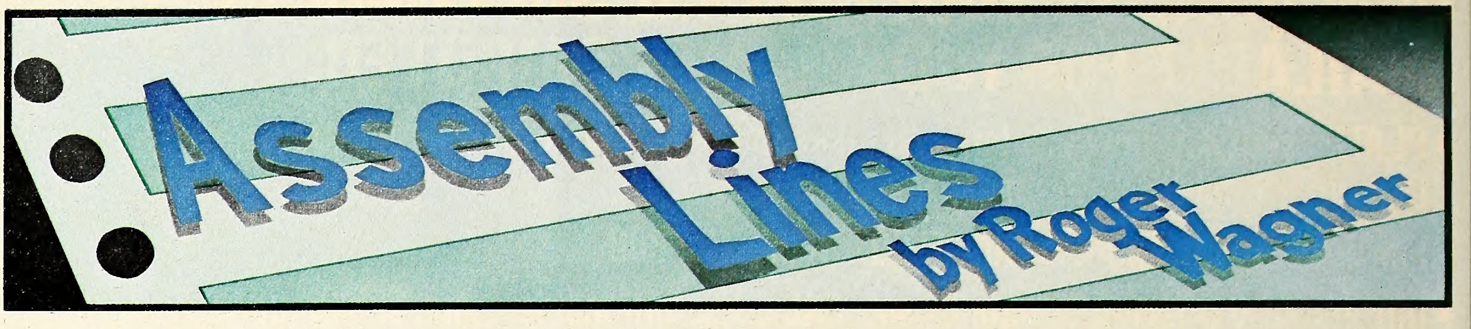
TOTAL PRICE: . . . . . . . . . . . $________

☐ Please send information on ______________________ lesson(s).

If not completely satisfied, you may return all lesson material within 10 days of receipt for a refund. All orders subject to acceptance. Offering expires May 31, 1983.

Control Data Publishing Company supplies this software under a personal license agreement rather than for sale. Customers obtain the right to possess and use PLATO software by paying the prices specified and agreeing to the terms and conditions of the license agreement contained in the package.

ST-11




## Everyone's Guide to Assembly Language, Part 26

Earlier this year in the January and February columns we discussed how Applesoft variable data could be passed from Basic to machine language and back again. The rationale was that in many cases a program created by combining Applesoft and machine language is an effective approach to a problem. Successive issues on hi-res graphics included these techniques so as to have a convenient way of experimenting with the various routines.

It is *highly* recommended that you review the appropriate issues if you're not entirely familiar with the nature of Applesoft variable storage. This article will not extensively review the material presented there. Pages 127 and 137 in the *Applesoft Reference Manual* also provide very valuable information well worth referring to in the course of reading the material presented here.

For the most part, however, all the past discussions were limited to dealing with two-byte integer data. That is to say, the possibility of dealing with true floating point data was not considered. In many cases, integer values from 0 to 65,535 or −32,767 to 32,767 are more than adequate for our purposes, as was the case when passing tone routines or X and Y coordinates for plotting. There are times, though, when greater precision, or fractional values, are required.



Dealing with floating point numbers from a pure machine language program is a fairly complex topic, and our intent here is not to explain completely the inner workings of floating point operations. Rather, let's explore the options made available by taking advantage of the existing routines already present in the Applesoft Basic interpreter. These can generally be considered to be always present in the background of an operating machine language program.

For those of you who hope to speed up floating point operations in Applesoft, writing your own routines may not be that effective. This is because the routines in Applesoft are already written in machine language. We can, however, gain important speed improvements just by calling the routines directly. This is because we can eliminate the normal process of interpreting Basic statements that would otherwise occur in Applesoft. This is what the currently available compilers do, and we can expect similar speed improvements to a Basic program by using routines directly from machine language (two to five times faster than straight Applesoft).

**Internalization of Data: Integer versus Real Variables.** The first step in our inquiry is to investigate how Applesoft stores numeric data and to look at the differences in how integer variables and real variables are stored.

Start by initializing your Apple's memory with an FP statement. Then enter:

```
A% = 10:A=10
```

The result is that two variables and their values have been set up in memory. Now to find them!

Enter the Monitor with the usual *call−151*. Then enter:

```
67  68  AF  B0
```

and press return. You should get:

```
67- 01
68- 08
AF- 03
B0- 08
```

You may recall from a previous discussion (January 1982) that these four memory locations (67,68 and AF,B0) are used to store the beginning and the end of the current Applesoft program. We can see from the display that the program resides from $801 to $803. A very short program, indeed, but that's understandable since we haven't entered any program lines.

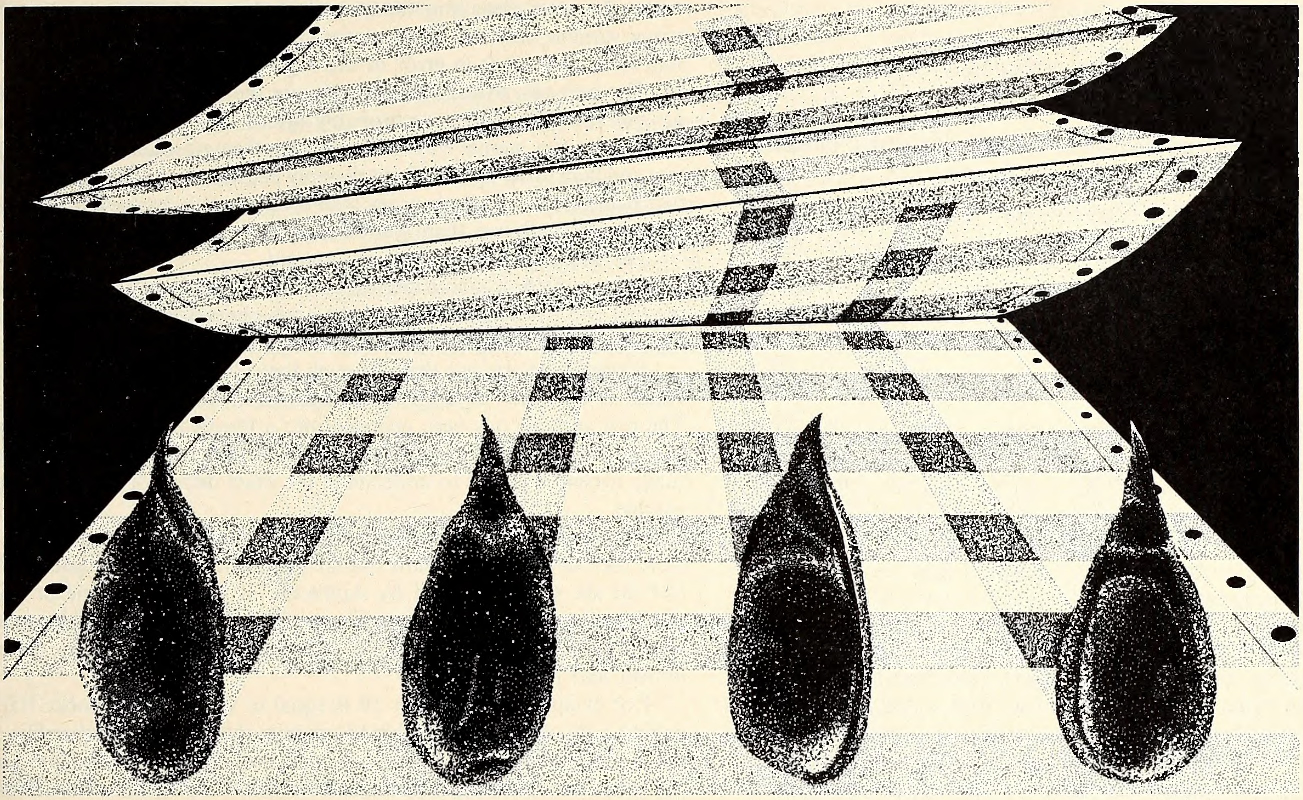
Now let's examine the pointer at $69,6A and $6B,6C. Do this by typing:

```
69.6C
```

and pressing return. You should get:

```
69-  03  08  10  08
```

# FOUR TO GROW



Four software packages, that is, to make your Apple® computer really grow! Versatility is the key to computing success, and with Advanced Operating Systems, it's as easy as one, two, three.

**APPLE-AIDS** is a collection of 12 Applesoft and machine language programs designed to bring you greater flexibility and control over your microcomputer.

In addition to several programs which allow you to examine and change various track/sector contents, format new disks, and create and edit EXEC files, APPLE-AIDS includes two programs you can't afford to be without. UNDELETE FILES allows you to recover accidentally deleted files which you thought were lost forever. KILL DOS removes DOS from any disk freeing up more space for your programs or data.

Russ Adams, a reviewer for INFOWORLD, writes, "Its (APPLE-AIDS) documentation is the finest I have seen since joining the INFOWORLD Software Review Board." Ease of use and extensive documentation make APPLE-AIDS a valuable addition to every Apple software library. #26066, $49.95

**HELLO CENTRAL!** changes your Apple II® or Apple II Plus® into a highly versatile communications center.

Establish direct communications with other computers of any make or tap into the wealth of information available from the hundreds of public-base services like THE SOURCE.

HELLO CENTRAL!'s most unique feature is its text buffer. Messages or entire files can be uploaded and downloaded into the text buffer which holds up to 18,000 characters. The buffer text can be scanned, changed, and then saved to disk.

The best thing about HELLO CENTRAL! is that you don't have to be a professional to get professional results. A few simple keystrokes will accomplish any of your text/communications tasks. #26081, $99.00

**MUSIC GAMES** skillfully combines the sound and high resolution graphics of the popular Apple computer to provide both visual and audible reinforcement to the study of music. Twelve different programs, selected from a menu, cover ear training, note recognition and writing, rhythm practice, and listening enjoyment. These versatile games are written for ages 5 through adults. #26116, $39.95

**PEN PAL** gives you the power and versatility of higher-priced word processors in an easy-to-use format. With only 29 commands, you can quickly and easily create lengthy reports or short memos. PEN PAL gives you 40- and 80- column screening, joystick control of the cursor for quick editing, complete printer control, plus both upper and lower case capabilities. PEN PAL is the affordable solution to your word processing needs. #25115, $59.95

Advanced Operating Systems has four good ways to make you and your Apple grow and grow!

Now available at your local software retailer, or call 1-800-428-3696 to order. (Indiana residents, call (317) 298-5566.) MasterCard and VISA accepted.

## ADVANCED OPERATING SYSTEMS

4300 West 62nd Street
P.O. Box 7092
Indianapolis, IN 46206

A Division of
Howard W. Sams & Co., Inc.

This tells us that all simple (that is, nonarray) variables are stored from $803 to $810. Examine this area by typing:

```
803.810 (return)
```

You should get:

```
803-  C1  80  00  0A  00
808-  00  41  00  84  A0  00  00  00
810-  ??
```

You'll recall from our discussions in earlier issues that integer and real variables are stored in the following format:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| INTEGER: | C1 | 80 | 00 | 0A | 00 | 00 | 00 |
| | "A" | " " | "0" | "10" | – | – | – |
| | Name Char1 (neg) | Name Char2 (neg) | High Byte | Low Byte | Unused | Unused | Unused |
| REAL: | 41 | 00 | 84 | A0 | 00 | 00 | 00 |
| | "A" | " " | 1000 0100 | 1100 0000 | 0000 0000 | 0000 0000 | 0000 0000 |
| | Name Char1 (pos) | Name Char2 (pos) | Expo-nent | Mant-issa | Mant-issa | Mant-issa | Mant-issa |

Starting at $803, we find the variable A% stored from $803 to $809. The first two bytes are the name characters. Two bytes are always used. If the variable name is only one character then a null ($00 or $80) is stored in the second position. Note that integer, real, and string variable names are differentiated by the combination of high bit settings in the two-name character bytes. Since only bits 0 through 6 are used for the character (ASCII is only a seven-bit code), bit 7 (the high-order bit) is available for encoding the variable type.

Integer variables always have both high bits set. Real variables always have both high bits clear. String variables always have the first name character clear and the second character set. (The notation for string variable names to the opposite effect on page 137 of the *Applesoft Reference Manual* is in error in this regard.)

The next two bytes, $00 and $0A, are the high and low order bytes for the value 10. You have probably noticed that integer variables are stored in a very simple way, with the value being broken down into the low and high order bytes. About the only peculiar item is the fact that the two bytes are stored high order byte first, which is backward from the way we normally see them paired in most machine language code.

The three remaining bytes are unused.

$80A to $80F is where the real variable A is stored. You can see that the first two bytes again make up the name characters, this time with the high bits set. The remaining bytes make up the value for the variable A.

It should be obvious that although the values of the integer and real variables are stored as equal, the manner in which they are stored is not. The real variable has been encoded into a five-byte sequence, the logic behind which is not readily apparent. Well, don't despair; it is not actually necessary for us to understand the exact details of the conversion routine.

In general, it will suffice to say that an exponential notation is used to store the number. This is how numbers of such large magnitudes (+/− 10^38) are accommodated by Applesoft. If you rouse some of your more ancient high school memories, you'll recall that the basic idea to exponents is that any number can be expressed with two numbers, the exponent and the mantissa.

For example, the number 10 is equal to 10^1. The number 100 is equal to 10^2. It is reasonable to assume, then, that a number like 50 might just happen to be equal to 10^1.5. As it happens, that's not quite right, but the basic idea is there. In fact, 50 is really equal to 10^1.69897 (or thereabout). The 1 part of the number is called the exponent (or occasionally the order of magnitude) of the number. The 69897 is called the mantissa. You may have fond memories of spending pleasant hours in math classes looking through books with look-up tables to find these values for given numbers.

In any event, it's precisely this type of technique that is used to encode the values of real variables. Fortunately for us, it will not be necessary to create our own routines to handle numbers in this format, since a wealth of such routines already exists in Applesoft.

The remainder of this article will concentrate on some brief exercises in passing floating point numbers back and forth between Applesoft and machine language. Then in upcoming articles we'll explore how to perform various mathematical operations once your machine language program has possession of the data.

**The Floating Point Accumulator** (**FAC**). Applesoft has its own internal set of registers that it uses during its various calculations. The most important of these by far is the floating point accumulator. This is usually labeled FAC in source listings that access this register.

The word *register* is used in a slightly different way here than it is when referring to 6502 registers such as the accumulator or the X or Y registers. Because a floating point number is represented by a series of bytes, the FAC occupies the bytes from $9D through $A2.

You may be puzzled as to why the FAC uses six bytes when variable storage uses only five. This is because the FAC uses $A2 as the sign byte to indicate the positive or negative status of the value. When finally encoded, the sign is included in the exponent and mantissa bytes and thus is no longer needed. Floating point numbers in the five-byte format are said to be "packed." The six-byte format is "unpacked." The unpacked format is faster for the calculation operations. The packed format is used to minimize storage space.

In general, whenever any type of calculation is done by Applesoft, the FAC is the primary register used to hold the result. A second register, ARG (for Argument), is used for two-value calculations, such as 1.5 x 17. ARG uses the bytes $A5 through $AA. For the time being, though, we need only concern ourselves with FAC.

**Passing Data from Applesoft to the FAC.** The first area to investigate is how to get a floating point number passed from Applesoft to a machine language routine. The easiest way is by means of the USR function. The USR is a rather neglected part of Applesoft, probably because of the lack of documentation on its nature and applications. A program statement using USR might look something like this:

```
10 X = USR (Y)
```

When this statement is executed, three things happen.

First, the expression or variable within the parentheses is evaluated and the result put in the FAC.

Second, a call to location 10 ($0A) is done. This is equivalent to a *call 10* in Applesoft. There is a three-byte jump instruction at location $0A. It is assumed that the user has inserted the location of an existing machine language routine. For example, the code JMP $300 might be found at $0A. The program would then jump to $300 to execute whatever routine the user might have put there.

Third, when the user routine eventually does an RTS, the contents of the FAC are assigned to the variable to the left of the equal sign.

For example, type in and run this program:

```
10   POKE 10,0
20   Y = 10
30   X = USR(Y)
```

When run, the program should fall into the Monitor. Then type in:

```
9D.A2 (return)
```

You should get:

```
9D-   84  A0  00
A0-   00  00  00
```

This is the same data we saw for the value 10 when examining the real variable storage. Here's what happened: Line 10 set location $0A to a BRK. When the USR function was called, it put the sequence for 10 in the FAC and then called $0A as expected. Since this was a break, we went into the Monitor and could then immediately examine the FAC.

It is not possible to set the FAC from Applesoft and then to verify the status of the FAC by entering the Monitor with the usual *call−151*. Since the FAC will be used in calculating the value of −151, any prior data would be overwritten.

While you're in the Monitor, let's set up $0A for our next experiment. Type in:

```
0A:  4C  00  03  (return)
```

This will set the vector to point at location $300. Now create a trivial program (in this case, an immediate RTS) at $300 by typing:

```
300: 60 (return)
```

Now return to Applesoft and enter and run this program:

```
10   Y = 10
20   X = USR(Y)
30   PRINT X
```

You should get the number 10 printed out. If you consider what we've discussed so far, it should be apparent why. The value 10 held by Y was passed to the FAC by the USR function. When our "routine" at $300 was called, the FAC remained unchanged. Upon return from our routine, the FAC (still equal to 10) was assigned to the variable X.

Although the USR function is a convenient way of passing data, it is rather limited in terms of syntax. If you wanted to pass a number of parameters to a routine, another technique would be required. You may recall from previous issues a routine called FRMNUM (for Formula Numeric evaluator—$DD67) that we used to evaluate variables being passed to machine language routines. After calling FRMNUM, GETADR ($E752—Get Address) was used to convert the number to a two-byte integer LINNUM ($51,$52—Line Number).

Well, since what we want *is* the FAC, we've already got the solution:

```
                 1  ******************************************
                 2  *                                        *
                 3  *             BASIC - > FAC              *
                 4  *                                        *
                 5  *          SYNTAX: CALL 768,Y            *
                 6  ******************************************
                 7  *
                 8               OBJ   $300
                 9               ORG   $300
                10  *
                11  CHKCOM     EQU   $DEBE
                12  FRMNUM     EQU   $DD67
                13  *
0300:  20 BE DE 14  ENTRY      JSR   CHKCOM
0303:  20 67 DD 15             JSR   FRMNUM
0306:  00       16             BRK
```

This code should be assembled at $300 and called with the following Applesoft program:

```
10  Y = 10
20  CALL 768,Y
```

When this program is run, you should fall into the Monitor. Then type in:

```
9D.A2 (return)
```

You should get:

```
9D-  84  20  00
A0-  00  00  00
```

This should verify that the FAC was properly loaded with the value 10.

In reviewing the listing, you'll see that line 14 calls CHKCOM ($DEBE—Check for Comma) to advance Applesoft's TXTPTR ($B8,B9—Textpointer) past the comma following the 768. Line 15 then calls FRMNUM, which evaluates the variable or expression following the comma and puts the result in the FAC. Line 16 then does the BRK to leave us in the Monitor, from which we can check the FAC to verify that the correct value has been stored.

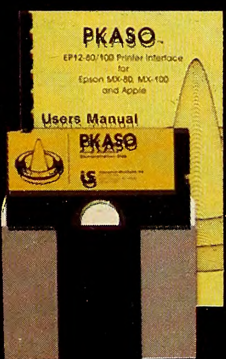
We have now, then, two techniques for passing data from Applesoft to the FAC. The first is to use the USR function (being sure, of course, to set up the vector at $0A). The second is to use FRMNUM ($DD67) to evaluate the expression or variable as part of a parameter list following a call statement.

**Moving FAC to a Memory Location.** Since the FAC is so heavily used, it is sometimes helpful to move the data in the FAC to another location for use later. In Applesoft, this is most often a temporary register or an actual variable. For now, let's see if we can move the data to an arbitrary location.

```
******************************************
*                                        *
*            FAC - > MEMORY              *
*                                        *
*          SYNTAX: CALL 768,Y            *
******************************************
```

# PKASO™
## Printer Interface Family

*A Masterpiece*




PKASO Interfaces come complete with Cable, Instructional Diskette and Comprehensive Manual.

## The PKASO family makes you and your Apple II or Apple /// a master of text and graphics.

PKASO makes it easy to use the features of your printer—select character sizes, vary line spacing, even print in colors. Simple PKASO commands make these features usable from the keyboard or a program.

PKASO also adds features to your system. Press a few keys and get a snapshot "dump" of the image you see on the screen—text or graphics. Add new characters and symbols that you couldn't print before, using our SuperFont™ system. Add our new PipeLine™ printing buffer and your printer can take its time while you and your Apple move on to the next task. The PipeLine is a modular add-on to the standard PKASO board.

The PKASO interface is designed for Apple II and Apple /// in all the popular configurations. It prints in full color on the IDS Prism Printer, and in striking black on C. Itoh, Centronics, Epson, IDS, NEC, and Okidata matrix printers.

## NEW!

**The PKASO PipeLine™ printing buffer keeps your printer from tying up your computer.**

- 8K to 64K Bytes of Memory.
- Micropressor controlled.
- Compression! Increases capacity to millions.
- Clear buffer and select modes easily.

is

Interactive Structures Inc.
146 Montgomery Avenue
Bala Cynwyd, PA 19004
Telephone: (215) 667-1713

# Introducing better company planning, company-wide, with VisiCalc Advanced Version.

More than 300,000 business professionals have discovered how much help our VisiCalc® program can be in planning, budgeting and forecasting. Now we can help businesses even more.

### VisiCalc Advanced Version does it all

VisiCalc Advanced Version makes it easy to extend the problem-solving power, speed and accuracy of the "electronic worksheet" throughout an organization.

With this program, you'll be able to leverage the knowledge of your most knowledgeable people. Make more productive use of more of your staff. And do a better, faster, smarter job.

*"I create a complex, yet easy-to-use worksheet..."*

### The best way to do what every company wants to do better

With VisiCalc Advanced Version, the people and departments in your firm will be able to quickly develop their plans, in a uniform format, for easier consolidation into your overall plan.

And everyone involved in your planning process will be able to make their most productive contribution to the effort.

### Sophisticated planning made easy for everyone

Using VisiCalc Advanced Version, knowledgeable individuals within your company can create powerful, preformatted worksheets, or "templates," for nearly anything numerical.

For example, the knowl-

*"Then, doing our department's budget is as easy as filling in the blanks!"*

*"The program makes it practically impossible to make mistakes!"*

*"I ask 'What if?' by changing any number... the new results are instantly calculated."*

edge of a corporate financial officer can be applied to create templates for complete and consistent financial reporting. Then others in your company without extensive financial or computer expertise can easily use these templates to provide the desired information in a uniform manner.

It's so easy. They simply "fill in the blanks" on the worksheet. In seconds, the program performs all the complex calculations built into the worksheet, and displays the answers to problems that could have taken hours to solve by hand.

### Everyone can consider "What if...?"

The speed and versatility of VisiCalc Advanced Version make it easy to explore alternatives. Both template build-

*"With just a few keystrokes I consolidate the worksheets..."*

ers and worksheet users can change any number in a problem and instantly see the new results. Everyone will be able to analyze the impact of decisions before they make them.

### Mistakes locked out, help built in

With VisiCalc Advanced Version there's no worry about someone accidentally altering formulas, titles or any other important information. The program can protect all these parts of a worksheet. It can shield confidential data a user need not see and also guide a user step-by-step through a worksheet.

You can also insure that a user can't enter a label when a number is called for—and vice versa.

And if users ever need additional guidance in completing their worksheets, VisiCalc Advanced Version provides comprehensive on-screen help, at a touch of the "?" key.

### Multi-sheet consolidation and finished reports, too

VisiCalc Advanced Version makes it easier to consolidate worksheet totals from various

*"And produce presentation-quality reports for my boss."*

groups, departments or divisions into concise, presentation-quality reports.

### See for yourself

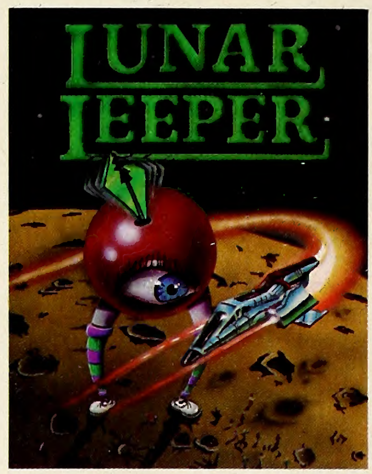
Visit your nearest computer store for a demonstration of VisiCalc Advanced Version. Or write us at 2895 Zanker Road, San Jose, CA 95134 for a copy of our brochure.

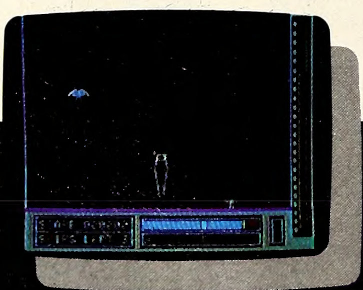
Discover how easy it is to use electronic worksheets in your company for problem-solving, company-wide.

VISICORP™
PERSONAL SOFTWARE™

# JEEPERS CREEPERS!

# DON'T KILL THEM LEEPERS!



Them leepers is cute.
Them leepers ain't kind.
Them leepers will jump up
and bite your behind.

Your job is to fly through
them leepers galore
to rescue some men
is the reason what for.

So be a good pilot
be quick and be sly
Don't kill the leepers
but shoot up the eye

Save all the men
and avoid all the leepers
fly through a cave
and shoot the lunar leepers keepers

If you want to know
what this silly poem's for
check out LUNAR LEEPERS
at your local computer store

Lunar Leepers is a funtastic new APPLE arcade game from CHUCKLES, creator of Laff Pak. Lunar Leepers is available for $29.95 at your local computer store or directly from:

SIERRA ON-LINE INC
36575 Mudge Ranch Rd.
Coarsegold, CA 93614
Phone (209) 683-6858

VISA MASTERCHARGE
CHECK or COD

APPLE is a registered trademark of APPLE INC.

**SIERRAVISION**

Sierra ON-LINE INC

```
       *
                 OBJ  $300
                 ORG  $300
       *
       CHKCOM    EQU  $DEBE
       FRMNUM    EQU  $DD67
       MOVMF     EQU  $EB2B
       *
0300:  20 BE DE  ENTRY  JSR  CHKCOM
0303:  20 67 DD         JSR  FRMNUM   ;BASIC->FAC
0306:  A0 03            LDY  #$03     ;HI BYTE
0308:  A2 80            LDX  #$80     ;LO BYTE
030A:  20 2B EB         JSR  MOVMF    ;FAC->MEM
030D:  60        DONE   RTS
       *
```

The key to this technique is a routine in Applesoft called MOVMF ($EB2B—Move to Memory from FAC), which takes the value in FAC and moves it to the location pointed to by the Y and X registers (X,Y = low byte, high byte).

The listing given here uses our previous FRMNUM technique to get a predictable number into the FAC. The X and Y registers are then loaded to point to $380. When MOVMF is called, the contents of the FAC will be deposited there.

To see this, run the program, then enter the Monitor and type in:

```
380.384 (return)
```

You should get:

```
380-  84  20  00  00  00
```

This proves that we have successfully moved the data from FAC to an arbitrary place in memory.

**Moving Memory into the FAC.** The converse of this operation is accomplished in much the same way. In this case, the Applesoft routine MOVFM ($EAF9—Move to FAC from Memory) is used. It requires that the Y register and accumulator be loaded with the high and low order bytes of the address to be used as the data source for the FAC. (Note that there is a difference here. MOVMF uses X and Y; MOVFM uses X and A!)

```
*********************************************
*                                           *
*              MEMORY -> FAC                *
*                                           *
*            SYNTAX: CALL 768               *
*********************************************
*
                 OBJ  $300
                 ORG  $300
*
      MOVFM      EQU  $EAF9
*
0300:  A0 03     ENTRY  LDY  #$03     ;HI BYTE
0302:  A9 80            LDA  #$80     ;LO BYTE
0304:  20 F9 EA         JSR  MOVFM    ;MEM->FAC
0307:  00               BRK
```

Assuming that the previous routine has already been executed and that $380 is loaded with the data appropriate to the value 10, type in *call 768*.

You should end up in the Monitor, at which point you can verify the contents of the FAC by typing in:

```
9D.A2 (return)
```

You should get:

```
9D-  84  20  00
A0-  00  00  00
```

Again, the BRK was used to end the routine so that we could immediately examine the contents of the FAC. This routine shows that we can move data from a section of memory back into the FAC.

**Passing FAC Data Back to Applesoft.** If the FAC does contain the result of an operation, how can we pass it back to a calling Applesoft program, preferably into the variable of our choice? Again, the answer is to use the MOVMF routine. In this case, rather than moving the contents of the FAC into an arbitrary memory location, we'll find the location of the data bytes of a given real variable and then move the FAC into them. This has the effect of setting the variable equal to the contents of the FAC.

Consider this listing:

```
                1  *******************************************
                2  *                                         *
                3  *               FAC- >BASIC               *
                4  *                                         *
                5  *           SYNTAX: CALL 768,Y            *
                6  *******************************************
                7  *
                8             OBJ   $300
                9             ORG   $300
               10  *
               11  CHKCOM     EQU   $DEBE
               12  PTRGET     EQU   $DFE3
               13  MOVMF      EQU   $EB2B
               14  MOVFM      EQU   $EAF9
               15  *
0300: A0 03    16  ENTRY      LDY   #$03      ; HI BYTE
0302: A9 80    17             LDA   #$80      ; LO BYTE
0304: 20 F9 EA 18             JSR   MOVFM     ; MEM-> FAC
               19  *
0307: 20 BE DE 20             JSR   CHKCOM
030A: 20 E3 DF 21             JSR   PTRGET
030D: AA       22             TAX
030E: 20 2B EB 23             JSR   MOVMF     ; FAC-> VARIABLE
                                                   (MEM2)
0311: 60       24  DONE       RTS
```

This routine again assumes that the floating point data for the number 10 still exists at $380. When this routine is run, lines 16 through 18 duplicate the previous listing to move the floating point data from $380 through $384 into the FAC.

Line 20 uses CHKCOM to check the comma and move TXTPTR to the first character past the comma. Line 21 uses the PTRGET ($DFE3—Pointer Get routine) to locate the variable currently pointed to by TXTPTR. PTRGET is handy also in that it will create the variable in the variable table if it does not already exist. PTRGET returns with the Y register and accumulator pointing to the data bytes of the specified variable. This will be precisely where we want the data in the FAC to be moved to. The only correction to be made is in regard to the fact that MOVMF requires that the Y and X registers (rather than Y and the accumulator as was left by PTRGET) hold the destination address. Line 22 solves this by the TAX command, at which point MOVMF is called. We're now done, and the RTS will return to the calling program.

Test this routine with the following listing:

```
10  CALL 768,X
20  PRINT X
```

X gets set to 10 by having our routine transfer the floating point data from $380 through $384 to the data bytes for the variable X.

**Putting It All Together.** For a real test of these combined techniques, let's see if we can successfully pass data from Applesoft to the FAC to a memory block and then back to the FAC and back to Applesoft. The following routine should demonstrate the entire operation as an overall example of the ideas presented thus far.

```
                1  *******************************************
                2  *                                         *
                3  *  BASIC-> FAC-> MEM->FAC->BASIC          *
                4  *                                         *
                5  *          SYNTAX: CALL 768,Y,X           *
                6  *******************************************
                7  *
                8             OBJ   $300
                9             ORG   $300
               10  *
               11  CHKCOM     EQU   $DEBE
               12  PTRGET     EQU   $DFE3
               13  FRMNUM     EQU   $DD67
               14  MOVFM      EQU   $EAF9
               15  MOVFM      EQU   $EB2B
               16  *
0300: 20 BE DE 17  ENTRY      JSR   CHKCOM
0303: 20 67 DD 18             JSR   FRMNUM; FP->FAC
               19  * - - - - - - - - - - - - - - -
0306: A0 03    20             LDY   #$03      ; HI BYTE
0308: A2 80    21             LDX   #$80      ; LO BYTE
030A: 20 2B EB 22             JSR   MOVMF     ; FAC->MEM
               23  * - - - - - - - - - - - - - - -
030D: A0 03    24             LDY   #$03
030F: A9 80    25             LDA   #$80
0311: 20 F9 EA 26             JSR   MOVFM     ; MEM->FAC
               27  * - - - - - - - - - - - - - - -
0314: 20 BE DE 28             JSR   CHKCOM
0317: 20 E3 DF 29             JSR   PTRGET
031A: AA       30             TAX             ; MOVE LO BYTE TO 'X'
031B: 20 2B EB 31             JSR   MOVMF     ; FAC->FP
               32  * - - - - - - - - - - - - - - -
031E: 60       33  DONE       RTS
```

Try this Applesoft program to call the routine:

```
10  Y = 10
20  CALL 768,Y,X
30  PRINT X
```

The value 10 should be printed for X. Dashed lines have been used to separate the four major sections of the routine. When you compare each section with the four routines presented, the net operation of the example should become clear.

The USR routine could also have been used and would eliminate two of the sections:

```
                1  *******************************************
                2  *                                         *
                3  *  BASIC->FAC->MEM->FAC->BASIC            *
                4  *           VIA THE 'USR'                 *
                5  *                                         *
                6  *          SYNTAX: X = USR(Y)             *
                7  *******************************************
                8  *
                9             OBJ   $300
               10             ORG   $300
               11  *
               12  CHKCOM     EQU   $DEBE
               13  PTRGET     EQU   $DFE3
               14  FRMNUM     EQU   $DD67
               15  MOVFM      EQU   $EAF9
               16  MOVMF      EQU   $EB2B
               17  *
0300: A0 03    18  ENTRY      LDY   #$03      ; HI BYTE
0302: A2 80    19             LDX   #$80      ; LO BYTE
0304: 20 2B EB 20             JSR   MOVMF     ; FAC->MEM
               21  * - - - - - - - - - - - - - - -
0307: A0 03    22             LDY   #$03      ; HI BYTE
0309: A9 80    23             LDA   #$80      ; LO BYTE
030B: 20 F9 EA 24             JSR   MOVFM     ; MEM->FAC
               25  * - - - - - - - - - - - - - - -
030E: 60       26  DONE       RTS
```

Notice that since the USR function calls the routine with the FAC already loaded with the value for Y, the first section of the previous routine is not needed. Also, since the USR function will automatically assign the contents of the FAC to the variable X, the last section of the previous routine is not needed.

The calling program for the routine would look like this:

```
10  POKE 11,0: POKE 12,3: REM SET UP USR VECTOR
20  Y = 10
30  X = USR(Y)
40  PRINT X : REM SHOULD PRINT '10'
```

**Conclusion.** By now you probably feel fairly comfortable with the idea of the floating point accumulator (FAC) and how data can be moved about between Applesoft and machine language. Next month we'll begin looking at some of the more sophisticated routines Applesoft uses to perform the various arithmetic functions. See you then!

# The Schoolhouse Apple

by Jean Varven

Dr. Wilbur Pillsbury has cause to celebrate. Thanks in part to an Apple Education Foundation grant he received early in 1980, Pillsbury has just seen the first fruit of a project that has occupied much of his spare time for nearly three years. He has just paged through the first printed copy of his new book.

**Setting the Scene.** A professor of economics at Knox College in Galesburg, Illinois, Dr. Pillsbury began writing about computers and accounting in the late sixties. In 1970, he published two textbooks for South-Western Publishing Company, the first intended for the nonmajors college course, the second for the first course at the high school level.

The computers Pillsbury wrote about in the early seventies came from corporations like IBM, Burroughs, Honeywell, and Control Data and were, of course, of the mainframe variety. Because his books met a real need, both were quite successful, selling well and seeing three editions in the years that followed.

But times are changing, and so are computers. Fittingly, Pillsbury's most recent book, written with the assistance of William Ripperger and also published by South-Western, bears the title *Microcomputer-Oriented Accounting: A CAI Approach*. Intended for use in a high school introductory course in financial accounting, the newly released volume is accompanied by software that runs on the Apple II. The college-level text, scheduled for publication next spring, will also come with Apple-compatible software.

Both publishing projects began because South-Western and Pillsbury believed that micros would play a significant role in education in the future. The plan was to adapt the basic course material to take advantage of the microcomputer's flexibility, menu-driven programs, and other special features.

The Apple Education grant that was awarded to Pillsbury's project in March of 1980 helped to make this possible. The grant took the form of equipment—a 32K Apple II Plus, two disk drives, a monitor, and a printer. Pillsbury already had a contract from South-Western to do the new project; all he had needed was the equipment that would enable him to move ahead. (The equipment is also used by others in the economics department.)

**Working Together.** The traditional accounting course, which has been offered for many years, has been altered for the last twelve years by the use of computers. Students who study this material learn to journalize, post, and do various kinds of balances. They also prepare worksheets and balance sheets, do cost accounting, and do interest accounting (figuring such things as simple interest, compound interest, present value of an annuity, and so on).

The text of Pillsbury's *Microcomputer-Oriented Accounting* is designed so that chapters can be studied in any order. It is intended to take students "step by step through a financial accounting course." The programs on disk are menu-driven, and there's room on each disk for fifty students to store one set of data, ten students to store five sets of data, or any equivalent combination. Data stored on disk can be recalled, altered, or added to.

As many authors and editors will attest, it's no simple matter to create a learning package that really teaches the material it's intended to cover. The task requires the effort, expertise, and cooperation of many people. In the case at hand, author and publisher had the advantage of being able to draw on their experiences working together on other books.

In addition, the agreement between South-Western and Apple Computer called for the materials to be class-tested in various high schools and to be revised in response to the critiques received. South-Western made arrangements to test the package in four high schools. This ensured that author and publisher would receive feedback from teachers and students who had actually used the book and its accompanying software in various stages of development.

Pillsbury says that the feedback they received made a real difference in the final product. Participating teachers and students filled out a questionnaire prepared by Pillsbury, and, to learn firsthand what kinds of experiences people were having with the package, Pillsbury visited each of the high school classes twice during the year.

So what did teachers and students have to say? Well, one initial reaction, which Pillsbury says came as a bit of a surprise, was that the computer was doing too much. At first, for instance, the program was set up so that the computer closed the books automatically. Experienced accountants would have welcomed such an arrangement as a real time saver. But for high school students it was all wrong—they would never learn how to close the books themselves. So the program was revised.

The matter of worksheet preparation provided a similar lesson to the creators of the package. The program had originally been set up to prepare a complete worksheet automatically, but, as it turned out, a much better arrangement was for students to prepare a portion of the worksheets themselves. Once again, what would have been handy for professionals was not right for novices—it would have taken away too much of the learning experience students needed.

**Little Big Difference.** Asked what differences he sees between using micros in the classroom versus working with the larger computers, Pillsbury is reminded of the first time the University of Puerto Rico asked him to come there and help teachers learn to use computers in the classroom.

"At that time, we had to bring two large boxes with four thousand IBM cards in them. This year," he reports, appreciatively, "we will bring one diskette for the Apple. The disk will have more information on it than the four thousand cards did."

Chances are that students and teachers who get the opportunity to use the Apple in an accounting class will soon have their own stories to tell about the advantages the microcomputer offers.

**In Capsule Form.** Let's look now at some more companies that produce educational software for the Apple.

**Developmental Learning Materials**, One DLM Park, Allen, TX 75002; (214) 248-6300.

DLM produces the *Arcademic Skill Builders in Math* series, six programs for learning basic addition, subtraction, multiplication, and division skills. The programs are drill and practice but with a difference—they have sound, color, and graphics designed to give them the appeal of arcade games.

Although the programs can be used separately, they are actually intended to be a "total approach to building essential mathematics skills." The series includes *Alien Addition*, *Minus Mission*, *Meteor Multiplication*, *Demolition Division*, *Alligator Mix*, and *Dragon Mix*.

Each of the games has a different theme but the same basic objective—you must eliminate the problems that are coming at you by firing the correct answers at them before it's too late. Each game has three different skill levels and nine different speeds. Paddles or sets of designated keys can be used in playing the games.

A central premise in the design of these programs is motivation. Motivation can be an especially important factor in promoting learning that involves drill and practice. Students are likely to learn better and to spend more time on learning if they feel motivated and engaged than if they feel bored or if they believe that something is beneath them. According to the company, young people in grades two through eleven have used these materials successfully. Field testing has shown that students using the programs learned rapidly and were highly motivated.

Programs can be purchased separately or as part of the complete package. Each individual program is accompanied by a teacher's manual that explains how to set up individualized learning plans for each student and various other instructional components. Included are an instruction sheet, reproducible master worksheets and record sheets, and fifty-two four-color flashcards that contain the game's problems and match its motif. The record sheets include a sheet to be used by the student in recording performance, a progress chart to be used for recording goals and charting progress over a ten-week period, and a strategies for improvement sheet on which to record specific plans for helping individual students improve their skills.

Six new programs, *Arcademic Skill Builders in Language Arts*, are scheduled for January release.

**Right On Programs**, Box 977, Huntington, NY 11743; (516) 271-3177.

As former high school English teacher Don Feinstein explains, Right On Programs was formed two years ago "out of need" and somewhat by accident.

Don's wife, Barbara, an elementary school librarian, had access to a school computer but almost no software for it. After thinking about the kinds of library skills she wanted to help kids learn, Barbara asked their then high school age son whether he could program some software to her specifications. He agreed to give it a try, and the program that resulted from their collaboration soon caught the attention of other librarians in the district. When a number of them wanted to buy that first program, a small company was born.

This pattern of seeing a need and responding to it has been repeated many times now. At present, the company offers 130 software programs for the Apple. In addition, a series of elementary school language arts programs will be released around the first of the year.

For students in elementary school, the company has programs in library skills, social studies, science, and math. The programs are designed by teachers, programmed by programmers, and tested by young people in various parts of the country. Don Feinstein stresses how valuable children's input is—it lets the company know what they should be doing, what they may have missed, and to what level a program should be geared.

The library programs cover such areas as the Dewey decimal system, understanding and using the card catalog, locating books on the shelf, understanding the title page, and using an index. Two library management programs—one for dealing with overdue books, the other for providing library patrons with information about new books—are also available.

Social studies titles include *Farm Life*, a program that introduces elementary school students to the basic concepts of farms, farm animals, and farm products; *Community Helpers*, a discussion of community services and the people who provide them; and *Who Built America*, a program about why people came to this country, what they brought, and what they contributed.

Among the science offerings are programs on the solar system, light, electricity, matter, energy, birds, mammals, and insects. Math programs cover such areas as basic arithmetic skills, math measurements, telling time, and dealing with money. The elementary school programs are geared to cover information that is part of the current curriculum. They can be purchased individually or as part of various multiprogram packages.

Secondary school programs include a comprehensive language arts series and a series of programs designed to help high school juniors improve their skills in preparation for the SAT test.

**Vernier Software**, 2920 S.W. 89th Street, Portland, OR 97225; (503) 297-5317.

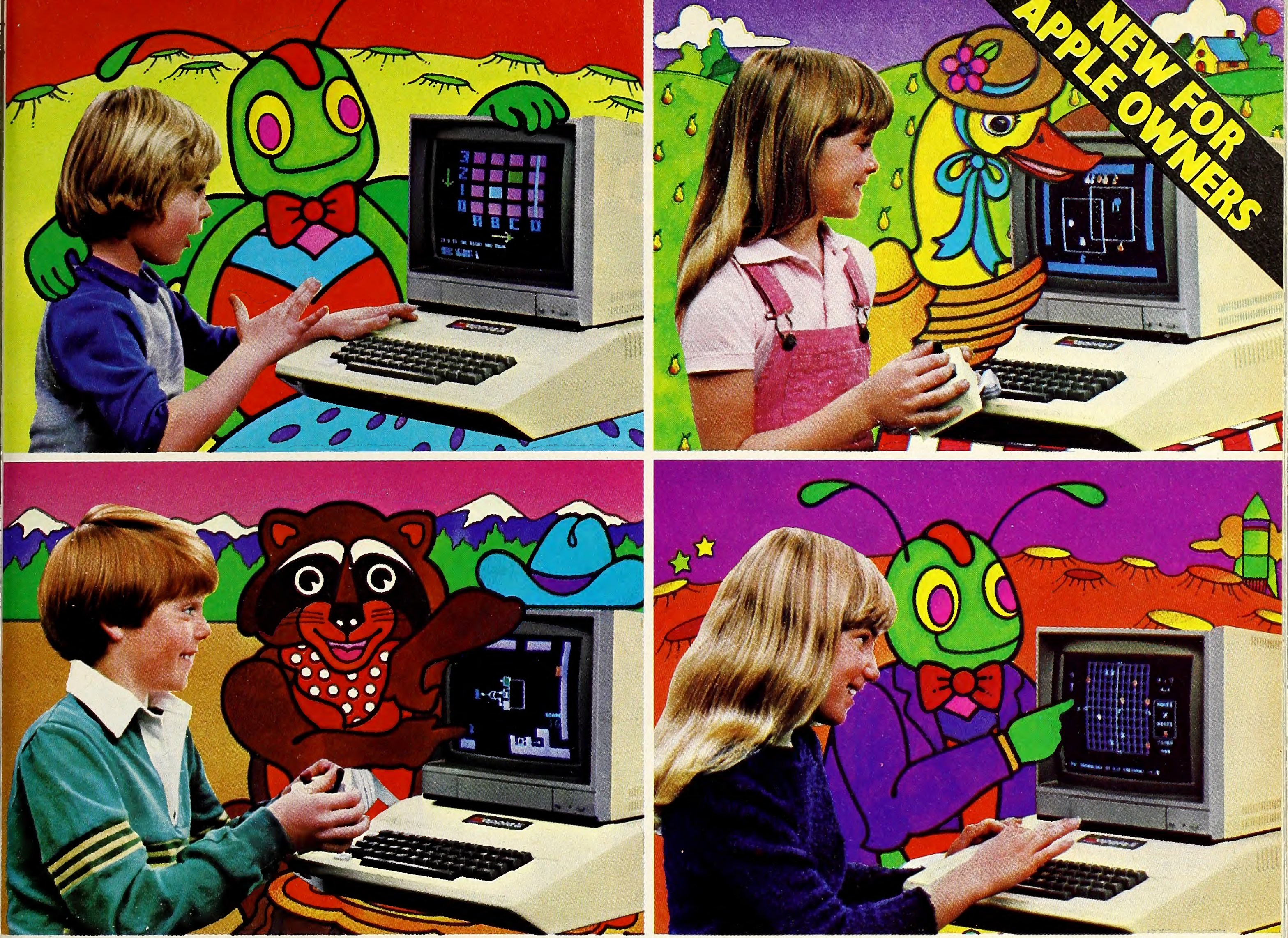
All nine programs from Vernier Software run on the Apple and are designed primarily for use in introductory level high school and college physics courses. Program titles include *Wave Addition*, *Projectiles*, *Vector Addition*, *Kinematics*, *Charged Particles*, and *Orbit*. All programs feature hi-res graphics.

*Wave Addition*, for example, employs hi-res graphics to demonstrate the superposition of waves. Eight different operating modes are available; some are preprogrammed to demonstrate various principles, while others require that the student select the waves to be added. The program can be used in the classroom to show the relationship between wavelength, frequency, and wave speed; constructive and destructive interference; and how wave shape is affected when various overtones are added. It can also be used to simulate the shapes of the waves produced by the sounds of various musical instruments.

In *Projectiles*, students get the chance to study a projectile's motion under various conditions. Seven modes are available. This allows for launching from the top of a cliff or from ground level with or without air resistance and with or without horizontal wind.

The company recently announced the release of three new programs—*Graphical Analysis*, which plots graphs in hi-res; *Precision Timer*, which allows you to use the computer as a lab timer that displays times to a tenth of a millisecond; and *Ray Tracer*, which draws ray diagrams to illustrate the principles of geometrical optics. *Ray Tracer* can be used to illustrate concepts in an optics lesson or as an educational game to help students learn to anticipate how the optical devices and interfaces that a ray encounters will affect it.

All of Vernier's programs are written by company founder David L. Vernier, an experienced physics teacher. All have been field-test-




# NOW THE COMPUTER AGE IS THREE TO THIRTEEN.

With Computer Learning Games™ from The Learning Company and an Apple computer, you can give your kids one of life's greatest gifts — the love of learning.

As the company that makes learning playful, we can help you give your kids a head start on life.

Younger kids will learn direction concepts like "above or below" and "left or right." Older kids acquire logic skills while creating patterns with colors and shapes. They'll also learn computer-age skills like number plotting and computer chip circuit design.

All learned from creative games using imaginary characters and lively themes. From Juggles' Rainbow to Gertrude's Puzzles, there are six packages in all.

So if you've got kids aged three to thirteen, take them to your local Apple dealer and try our learning games.

Then you'll see how exciting learning really can be.

Please send me a copy of your computer learning games catalog.

NAME ____________________

ADDRESS ____________________

CITY ____________________

STATE __________ ZIP ______

The Learning Company
4370 Alpine Road, Dept. 104
Portola Valley, CA 94025
(415) 851-3160

THE LEARNING CO.
COMPUTER LEARNING GAMES

# LET US ENTERTAIN YOU!

Avant-garde means "ahead of its time" and Avant-Garde Creations truly lives up to its name with exciting entertainment software that promises more than the mundane! Not just the usual bang-bang shoot-em-up types, but challenging games that require skill, timing, fast thinking and finesse.

**ZERO GRAVITY PINBALL** keeps you busy even at speed 1 with 10 flippers, 2 force fields, an unpredictable black hole and of course zero gravity.

**LAZER MAZE** demands a keen eye and a quick trigger finger. The mirror maze challenges perceptual skills you probably didn't even know you had!

**FEDERATION** puts you in command of a space fleet on a mission to wipe out the merciless invading Drorn Drones. Its lightening fast action requires perfect timing.

**DEATH RACE '82** combines the skill of high-speed driving with the thrill of a perilous chase as you attempt to escape from robot guards in Death Squad Cars.

**HI-RES COMPUTER GOLF**, our all-time best seller, is better than the real thing! This highly acclaimed game requires strategy, judgement, visual keeness and skill to meet the challenge of five 18-hole courses. Yelling "Fore!" is optional.

All of these great games run on Apple II Plus 48K and retail for $29.95 They are available at your local dealer or you may order direct from Avant-Garde Creations.


AVANT-GARDE CREATIONS

**AVANT-GARDE CREATIONS**
P.O. BOX 30160 • EUGENE, OR 97403 • (503) 345-3043
*BUSINESS, EDUCATION, UTILITY AND ENTERTAINMENT SOFTWARE*

ed and modified to take the feedback received into account.

Each program's documentation includes a user guide for students and a teacher's manual containing background information and suggestions about how to use the program in the classroom. The programs have been left unprotected, and Applesoft program listings, with special notes explaining program operation, are provided. This makes it possible for teachers to modify programs to suit their own special needs.

**Review of Delta Drawing.** Designed especially for youngsters ages four through twelve, this easy-to-use, class-tested program turns the computer screen into a drawing board. Because of the way the commands are set up, even children who haven't yet learned to read can use *Delta Drawing* once they've learned to associate the correct letters with the correct commands.

*Delta Drawing* teaches young learners about lines, shapes, and colors. They'll become more adept at recognizing patterns and thinking visually. In the process, the program introduces children pleasurably to the kind of deliberate, step-by-step approach that is required for solving problems on the computer.

The orientation of *Delta Drawing*'s creators is refreshing. They stress the value of exploration, rather than concern with right and wrong answers. As the manual puts it, in *Delta Drawing*, "There are no mistakes, only unexpected results."

The program requires 48K and works best with a color monitor or television set. This allows the user to draw in green, blue, orange, and purple in addition to white. Shapes can also be colored in or a background color can be added as a finishing touch to a child's elaborate "creation."

Drawings are created by means of logically named single-keystroke commands. Each command controls a single, separate function.

The drawing instrument is the delta cursor, a small pointer similar to the Logo turtle, whose start state is the center of the screen. The cursor has two attributes, position and heading. Position refers to where the cursor is on the screen; heading refers to the direction it's going. The cursor's voice (a beep tone that can be turned off) lets the user know that a command has been "heard" and acted on.

The commands that a young learner will start out using include D for drawing a line of fixed length on the screen; M for moving to various locations on the screen without drawing anything; L, R, and U for turning the cursor left or right (in thirty-degree increments) and executing a U-turn; and E, the erase command.

With a bit of assistance from a parent, a teacher, or an older child, a small child can learn to use the half-draw and half-move commands (which involve using the control key) and the color commands.

Several qualities make *Delta Drawing* more than and different from a sketch pad or coloring book. To begin with, the delta cursor remembers what it draws, and it records this information in the form of a simple drawing program. Pressing T for text reveals a program listing showing what sequence of commands produced a particular shape. G returns to graphic mode.

Being able to toggle with single keystrokes makes it easy to see the relationship between pictures on the screen and the step-by-step programs that create them. And understanding this relationship opens up a whole new way of making pictures—creating a program in text first and then viewing the results in graphic form.

There are, in all, thirty-five command keys and eight types of commands, including those for drawing and editing, for storing and executing programs created by the user, for color and display, for printing, for system manipulation, and for repeating action. It's great that the program offers all these capabilities, and it's also nice that a child can enjoy using the program without needing to know about many of them.

The keys for numbers one through nine can be used for storing that many drawing programs in memory at once. These programs can be recalled and reused throughout one session. It's also easy to nest programs within one another to create more elaborate designs or more complex illustrations.

The pictures and designs a user creates can be saved on a data disk. Picture files can be reloaded at later sessions to be worked with some more or to be shared with a user's admiring friends. A Centronics parallel printer can be used to print out the text of a drawing program, while an Epson with the Grappler interface card makes it possible to print out both program text and pictures.

Certain things about this program take some getting used to. It's important to realize, for example, that the delta cursor always begins drawing from its current position and heading. This means that when a stored program is executed the result may differ from the way the drawing looked originally. This is because the cursor's position and heading when the program is called differ from its start position when the program was created.

And, as good as this program is, it has its limitations. Some of these, such as a jagged, striped effect you get when you put certain colors next to each other, can be traced to the idiosyncracies of the Apple's hi-res graphics.

Learning to take advantage of the program's more complicated features—such as the capacity to edit one's own drawings and programs—will require time and effort, even on the part of an older user. Nor will the program turn a child into a great artist; but, then, that's not its intent.

But don't let these factors dissuade you from seriously considering this program. Fiddling with *Delta Drawing* will remind you just how exhilarating it can be to explore, experiment, discover, and create.

*Delta Drawing*, Version 2.0, by Computer Access Corporation, Spinnaker Software (26 Brighton Street, Belmont, MA 02178; 617-868-4700). $59.95.

A Schoolhouse Apple Tutorial

# LOGO The Voice of THE TURTLE

BY DONNA BEARDEN

When is a house not a house? When it becomes a tire, of course.

What? How can a house become a tire?

Simple. When a bug is not a bug.

If that's one too many riddles for you, stick around. Logo has a wonderful way of working out puzzles. It's full of possibilities that will make you think.

Brenda is a young mother who recently enrolled in a beginning Logo class. Her son is in kindergarten and is a member of the Young Peoples' Logo Association. Brenda became fascinated with the Logo language and wanted to learn more about it, both for her own benefit and to be able to work with her child.

As might be expected, adults learn Logo much the same way children do, by means of lots of experimenting. After using the initial random commands, you find yourself wanting to "make something"—a defined shape. The logical sequence is to move on to a triangle and then to put a triangle and a square together to make a simple house. To make a suburb, you simply repeat HOUSE a certain number of times. Thus, each step builds logically on the step before.

One of the things that makes Logo such a powerful tool for children is that it teaches them to break problems down into pieces and to solve the pieces first. Before they ever try to draw a neighborhood, they learn to draw a house. Before they try to draw a house, they learn to draw a square and a triangle. And all this is preceded by doodling with simple commands and numerical inputs just to see what will happen.

Perhaps even more powerful is what can happen in the imagination when pieces don't do "what they're supposed to." That's when the "art" of programming comes in—the inspiration that sends someone down a totally new path of discovery and creation. It happened to Brenda, and she turned a house into a tire.

First, Brenda went through the process of defining a square and a triangle. Then she put them together to make a house. When she had one house on the screen, she decided that she'd like to put another one there also, so she typed in the command "HOUSE." The turtle, of course, followed directions. And since Brenda had not moved the turtle from its ending position on the first house, she wound up with the new house growing sideways out of the roof of the original one. See figure 1.

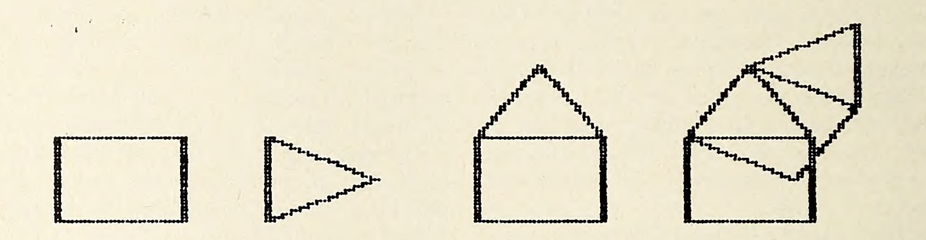


Figure 1. Making a house.

A bug? Well, yes, if Brenda had really had her heart set on trying to develop a suburb. But to her, the "bug" just presented new possibilities. She wondered out loud what would happen if she just kept repeating HOUSE. She did this and when the design was completed, the angular lines of the squares and triangles had gently curved into a circular pattern. To one person, it looked like a sun, another said it looked like a Ferris wheel, and to a third, it was a tire. Take a look at figure 2.

And then, as so often happens, the artist in Brenda came out and took over. The conversation had turned to auto racing and what had begun as a simple house procedure turned into a TIRE. It wasn't long before TIRE became TIRES, and by then it wasn't hard at all to visualize a custom-made sports car to fit on such stylish tires. See figure 3.

And thus the bug was not a bug and the house was not a house. Not anymore.

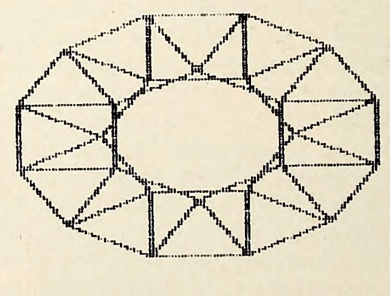


Figure 2. What do you see?

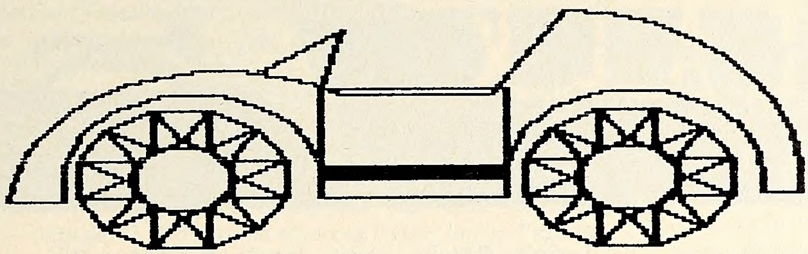


Figure 3. Procedure for creating a sports car.

```
TO SPORTSCAR
SETUP
TIRES 15
BODY
GRAPHICSDUMP
END

TO GRAPHICSDUMP
.PRINTER 1
TYPE CHAR 9 TYPE "G TYPE "E TYPE CHAR
   13
POPS
.PRINTER 0
END

TO TIRES :N
TIRE :N
PU LT 90 FD :N * 8 RT 90 PD
TIRE :N
END

TO TIRE :N
REPEAT 12 [HOUSE :N]
END

TO HOUSE :N
SQUARE :N
FD :N
RT 30
TRI :N
END

TO SQUARE :N
REPEAT 4 [FD :N RT 90]
END

TO TRI :N
REPEAT 3 [FD :N RT 120]
END

TO SETUP
HT
PU RT 90 FD 30 LT 90 PD
END

TO BODY
PU RT 90 FD 65 PD FD 50 LT 90
WHEELWELL
SETH 90 FD 10 LT 90
REPEAT 9 [FD 12 LT 10]
LT 60 FD 40 SETH 270 FD 40 RT 30
REPEAT 12 [FD 10 LT 10]
SETH 90 FD 15 LT 90
WHEELWELL
STRIPES
WINDSHIELD
END

TO WHEELWELL
FD 7 REPEAT 36 [FD 3 RT 5] FD 9
END

TO STRIPES
BK 12 LT 90
REPEAT 3[FD 50 RT 90 FD 1 RT 90 FD 50 LT
   90 FD 1 LT 90]
END

TO WINDSHIELD
LT 90 FD 38
DOORS
RT 15 FD 25 LT 143 FD 30
END

TO DOORS
RT 90 FD 50 RT 90 FD 38
BK 38 LT 90 BK 50 LT 90
END
```

**Word Geometry.** The turtle isn't the only feature of Logo that can be used to make geometric figures. Some of the list processing features can also be used to make a variety of different shapes. Rena Upitis, of Queen's University in Ontario, Canada, submitted a procedure to make a triangle using BUTFIRST. BUTLAST can also be used. In fact, if you want to print out some graphic shapes, combine the two commands.

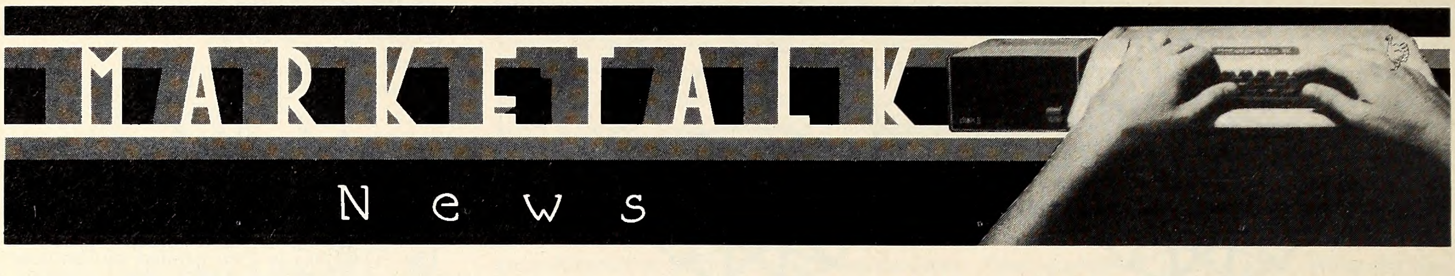
Here's one way to make a word triangle:

```
TO WORDTRIANGLE :STRING
.PRINTER 1
IF :STRING = " [STOP]
PRINT :STRING
WORDTRIANGLE BUTFIRST :STRING
END
```

The first two lines set up the procedure. First, output is directed to the printer through slot 1. Then the conditional statement says that if :STRING is equal to " then STOP.

The :STRING is then printed by the next line. The last line then tells the program to repeat but without the first element of :STRING. Here's what happens:

```
THAT'S.ALL.FOR.NOW,FOLKS!
HAT'S.ALL.FOR.NOW,FOLKS!
AT'S.ALL.FOR.NOW,FOLKS!
T'S.ALL.FOR.NOW,FOLKS!
'S.ALL.FOR.NOW,FOLKS!
S.ALL.FOR.NOW,FOLKS!
.ALL.FOR.NOW,FOLKS!
ALL.FOR.NOW,FOLKS!
LL.FOR.NOW,FOLKS!
L.FOR.NOW,FOLKS!
.FOR.NOW,FOLKS!
FOR.NOW,FOLKS!
OR.NOW,FOLKS!
R.NOW,FOLKS!
.NOW,FOLKS!
NOW,FOLKS!
OW,FOLKS!
W,FOLKS!
,FOLKS!
FOLKS!
OLKS!
LKS!
KS!
S!
!
```




*Unless otherwise noted, all products can be assumed to run on either Apple II, with 48K, ROM Applesoft, and one disk drive. The requirement for ROM Applesoft can be met by RAM Applesoft in a language card. Many Apple II programs will run on the Apple III in the emulator mode.*

□ A new computer center large enough to accommodate the growth of subscribers to the **Source** (1616 Anderson Road, McLean, VA 22102; 703-734-7500) from its current 22,218 to 250,000 is the latest addition to the nationwide information utility. The center includes nine Prime 750 computers, and future components include a UPI Series 6100 satellite dish, incoming lines for multiple public networks, and more than 100 direct dial WATS lines. Another attraction, the *U.S. News Washington Letter*, is an electronic newsletter for Source subscribers containing a dozen or so brief stories on investment opportunities, economic trends, and key government activities each week.

Users of the Source will now be receiving *Sourceworld Newsletter*, a monthly subscriber publication that replaces *Sourceworld Magazine*. Every month, *Sourceworld* will tell you about practical and useful ways that subscribers are putting the Source to work as an integral part of their personal systems. It includes little-known facts about time-saving ways to use the Source, anecdotes and case histories of fellow subscribers, announcements about new services, and important improvements to old ones. Ideas and suggestions can be sent via SourceMail to TCA098.

□ More in *Omniware* from **Educational Computing Systems** (106 Fairbanks Plaza, Oak Ridge, TN 37830; 615-483-4915): *Omnipack* is a set of three programs: The first, *Omnigraph*, is a system for the interactive creation and display of data graphs. Graphs may be either customized or automatically generated. Produces X,Y plots of data on Cartesian coordinates and constructs pie and bar charts as well. $49.95. *Omnitrend*, a multiple-linear regression and data analysis program, is designed to operate on data files compatible with those used by *Omnigraph* and *Omnifile*. $59.95. And *Omnifile* is a file-management program written in Applesoft that can be used both as a database management and reporting system and as an addition to your own applications programs. Its capabilities include mathematical manipulation of numeric data in files, combining different files into a single file, as well as general database functions. Files are accessible by *Omnigraph* and *Omnitrend*. $59.95. All three programs can be modified by the user and are also sold together for $129.95.

□ **Talmis** (115 North Oak Park Avenue, Oak Park, IL 60301; 312-848-4000) releases *Sources for Courses*, the first of its annual multi-indexed reference guides to microcomputer courseware for kindergarten through college. The index will list more than nine hundred educational programs by title, topic, style, grade level, publisher, price, and the type of computer they run on. It will also indicate related print or audio-visual materials and hardware requirements. Gives references of software reviews published in major educational or software magazines so parents and teachers may compare critiques of programs. $9.95.

□ Joining the more than one hundred fifty on-line databases from **Dialog Information Services** (3460 Hillview Avenue, Palo Alto, CA 94304; 800-227-1927; 800-982-5838 in California) are *Telegen*, containing information about biotechnology and genetic engineering from 1973 to present; *Books in Print*, containing 650,000 records listing current United States book publishing inventory; *Laborlaw*, providing summaries of decisions related to labor relations, fair employment, and wages and hours; *Paperchem*, containing 160,000 records on the paper industry; *Electronic Yellow Pages*, listing more than 880,000 records covering all contractors and construction agencies; and *BLS Employment, Hours, and Earnings*, providing numerical data gathered from the United States Bureau of Labor Statistics. Requires modem. Hourly rates, $45 to $130.

□ Step aside, Richard Simmons. **N-Squared Computing** (5318 Forest Ridge Road, Silverton, OR 97381; 503-873-5906) introduces the *Nutritionist*, a program that shows your nutritive deficiencies and excesses and then balances your diet for you. It creates optimum nutritional menus for individual needs and personal preferences. Displays seventeen nutritive components of more than seven hundred foods both in weight and in recommended daily allowance according to age, sex, weight, and height. To be introduced at the Applefest in San Francisco in October. $145.

□ **Sofstar** (13935 U.S. 1, Juno Square, Juno Beach, FL 33408; 305-627-5511) has announced the *Business Planning Tool with VisiGen*, which will let users generate sophisticated spreadsheets for planning and analysis in minutes instead of hours. After manipulating a bar graph with the arrow keys, the user runs *Business Planning Tool* to generate *VisiCalc* formulas and assembles the information into a custom *VisiCalc* model with *VisiGen*. Options include the *Peachtree Connection*, which allows the user to extract information from Peachtree data files and put it into a *VisiCalc* model, and the *WordStar Connection*, which will convert a DIF file to *WordStar* format so the model may be included in the text of a report. $150. *Peachtree Connection*, $50. *VisiCalc* is separate.

□ *Fast Figure*, the spreadsheet program from **Hourglass Systems** (Glen Ellyn, IL), is being distributed through **American Square Computers** (4167 Kivett Drive, Jamestown, NC 27282; 919-883-1105, 919-889-4577). Its features include depreciation, present value and net present value, internal rate of return, compound growth functions, standard deviations, and what-if analysis. Three-dimensional file sharing allows you to prepare multiple reports from the same data file without data re-entry. Requires CP/M. $99. Other Hourglass programs from American Square: *Market Time*, a stock market investment timing program that comes with a database of New York Stock Exchange statistics. CP/M needed. $75. *High Yield*, an investment management program that records transactions for mutual and money market funds and Individual Retirement Account and Keough plans. It then prepares individual reports as well as portfolio summaries. $75.

□ It's not R2-D2, nor is it even close. But the RB5X robot from **RB Robot Corporation** (14618 West Sixth Avenue, Suite 201, Golden, CO 80401; 303-279-5525) is the first manufactured intelligent robot for the home experimenter. With its own microprocessor, memory, programs, and tactile sensors, the robot detects and responds to objects in its path. It then remembers its actions and repeats the correct response when confronted again with the same situations. Interfaces with the Apple for program entry and data transfer. Recharges its batteries by itself, and has option for the Polaroid Rangefinder sonar sensor. Future add-ons include a mechanical arm, a voice synthesizer, and digital radio communications between RB5Xs. $1,195.

□ Take off to the Great White North and take a look at the *Cash Register and Inventory System* (*CRICS*) from **Canuck Software** (Box 11984, Edmonton, Alberta T5J 3L1, Canada). *CRICS* is a point-of-sale inventory control system capable of handling 1,200 stock items. It keeps statistics on stock number, description, vendor, buying and selling prices, margin, quantity on hand and on order, and number sold during current month and year. *CRICS* also produces daily, monthly, and yearly sales reports; price lists; margin reports; and others. Requires two disk drives and an eighty-column printer. $99.95.

□ *Micromarkets* is a database from **Orrington Economics** (700 North Edison Street, Arlington, VA 22203; 703-527-5990) for use by market researchers and sales managers. Designed for easy use with *VisiCalc*, the disk contains thirty-five key demographic and retail sales variables for each state and for each of the thirty-eight largest metropolitan areas. You can manipulate data using *VisiCalc* to customize trading areas and indices of market potential. Use company sales data to compute market share in different regions of the country. The program stores data in DIF format to make it compatible with other programs as well as with *VisiCalc.* $119.

☐ October Erratum: The telephone number of **Study Guide Software,** makers of *Spanish Language Review,* is (714) 540-8343.

☐ **National Computer Shows** (824 Boylston Street, Chestnut Hill, MA 02167; 617-739-2000) announces CP/M '83, an international exposition and conference for the CP/M industry and users that will be held January 21 through 23, 1983, at the Moscone Center in San Francisco. CP/M '83 is sponsored by Digital Research and will feature manufacturers, software developers, OEMs, software publishers, distributors, dealers, and users. Hours are 11:00 a.m. to 6:00 p.m. daily. CP/M recommended, not required. One day admission, $10; three days, $20.

☐ Bargain: **Advanced Logic Systems** (1195 East Arques Avenue, Sunnyvale, CA 94086; 408-730-0306) has announced an agreement with MicroPro to introduce *WordStar,* the popular word processing program, at special prices when purchased with the Synergizer package or with a Z-Card. The Synergizer includes three interface boards: a Z-Card with CP/M system software, a Smarterm eighty column display, and an Add-Ram 16K RAM expansion. The Synergizer also includes a free copy of *SuperCalc.* Synergizer, $750. Z-Card with *WordStar,* $395.

☐ Another bargain: Subscriptions to the Dow Jones Information Services, the Source, and the Comp-U-Star home shopping service are now included with the purchase of Scanset personal information terminals from **Tymshare** (20705 Valley Green Drive, Cupertino, CA 95014; 408-446-6000). Purchasers get a Dow Jones password at no charge, plus one hour free usage of the Dow Jones service, which includes access to news from *The Wall Street Journal, Barron's* and the *Dow Jones News Service;* current quotes on stocks and bonds; and more. If you want the Source, you get basic services without payment of the usual registration fee. Finally, a six month free trial membership to the Comp-U-Star shop-at-home service includes a half hour free usage. Scansets come in two models; both feature programmable function keys, a nine-inch diagonal screen with twenty-four lines of text, forty or eighty character line lengths, and limited graphics capability. The Scanset Model 410 features automatic computer log-in. $495. The Model 415 has a built-in modem, automatic dialer, and automatic log-in. $649.

☐ *Apple Fortran,* by Brian and George Blackwood, presents a complete and specific explanation of Fortran in understandable language aimed at the nontechnical user. It explains keyword use, program statements, edit descriptors, and block statements. Immediately usable Fortran business and engineering programs are included. From **Howard W. Sams** (4300 West 62nd Street, Box 7092, Indianapolis, IN 46206; 317-298-5400). 256 pages. $17.95.

☐ **Persimmon Software** (502 C. Savannah Street, Greensboro, NC 27406; 919-275-5824) helps researchers with *Telephone Survey,* a program for both large and small survey applications. The program generates and dials randomly generated telephone numbers, prompts the interviewer with questions and possible responses, records subject responses, and prints frequency tables and results. Requires direct connect modem and a printer. $35. An updated version of *Test Writer* is now available. It produces multiple choice examinations from a pool of questions developed by the user. The new version simplifies the process of customizing parameters of the output and contains more error trapping than the original. Requires a printer. $35.

☐ **Software Dimensions** (6371 Auburn Boulevard, Citrus Heights, CA 95610; 916-722-8000) has announced their Plus promotion package that will provide end users the opportunity to receive up to $620 worth of software free. Until December 31, 1982, Software Dimensions and their participating distributors and dealers will include one or more free software packages in every *Accounting Plus II* system purchased. The free packages include *Data Plus, GL Plus, Labels Plus,* and *Invoices Plus.* And once you master *Accounting Plus II,* you can use *Data Plus* to convert the data files into the DIF format that is compatible with *VisiCalc.* Now you can manipulate and plot data from the general ledger, accounts receivable and payable, and inventory using products from VisiCorp. Those familiar with *Accounting Plus II* can quickly learn how to use all the features of this menu-driven system. $99.

☐ When sending files, you can reduce transmission time and costs when you compress your CP/M files with *Compress*, a data compression program from **Digital Marketing** (2670 Cherry Lane, Walnut Creek, CA 94596; 415-938-2880). It reduces the size of files by 30 to 40 percent by assigning shorter bit codes to frequent characters and longer bit codes to less frequent ones. Requires CP/M. $59.95. A game for CP/Mers! *Orbquest* is a fantasy game in which the player must find a glittering orb buried in the depths of a multileveled dungeon. Each trip into the dungeon gives you experience and magic, allowing you to go deeper and closer to the orb—if you survive. Requires 56K, CP/M, and a cursor-addressable terminal. $39.95.

☐ *Fred III* is here. This home money management program from **The Creative Mind Workshop** (Box 3017, Columbus, OH 43210; 614-252-2593) uses conversational English to communicate with the user. It's designed to blend with your present money management methods and improve them, not change them. Maintains as many as thirty budget accounts and monitors as many as twelve savings or investment type accounts. $30.

☐ For your reading pleasure from **Prentice-Hall** (Englewood Cliffs, NJ 07632; 201-592-2348): *CP/M Assembly Language Programming*, by Ken Barbier, is a self-teaching manual covering the CP/M operating system and assembly language programming. Follows a learn-by-doing approach. $12.95. *UCSD Pascal: A Considerate Approach*, by David Price, offers a simple way to write programs that are easy to read and use. Written specifically for the novice programmer, the book offers a low-level introduction and focuses on essential programming skills without going into technical detail. Includes many sample programs. $12.95. For the beginning programmer, *Apple II Programmer's Handbook*, by Richard Vile, includes tips and techniques for Integer and Applesoft Basic, Apple Pascal, and 6502 assembly language, for graphics, education, utilities, and entertainment applications. A hands-on approach. $16.95. For spreadsheet users, *The Power of: SuperCalc*, by Robert Williams and Bruce Taylor, is the first in a series of *The Power of:* books. Written for business owners, accountants, teachers, students, and homeowners, it presents seven exercises for developing and expanding skills with *SuperCalc*. $9.95. And for users of the original *Calc*, *The Power of: VisiCalc*, volume I, by the same authors, presents a simple approach to learning how to use *VisiCalc*. It demonstrates the use of *VisiCalc* through specific examples. Volume II covers accounting functions and specific business applications. $9.95 each. All debugged with no place to go? *A Guide For Software Entrepreneurs* provides skills computer professionals need to venture out on their own. Dealing specifically with aspects of software, the book takes a business approach rather than a technical one. Planning and marketing are emphasized, with a large portion of the material devoted to showing how to bring out and sell your product. 208 pages. $32.50.

☐ **Logitech** (165 University Avenue, Palo Alto, CA 94301; 415-326-3885) announces the Depraz mouse, a cursor mover that's faster than a light pen or step keys. It fits in the palm of your hand and slides over a flat surface on a track-ball in its underside. Electronics sense the ball's rotation and guide the cursor on the screen according to the ball's motions. $295. Quantity prices available.

☐ Confused about Basic? Can't tell *for* from *next*? **Osborne/McGraw-Hill** (630 Bancroft Way, Berkeley, CA 94710; 415-548-2805) understands and presents *Armchair Basic*, by Annie and David Fox, as a possible answer. The book is written for the computer neophyte who has successfully resisted all conversion attempts made by computer enthusiasts. Chapters explore variables, data input, if-then statements, loops, random numbers, read-data statements, and subroutines. 180 pages. $11.95. In his book, *Some Common Pascal Programs*, author Greg Davidson converts the same programs in *Some Common Basic Programs* to Pascal. Finance, management decision, statistics, math, and science are among the subjects covered by seventy-six programs that have been tested and debugged and are ready for use. Program listings are accompanied by program descriptions and examples that aid users in following the logic of Pascal programming. 195 pages. $14.99.

☐ High school seniors might like to know that **Harcourt Brace Jovanovich** (757 Third Avenue, New York, NY 10017; 212-888-4444) is on their side. *Computer SAT* is the first software/textbook package for personal computers that leads the student through complete preparation for the all-important Scholastic Aptitude Test. The package diagnoses strengths and weaknesses, prepares a study plan, and guides the student through a comprehensive set of study exercises. The included book, *How to Prepare for the SAT*, contains four complete SAT practice examinations. $69.95.

☐ **Bourne & Co.** (City National Center, Suite 1350, Charlotte, NC 28202; 704-377-9109) unveils *Manna*, a complete membership and giving record system for small and large churches. The system maintains family and member files, profiles on members by activities and interest, Sunday school class rolls, and church gifts by special categories. $1,800.

☐ *The Wedge*, a spreadsheet program from **Systems Plus** (1120 San Antonio Road, Palo Alto, CA 94303; 415-969-7047), has now been enhanced with an interface program so you can bring in data from other programs into the spreadsheet. Provides a "help" menu to minimize the number of commands you have to remember. Supports fifty-two columns and four hundred rows and interfaces with most word processors. Allows split-screen formatting, insertion of rows and columns, and format changes. $195; CP/M version, $295.

☐ To let users turn any CP/M system into a Telex machine, **Advanced Micro Techniques** (1291 East Hillsdale Boulevard, Suite 209, Foster City, CA 94404; 415-349-9336) introduces *MicroTLX*. With it you can directly send and receive Telex and TWX messages; have a direct connection to TWX, Telex, and International Telex; enjoy automatic dialing and retry and unattended operation: send Mailgrams, telegrams, and overseas cables; and have access to news, sports, weather, and stocks, all updated hourly. $150.

☐ Here's some more from **Peachtree Software** (3445 Peachtree Road, N.E., Eighth Floor, Atlanta, GA 30326; 404-262-2376). *Calendar Management System* is a complete appointment scheduling and time management system designed for both individuals and large organizations. You can schedule and reschedule appointments, check for schedule conflicts, maintain suspense files of unscheduled tasks to be performed later, and store and recall notes relative to each appointment. The program allows the day to be divided into slices of ten, twelve, fifteen, thirty, or sixty minutes. CP/M is mandatory. $375. *Fixed Assets Accounting System* is designed specifically for capital-intensive business and public accountants wanting to enhance tax return filings. The system produces depreciation and supporting schedules for one or more companies. For individuals, it has separate schedules for up to three sets of books. It also incorporates recently enacted economic recovery laws, giving verification of asset file additions and appropriate reports by current and previous tax laws. Interfaces with Peachtree's *General Ledger*. Runs on CP/M systems only. $600.

☐ **Inmac** (2465 Augustine Drive, Santa Clara, CA 95051; 408-727-1970) introduces antistatic dust covers for the Apple III and Epson printers. Made of six-layer vinyl, the covers will dissipate static buildup before it can spark. They also protect against dust and food spills. $14.

☐ Update: If you can't find the *Apple Speller* anywhere, it's because it's been changed to the *Sensible Speller*, an improved spelling verification program. From **Sensible Software** (6619 Perham Drive, West Bloomfield, MI 48033; 313-399-8877). $125.

☐ **U-Microcomputers** (300 Broad Street, Stamford, CT 06901; 203-359-4236) has RAM cards to give you 320K and more for running large programs and spreadsheet models. All complement their existing 16K card. As many cards as there are free slots can be installed, giving you 566K for future use; software currently available can only use up to 320K. U-RAM 16, the 16K RAM board, $90; U-RAM 32, $140; U-RAM 64, $240; U-RAM 128, $380.

☐ The *Creator* by **Software Technology for Computers** (Box 428, Belmont, MA 02178; 617-923-4334) is a program generator for beginners or experts. No programming knowledge is necessary. The *Creator* generates Basic programs to your specification in minutes. Programs can be compiled, renumbered, edited, and merged right after generation. Your final program can include input and output routines, file structures, sort and search routines, dollar formats, and totals. Requires two disk drives. $200. Also from Software Technology are several business packages. *Advanced Payroll Package* will handle as many as 125 employees and print


THE TOP BANANA...
FOR HEROES ONLY.
GEBELLI
GEBELLI

payroll checks with year-to-date data. The package maintains commissions, bonuses, sick pay, holiday pay, overtime, and cash advances. It computes federal and state income taxes, FICA, and local taxes to be withheld. Prints W-2 forms, summary reports, quarterly reports, and employee lists. $275. The *IFO* package is a database that comes in three versions for three levels of capability. *Version I* has flexible report formats and handles such jobs as client listings, inventories, employee records, and membership rosters. $120. *Version II* incorporates phonetic search data retrieval and has faster access to information. Requires two disk drives. $200. *Version III* is a hard disk version offering even faster access than its predecessors and also has the phonetic search function. Requires Corvus 5, 10, or 20 megabyte hard disk. $450. The *Professional Time and Billing* program keeps records of who performed what and for whom. It provides records of fees received from clients or other receipts you may want to record and assembles a set of reports and generates statements for services rendered. Requires two disk drives. $395. *Apartment Manager* is a recordkeeping system for residential income property management. The program maintains financial and managerial data for up to six separate complexes. Complexes can be as large as one hundred thirty units and have as many as eight different types of units in each. It produces rent and expense journals and managerial reports and provides records of monies received, expenses paid, late fees, and vacancies. Requires a 132-column printer and two disk drives. $395.

☐ **Electric Courseware Systems** (Box 2374, Station A, Champaign, IL 61820; 217-359-7099) has released a series of music activity lessons. *Music Note Name Drill, Pitches on the Keyboard, Key Signature Drill*, and the ear-training game, *Ear Challenger*, are among those included. $39.95 each.

☐ New from **Gryphon Microproducts** (Box 6543, Silver Spring, MD 20906; 301-946-2585): *dBRx* is a package of machine language and dBase code functions that expand the capabilities of *dBase II*. You can now trim extra blanks from strings and perform trigonometric functions, exponentation, and square roots. Requires CP/M and *dBase II*. $150. *Pup-1*, a Pascal utility package, has the following features: transfer of Basic files to a Pascal disk, formatted listings of Pascal text files, upper and lower case, forty or eighty column formats, and support of Pascal wild cards. $39.95. In addition to *Pup-1* features, *Pup-2* moves Pascal text files to a Basic disk, displays and modifies any byte from a Pascal or Basic disk, and edits Basic programs with the Pascal editor. $39.95.

☐ Additions to the *Eureka Learning System* from **Eiconics** (211 Cruz Alta Road, Taos, NM 87571; 505-758-1696) include the *Shape Editor*, which allows course authors to create hi-res line drawings to be written into the *Eureka Learning System*; and the *Character Editor*, which provides facilities for creating special characters and symbols for incorporation into lessons. Both are included in the system at no additional cost. Updates are being sent out to current owners of the *Learning System*.

☐ Writing good documentation is a problem, and **Myown Company** (26919 S.E. 146th Avenue, Issaquah, WA 98027; 206-392-7873) offers help. Myown will prepare your documentation to your company's standards or will write it to theirs, keeping the user in mind. They also provide ideas, suggestions, and concepts for product modification and enhancement. Royalties arranged.

☐ **American Software Publishing** (1010 16th Street N.W., Washington, DC 20037; 202-887-5834) lets loose the *Freeloader 500 Software Library*, a collection of more than 2,500 programs categorized by subject and packaged for easy use. It's a compilation of public domain software written on double-sided disks. The *Library* comes in seven binders, each containing seven to ten disks. Categories include business and finance, utilities, graphics and sound, education, home, games, and adventure. Source listings of programs are also included. *Library*, $500; individual binders, $75; sourcebooks, $20.

☐ How many programmers does it take to change a light bulb? Ask **Mentat** (667 Rugby Road, Brooklyn, NY 11230), an amateur publication devoted to computers and programming. They publish everything they receive, but only those who contribute will receive copies. No subscription; it's free to contributors. Just send them forty copies of your manuscript.

☐ **Broderbund Software** (1938 Fourth Street, San Rafael, CA 94901; 415-456-6424) announces *Seafox*, a game in which you control a lone submarine out after a convoy of enemy ships and its escort. Dodge exploding depth charges, avoid menacing mines, and evade speeding torpedoes. And watch out for those darn clams! Keyboard, paddles, or joystick. $29.95. As if *Softalk* contests didn't raise the hackles on your neck enough, **Broderbund** announces a contest of their own, the exciting Arcade Machine Contest. Purchasers of the *Arcade Machine*, the do-it-yourself arcade design package, design their own games with help from the package. If you think you have a pretty good one, submit it for competition. Between January 31, 1983, and June 30, 1983, monthly winners will receive $200 worth of software or hardware. Monthly winners become finalists eligible for the grand prize of $1,500 worth of software or hardware. Details can be found on the *Arcade Machine* package. *Arcade Machine* required.

☐ **Micro-Sparc** (Box 325, Lincoln, MA 01773; 617-259-9710) has an upgraded version of their *AmperSoft* utility. *AmperSoft II* provides additional machine language routines to enhance the capabilities of Applesoft. Also provides useful extensions to Applesoft Basic and DOS while offering extra programming space for RAM card users. Several new features are included. *Print* allows fast and easy formatting and alignment for output. *Sort* offers super-fast sorting of numerical and string arrays. *Store/recall* stores and retrieves numerical arrays as binary files, saving disk space. *Matrix* does math functions at machine language speed. Requires RAM card of 16K or larger. $49.95. Another upgrade from Micro-Sparc is *LexiCom II,* their word processing utility that allows nearly universal transfer of word processing files. Converts *Super-Text* files to *Apple Writer I* format (or for any word processor that uses standard Apple text files), *Super-Text* or *Apple Writer I* files to standard text files, and the reverse of both functions. Also contains utilities for creating a *Demuffin* program, changing Applesoft files to text files and transferring picture files created by Muse's *Data Plot* to standard Apple disks. *RobotWar* files may be converted to text files for modem transmission. $39.95.

☐ Shalom aleichem! **The Institute for Computers in Jewish Life** (845 North Michigan Avenue, Suite 843, Chicago, IL 60611; 312-787-7856) offers more than thirty programs to teach Jewish culture. Subjects include the Hebrew language, Jewish history, Jewish music, and Jewish festivals and customs. Educational video games such as *Jewish I.Q. Baseball, Purimaze,* and *the Game of the Maccabees* are just a few offerings that give youngsters entertaining ways to learn about the Jewish culture. Also available is a series of musical programs designed to assist in bar and bat mitzvah training. Prices vary.

☐ Putting on your own business computer seminar can be easier with the *Computers for Business Management Seminar Kit* from **The Baron Company** (1301 Fraser A-11, Bellingham, WA 98226; 206-671-8708). The kit consists of a presentation script and notes, a sixty-four—step checklist for planning, advertising material, forty-six presentation color slides, hand-out sales material, and more. $495.

☐ **Pro/Pac** (Box 21900, Houston, TX 77218; 713-496-1179) has released *Financial Management Models for the Service Firm.* This series of eleven *VisiCalc* templates will assist the manager of service firms in fields such as engineering, advertising, architecture, and consulting. Helps in determining billing rates, estimate fees, forecast cash needs, budget projects, and other costs. Models are set up to be easily modified and adapted to the individual firm's needs. Apple III versions require 128K. $295.

☐ **Synergistic Software** (830 North Riverside Drive, Suite 201, Renton, WA 98055; 206-226-3216) has a few things for business. *The Inventory Manager* is designed with retail operations in mind. It can deal with 2,700 different inventory items on a two-disk system and 1,200 items on a one-disk system. Inventory items can be broken down into thirteen different categories of stock for better organization. It can also list as many as ninety-nine vendors who supply those inventory items. Additional functions: it generates summary reports on profit margins, calculates mark-ups, lists back order status, recommends reorder points, and prints purchase orders. $149.95. Next is *Word Weaver III*, a word processor for the Apple III that contains standard functions found in most word processors, and more. With it, output can be printed with various-shaped margins in geometric designs for advertising, greeting cards, and so on. You can display text on the screen in eighty columns and print in one hundred eighty-five columns. If additional memory is available, the program can use up to 256K. One single file holds fifteen to twenty pages of text on 128K machines. *Word Weaver III* text files are compatible with other word or data processing programs. $99.95.

☐ Memory expansion for *VisiCalc* users is here. If you have either the Legend 128KDE or 64KC RAM expansion card, you can add the *VC-Plus* program from **Legend Industries** (2220 Scott Lake Road, Pontiac, MI 48054; 313-674-0953). *VC-Plus* gives the Apple II more power than the III at a fraction of the cost. It also offers you a choice of forty or eighty columns of workspace. The program is free with the purchase of Legend's 128KDE, 64KC, or 64KDE RAM card. $20. You have eight slots and nine cards; what will you do? Legend Industries suggests Soft 8, a card that plugs into slot 7 and provides both slots 7 and 8. Switching between slots is software driven, allowing you to switch between one card and another with software commands. You can PR#8, IN#8, or catalog yourself around. $84.95. Legend also announces the availability of the Pascal Super System. The system supports the simulation of fast access disk drive units for storage and retrieval of standard Apple Pascal files with the added speed of the 6809 Stellation Mill. Consists of a Legend 128KDE RAM card, a Stellation Two 6809 Mill, and a *Superl* disk. The system is available from either Legend Industries or **Stellation Two** (Box 2342, Santa Barbara, CA 93120; 805-966-1140). $749.

☐ *Shuttle Intercept*, the second in a series of arcade games from **Hayden Software** (600 Suffolk Street, Lowell, MA 01853; 617-937-0200), puts you in a daring rescue mission in space. Your spacecraft is directed to retrieve friendly satellites bearing vital data and must fight or avoid enemy craft, satellites, missiles, and meteors (enemy meteors?). Four quantum levels of play with hyperspace in between the first three. $34.95.

☐ **Hayden Book Company** (50 Essex Street, Rochelle Park, NJ 07662; 201-843-0550) releases *Basic Computer Programs for Business*, Volume Two, by Charles D. Sternberg. It's a collection of programs written for anyone using a business system. The book includes more than seventy complete programs which are grouped in four sections ranging from marketing and sales to planning personnel recording and analysis. All programs are completely modifiable to meet specific needs. A collection of statistical programs allows data to be analyzed and compared. 384 pages. $13.95.

☐ If you send lots of electronic mail, Visionary 100, a modem from **Visionary Electronics** (141 Parker Avenue, San Francisco, CA 94118; 415-751-8811), might be helpful. It has memory to store messages, software to send and receive messages automatically, and a clock and calendar to control when they should be sent. Stores memos for multiple distribution at a later date and will keep redialing busy numbers until the message is successfully sent to all. $595.

☐ Kids can do more than play games and do math problems on their Apples. **Circadian Software** (Box 1208, Melbourne, FL 32901) offers *Kids Stuff-I* and *Kids Stuff-II* as interactive programs that teach the child fundamentals of Basic programming in a manner that requires no computer background and no manuals to wade through. The future programmer will acquire skills by developing several simple programs in a fun, painless learning process. *Kids Stuff-I* and *II* can be mastered by the average twelve-year-old and are also effective for the above average ten-year-old. Both require game paddles; neither requires a child. $39.95 each.

☐ GraForth now supports the Hewlett-Packard 7470 Plotter, and **Insoft** (10175 S.W. Barbur Boulevard, Suite 202B, Portland, OR 97219; 503-244-4181) has just completed the documentation for this interface which provides 3-D graphics plotting and multiple character sets and sizes. Requires GraForth, the 7470 Plotter, and a standard RS-232 interface. $25.

☐ **GameMaster** (1723 Howard, Room 219, Evanston, IL 60202; 312-328-9009; modem, 312-475-4884) is the time-sharing network that lets you play games with or against other people. In addition to war, dungeon, adventure, and board games, the system offers electronic mail, on-line conferencing, and a number of other BBS services. Now with new rates. $40 to join. Hourly rates: weekdays midnight to noon, $2.40; noon to 6:00 p.m., $3; 6:00 p.m. to midnight, $3.60; weekends and holidays, $2.40.

☐ A Fortran documentation guide has been created by **Associated Technology** (Route 2, Box 448, Estill Springs, TN 37330; 205-837-4718) to help software developers formulate their own standards. Provides methods consisting of a set of company standards and examples. 57 pages. $24.

☐ *Stock Momentum Studies* is a graphic analytical program from **Troy Folan Productions** (29 Miller Road, Wayne, NJ 07470; 201-694-5618). Based on price movement, the program lets the user chart any stock, commodity, or market index in a variety of modes such as differentials, percentage changes, moving averages, and exponential averages.

Can show two modes overlaid on one screen. Works with time frames of a few weeks up to a full year. It also includes a sort program that will put data for as many as 350 entries into chronological order quickly. $525.

□ **Portware** (5724 Tucker Lane, Edina, MN 55436; 612-933-3510) announces the first in a series of Portaids programs for investors. *Income Records for Securities* (IR/S) helps investors maintain detailed records of dividends and interest income and support their reporting under new tax laws. The user can develop a complete and detailed journal of dividends and interest items received during the year. A summary report by security holding is always current and immediately available for supporting tax schedule B. The program separates nontaxable and taxable security income. Codes can be defined by categories such as types of bonds, types of stock, and broker. $29.95.

□ Star gazers can have an observatory on their screens with help from *Celestial Basic*, by Eric Burgess. With this book from **Sybex** (2344 Sixth Street, Berkeley, CA 94710; 415-848-8233, 800-227-2346), the amateur astronomer will be able to observe changes of hours and years in the configurations of the night sky. Simple programs in the book help alleviate calculations for everyday observation. 228 pages. $13.95.

□ If you have trouble remembering what your *VisiCalc* equation headings represent, the *VC-Documenter* from **Desktop Solutions** (Box 35659, Houston, TX 77235; 713-723-6170) spells it out for you in English. What used to look like >B3:+B1−B2 would instead be listed as row and column headings defined by the user, such as 1982 GROSS=+1982 SALES−1982 COST. The program can also list by row/column or by column/row, and has the option to print all or part of a model. There's no limit to the size or complexity of the model it will handle, and it runs independently of *VisiCalc*. $49.95.

□ The latest from **Quark Engineering** (1433 Williams, Suite 1102, Denver, CO 80218; 303-399-1096): *Catalyst* will let you boot from a hard disk on your Apple III. No more disk swapping and rebooting when you want to go from one program to another. Simply boot the *Catalyst* disk whenever you turn on the computer. *Catalyst* will display a menu of available programs which can be loaded instantly from the hard disk. Key sequence returns you to the menu. $149. Also for III users, *Discourse* will send printer output to disk rather than to the printer. Once the document has been put to disk, you can use the computer for other tasks. *Discourse* routes the output from the disk to the printer without interrupting you. Queues up to fourteen reports for print-out. $125. Finally, *Legal Dictionary* is an accessory to *Lexicheck*, the spelling checker for the *Word Juggler* word processor. It adds 8,000 legal terms to the basic 30,000 word *Lexicheck* dictionary. $85.

□ **Scientific Research Associates** (155 North Wacker Drive, Chicago, IL 60606; 312-984-7053) has a courseware program that introduces the user to Basic. For use at home or in high school, *Discovering Basic* is designed to teach fundamental programming concepts and to show how to write simple programs in Basic. The workbook contains nine chapters in problem solving, data storage, input and output, programming structures, loops, and subroutines. Features colorful graphics, pencil-and-paper exercises, and quizzes. $175.

□ Impress friends and liven up parties with *Kaleido-Sound*, a real-time graphics program from **Passport Designs** (116 North Cabrillo Highway, Half Moon Bay, CA 94019; 415-726-0280). The audio signal from a home stereo, cassette player, or sound system can be plugged into the Apple II cassette port to drive four different kaleidoscope patterns. The full-color kaleidoscopes change color, pattern, and location on a video screen as the music changes frequency and loudness. Includes connector cable. $39.95. Passport Designs also adds to software for the Soundchaser Computer Music System with its addition of *Turbo-Traks*. Included is a sixteen-track digital recorder that simulates an analog tape deck in software, variable number of oscillators per voice (up to sixteen), sync to tape or drum machine, and extended recording time. $150.

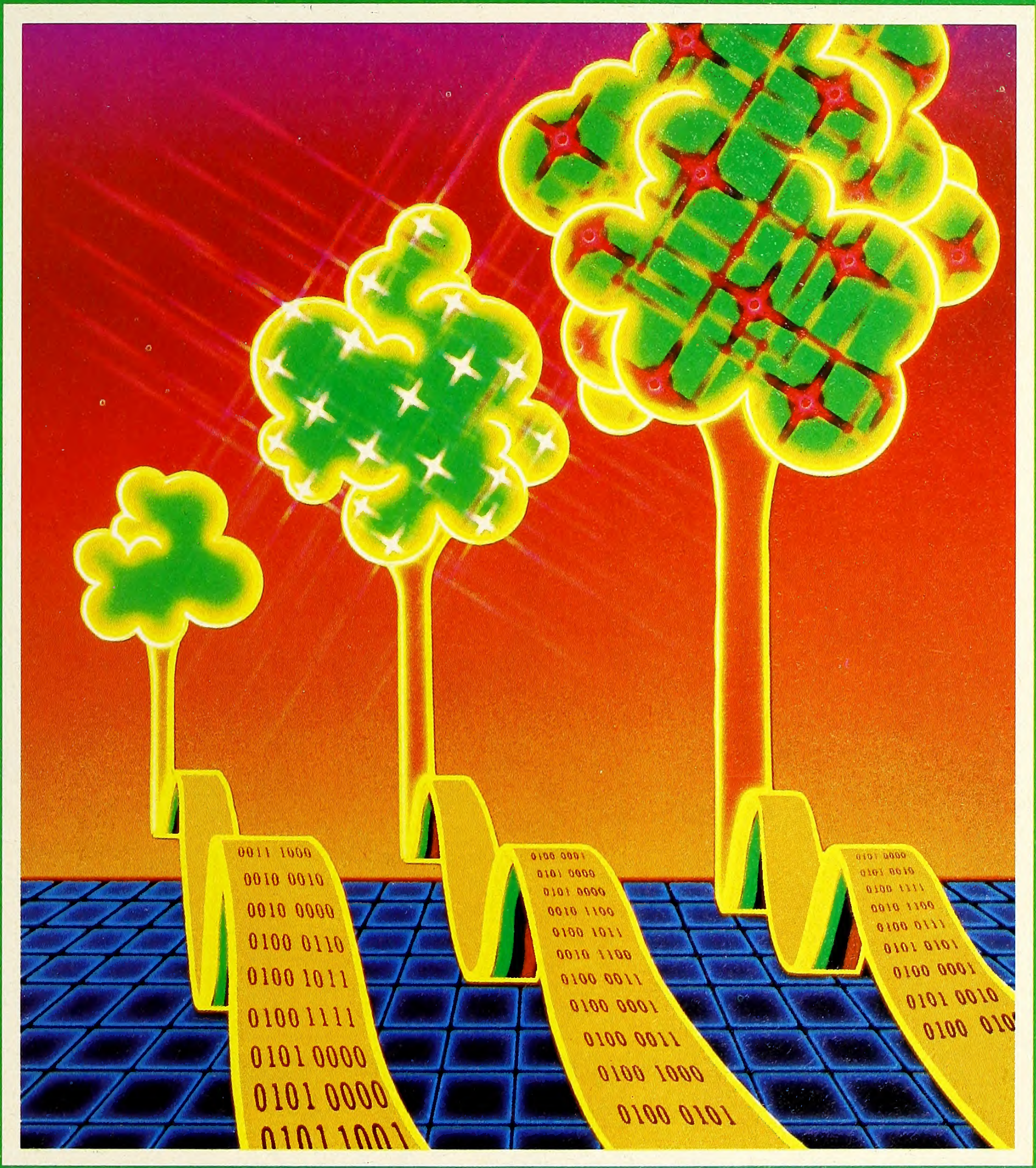
□ Announcements from **Single Source Solution** (2699 Clayton Road, Concord, CA 94519; 415-680-0202): You can sail any sailboat in the world on your Apple. *Comsail I* simulates a fast thirty-foot sailboat sailing in near-ideal conditions around a plotted course. Sailors in *Comsail* deal with the rudder, main sail, jib, wind angle, and elapsed time. Sailing dynamics that are present include tacking into the wind, luffing, and critical angle. If you're blowing it, help and sailing tips are at your command. $59.95. High school yearbook staffs might take a look at *Yearbook*, a tutorial program that emphasizes the basic principles and common technical practices in publishing yearbooks. It deals with layout, editing, and vocabulary. Hi-res graphics help you along. Have a nice summer; see you next year. $59.95. How many liters to a quart? Centimeters to an inch? *Conversion* is a program that performs more than a thousand common and not-so-common unit conversions for professionals involved in businesses requiring such functions. $29.95. *Extax* is a *VisiCalc* template designed for professions involved in estate planning. Attorneys, life insurance agents, certified public accountants, and financial planners can use this program to compute projected estate taxes quickly and accurately. It lets the user plan for variable normal life expectancy, assets growth, charitable gifts, and distribution of assets. Requires *VisiCalc 3.3* and 64K RAM. $49.50. *Bowling Secretary* provides a means of maintaining bowling averages, league standings, and team assignments for any number of bowling leagues. Includes a built-in word processor for writing personalized letters. It's menu-driven and has error detection and correction. $49.50.

Single Source Solution presents its entries in the accounting area with the KM Systems Accounting Package. The first in the series is the *KM Systems General Ledger*, a flexible method for keeping multiple sets of double-entry books of account. It's menu-driven and incorporates the Ampersort routine. $39.95. The *Busypro Accounts Receivable System* is the first in the Busypro series. It allows the user to acquire optional report generator and query utilities that will produce customized reports. Supports debit and credit memos, write-offs, and adjustments and maintains customer and sales territory master files. The system is protected with passwords for security of data. $149.50. *Statistical Modelling* includes a linear programming module and a regression analysis module. The linear programming module is limited to fifty constraints and one hundred variables; the regression analysis module is limited to twenty independent variables and one hundred observations. It also calculates regression coefficients, standard error of coefficient, T-value for coefficients, and standard error of the estimate. $39.95. *Usort* is a sort routine for Pascal programmers. It can be made part of menu-driven software and eliminates having to write a sort program for every record type. No recompiling or relinking is needed. Sort key fields may be integer, long integer to twenty digits in accuracy, character, Boolean, and packed arrays up to twenty-nine characters. $49.95.

Now for labels. *Superlabels* is a customized label printing system. Each line of the label may be given a different appearance using the compressed, expanded, or overstrike capabilities of matrix printers. $39.95. Single Source Solution hasn't forgotten CP/M users. *EPrint* and *MSIPrint* are printer programs that utilize the full capabilities of the Epson MX-80 or MSI printers respectively. Users have full command of compressed, doublestrike, and emphasized printing, with selectable vertical and horizontal tabs. CP/M is required, natch. $24.95. *Nums* is a tutorial program for teaching the various numbering systems that computers use and how to become adept at handling them. Covers binary, octal, and hexadecimal systems. You need CP/M for this one, too. $29.95. *Members* is a mailing list management system that allows fast sorts alphabetically by zip code and by as many as six other precoded variables. Requires (you guessed it) CP/M. $49.50.

Now, for teachers, *Student Record System* is a file management system to help in class management of students and their grades. It can keep track of ten separate grades for one hundred students and automatically computes the mean and standard deviation for each graded assignment. Sorts as many as ten assignments. Requires CP/M or UCSD Pascal system. $69.50. Single Source Solution offers *Upapple!* to increase the capabilities of the *Apple Writer* text editing system. Features include backspacing, word and character underscore, word and character overstrike, and text-embedded printer control and format commands. Also includes curly and square brackets, circumflex, backslash, "at" sign, and left single quote characters. $29.95. And finally, *SIBasic* (structured interpretive Basic) is an extension to Apple Basic interpreters. It allows nested macros, symbolic labels, deletion of remark statements, no line numbers, and ability to organize output at a specified line number. Commands include case of, if-then-else, while, repeat-until, do, and print-using. Also has specialized error messages. $49.95.

# MEGABYTERS
## Make Apples Grow




SVA developed the First 8" floppy controller in 1979, the First 256k RAM Card in 1981, and now introduces PRO-GUARD, the First 8" floppy controller for Apple *III*.

We offer the most complete line of Memory Expansion Systems available and have a worldwide reputation for Reliability, Compatibility and Support. Every product undergoes a seven day burn in period and is retested prior to shipment.

Our standard Apple DOS*, CP/M* and Pascal* distribution software provide maximum application flexibility, and our full-time Customer Relations Department is ready to assist you with start-up and maintenance requirements. SVA means business. For more information call or write today.

SVA

*Trademarks: CP/M-Digital Research, Apple Dos-Apple Computer, VisiCalc-VisiCorp.

**SORRENTO VALLEY ASSOCIATES, 11722 SORRENTO VALLEY ROAD, SAN DIEGO, CA 92121-1084 TELEPHONE (714) 452-0101 TWX 910-335-2047**

### ZVX4
- 8" Floppy Controller
- Single/Double Density Media
- Single/Double Sided Drives
- Automatic Density and Sidedness Detection
- IBM Standard Formats
- Corvus, Amlyn Compatible
- Autoboot DOS, Pascal, CP/M

### APP-L-CACHE
- 64k, 128k, 192k, 256k models
- Expands VisiCalc* worksheet to 252k
- Built-in Language Card
- Needs only 1 slot
- Disk Emulator Mode
- Speed of Hard Disk
- Eliminates need for second mini-drive
- Low Power Consumption


### PRO-GUARD
- Apple III 8" Controller
- Increases online Storage up to 2.2 megabytes
- Backup Apple Profile
- DOS, SOS, Compatible
- IBM Standard Formats
- Under SOS, data transfer speed approaches hard disk

### AMS 5000
- 5¼" Winchester Turnkey System
- 5, 10, 20 megabyte models
- Compatible with ZVX4 8" Controller
- Network up to 127 Apples
- User selectable storage allocation for DOS, CP/M, Pascal
- Apple III compatible

### AMS 8000
- Twin Drive Turnkey System includes ZVX4 Controller
- Interchange Data with other Computers
- Increase online storage up to 4.4 megabytes
- Use for Hard Disk Backup
- Includes DOS, CP/M, Pascal Distribution Software

Become a millionaire! A billionaire! A trillionaire! No difficult work involved, no long hours, no education necessary. All it takes is courage—a lot of courage plus swift reactions and quick thinking.

The money you want is right there in front of you all those beautiful $$$ signs that represent tremendous wealth. Just run through the maze and grab it as you go. Simple, right? Sure it is. But, keep a sharp lookout because the Money Munchers' gang want that loot for themselves.

Outwit the mean Money Munchers and get to upper levels and what happens? Why the real trouble begins . . . because now there are special nasty Guardians to help the Money Munchers. While the munchers pick up the bucks, the Guardians follow your trail—and if you let them catch you you'll only be remembered as the richest man in the cemetary.

Ah, but the riches that can be yours if you reach the top level, alive! Yes, this is your chance to collect the kind of $$$ you've dreamed about. All it takes is courage . . . and a great survival instinct!

# Beware the MONEY MUNCHERS

WHOLE LOTTA FUN

For the most maze fun you've ever had **and thousands of random mazes,** get MONEY MUNCHERS today—be a millionaire tonight!

**by Bob Bishop**

$29.95 for the Apple II*
At computer stores, or:

DATAMOST INC

9748 Cozycroft Ave., Chatsworth, CA 91311. (213) 709-1202

VISA/MASTERCARD accepted. $2.00 shipping/handling charge.
(California residents add 6½% sales tax.)

*Apple II is a trademark of Apple Computer, Inc.

# SOFTCARD Symposium

by Greg Tibbetts


Welcome to the November edition of SoftCard Symposium. This month we'll continue our in-depth reexamination of the CP/M operating system as implemented on the SoftCard. Specifically, we'll begin a step-by-step examination of the booting process: what comprises it, how it works, and so on. Last month we got a brief overview of the subject in the context of describing the complete system; now we'll get down to details.

In CP/M, the booting process is broken down into two distinct types, a cold boot process and a warm boot process. These processes are then broken down even further into single modules, each of which completes a subprocess that's necessary to bringing up the system. Since these modules are themselves made up of routines with their own individual functions, our definition of what constitutes a module and our explanation of why we treat a particular collection of routines as a module may sometimes seem arbitrary. However, in each case it will become apparent that the module defined is capable of operating more or less independently as a single unit, while the individual routines that form a particular module are dependent on one another to perform their functions. Treating these modules in their chronological sequence offers the clearest means of describing them since, in many cases, those modules that execute later on in the booting process are totally dependent on the ones that precede them.

In order, then, the modules are:

1. Apple Disk II controller boot ROM
2. boot sector
3. cold loader and Z-80 finder
4. warm loader
5. Z-80 cold boot code
6. Z-80 warm boot code

These names are intended to be descriptive only and aren't necessarily recognized officially. We will examine each of the modules in order. This month, so as to cover the subject in proper detail, we'll consider only the 6502 modules, items 1 through 4. We'll save the Z-80 modules for December's column.

**Sweet Little Sixteen.** The first module is the standard boot ROM on the Disk II controller card. Note that we'll deal only with the sixteen-sector version of this ROM because it is the latest and the most widely used. In this version of the ROM, the Apple disk is divided into thirty-five tracks of sixteen sectors each. Each sector contains 256 bytes of usable data that have been translated into 342 specially encoded disk bytes. As the disk spins under the read/write head of the disk drive, a separate ROM, called a *state machine*, reads and writes the actual bits stored on the disk. The translation of encoded disk bytes to usable data bytes and vice versa is handled by software that accesses the state machine when more data is ready to be read or written.

The theory of operation of the state machine is complex and beyond our scope in this case. Luckily, though, we can treat the state machine as a "black box," such that when the software calls for an encoded disk byte, the state machine supplies it from the disk. Conversely, when the software wishes an encoded byte written, it passes the byte to the state machine and the state machine writes it to the disk.

Besides reading and writing disks, the state machine controls the disk drive motor that spins the disk and the stepper motor that moves the head from track to track—all in response to software commands. Note that all of the actual work is done by the state machine, whether it's operating under the control of software in the computer's memory or under the control of the software in the boot ROM on the controller.

**Meet Your Mapper.** The communication required to pass bytes back and forth and activate the various mechanical disk processes through the state machine is handled in the manner described last month as *memory-mapped I/O*. In this system, external devices have access to the lines comprising the computer's address bus, and the circuitry is designed in such a way that when the computer reads or writes certain memory locations, it is in fact communicating with the external device. Each slot, except slot 0, is allocated one 256-byte page of memory in the $C100 to $C7FF memory range. The address of this range is $C*n*00 to $C*n*FF where *n* is the slot number.

In addition to that page of memory addresses, sixteen more addresses are allocated to each slot (including slot 0) in the $C080 to $C0FF range, such that $C080 to $C08F are for slot 0, $C090 to $C09F are for slot 1, and so on. How these 272 memory locations are used by a peripheral is up to the designer of the card. In the case of the Disk II controller, the page of addresses is used simply as memory (in this case, read-only memory) corresponding to the 256 bytes contained in the boot ROM. The other sixteen locations, however, are used as control addresses for the state machine. Accessing one of the control addresses for the slot occupied by the controller causes the state machine to perform a function. For example, turning the drive motor on requires an access of address $C089+$*n*0 where *n* is the slot number. For slot 6, this address is $C0E9. There are sixteen such functions for the state machine activated by means of the sixteen control addresses assigned to the slot.

**Head of State.** The boot ROM itself is comprised of 251 bytes of 6502 machine code. Its primary function is to operate the state machine and read track 0, sector 0, into memory page $08 ($800 to $8FF). Parts of the boot ROM can also be used as a general-purpose disk-read subroutine, as we will see a little later.

When executed from the beginning, the boot ROM first builds a table of values in memory page $03 that are needed to decode the disk data bytes read by the state machine. Next, by calling the address of an RTS instruction in the Monitor ROM and then looking at the return address created on the stack, the boot ROM determines the slot in which the controller card is located. That information is necessary before any activity can be performed with the disk drive. This is so, of course, because the control addresses for the state machine are slot-dependent.

Once the table has been set up and the slot determined, the boot ROM module proceeds to initialize the state machine. Among other things, it enables drive 1, turns on the drive motor, waits for the motor to come up to speed, and recalibrates the read/write head position by seeking track 0 (against the mechanical stop) several times. This latter operation causes the rattling sound you hear on boot-up.

Once everything is in readiness, the boot ROM starts reading bytes from the disk, using the state machine. It checks for the pattern of bytes that indicates the start of a sector address field—that portion of the sector that indicates which of the thirty-five possible tracks and sixteen pos-

sible sectors it is. As the boot ROM finds each address field, it loads the bytes corresponding to track and sector numbers and compares them with the values of the track and sector being sought. The values it compares the disk data against are not in the ROM at this point but are kept in temporary locations in memory page $00, which it initialized at the beginning of the ROM. This is an important point since, if these temporaries are set to some other value and the beginning of the ROM is bypassed, this same code can be used to load tracks and sectors other than 0.

Once the appropriate sector of the disk is found, in this case, sector 0, byte after byte is loaded from the data field of the sector and translated using the table in memory page $03. As each byte is translated, it is written to a page of memory whose value is contained in still another temporary location. This location was initialized in the beginning of the ROM to $08 so that sector 0 will load from $0800 to $08FF. After this sector has been loaded, the temporary location containing the load address page is incremented and the byte now stored at $0800 is checked. This byte, which was the first one on the disk, indicates to the boot ROM how many sectors to load before ceasing operation. With SoftCard, this value is 01, meaning load the single sector at $0800 and then exit. On exit, the boot ROM always branches to location $0801 to continue execution. Location $0801 marks the beginning of our next module.

**Sector Skewing Speeds Things Up.** Module number two, which we've named the *boot sector*, consists of forty-five bytes of 6502 code and sixteen bytes of data. The data bytes in this case are a sector interleave table. The term *interleaving*, also called *sector skewing*, describes a process whereby sectors on the disk are not stored in the sequence they will be read. This is done because the software is seldom fast enough to process all the data just read from the last sector in time to read the next sequential sector on the disk as it passes under the head. If sectors were stored sequentially, an entire revolution would be required to bring the sector back under the head; consequently, only one sector per revolution could be read in. When an interleave factor is used, the next sector to be read is placed two or more sectors away, allowing time for the software to finish its task and be ready to read again when the sector comes by the head. In this way, as many as eight sectors can be read on a single revolution, resulting in a possible eightfold increase in disk access speed.

There are two types of interleaving, *physical interleaving* and *software interleaving*. With physical interleaving, the actual sectors on the disk are numbered out of sequence, as in the series:

0,8,1,9,2,10,3,11,4,12,5,13,6,14,7,15

This is an example of a two-sector interleave, and in such interleaving the sectors would actually be written in this order during the format process. When you want to simplify the code required for reads and writes, this method is easiest, since reading a particular sector only requires that the software look for the proper sector address mark.

**Double Trouble.** Software interleaving is somewhat more complicated, however, and two sets of numbers must be taken into account when dealing with it. First, there are the actual sector numbers appearing in the address field. This set we'll call the *physical sector numbers*. The other set of numbers corresponds to the order in which we wish to read or write the sectors, and these we'll refer to as *logical sector numbers*. In software interleaving, the physical disk sectors are still sequential on the disk, that is:

0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15

To make it possible to use this method, a table is set up in the program that lists the physical sector numbers in the order they have to be read or written to correspond to the logical sector numbers. To form a two sector interleave, we would have to write first in sector 0, then sector 2, then sector 4, sector 6, sector 8, and so on. By making up a table of this sequence, we can use it to tell where to find our appropriate logical sector. The table is:

0,2,4,6,8,10,12,14,1,3,5,7,9,11,13,15

Thus, when we wish to find the fifth logical sector, we look at the fifth entry in our table (remember, we begin with zero) and we see that it is physical sector 10. Using this method, we can maintain the proper distance between logical sectors on the disk, making fast disk access possible.

Apple disk systems to date have used the software interleaving method; SoftCard is no exception. In SoftCard CP/M, the first three tracks (which contain the system) have an interleave factor of two and the interleave table is exactly like the one just given. Two is a sufficient interleave factor for the system tracks, since very little processing is done between reads just to load those tracks into memory, and therefore only a small amount of time between reads is required. The data portion of the disk has an interleave factor of three, given that with applications software, the processing required between disk reads is more extensive, requiring more delay between reads and therefore more separation between sectors.

**Move Over, Rover.** The code in the boot sector is basically very simple. Its function is to load the next nine logical sectors from track 0 into memory beginning at $0A00. It accomplishes this by using the read routines in the boot ROM beginning at $C*n*5C and altering the memory page $00 temporary locations used by the boot ROM to store sector number and load address values.

The boot sector module can be kept simple because the boot ROM handles all the necessary state machine access and byte translation and because the nine sectors are loaded sequentially from $0A00 to $13FF. These sectors consist of the Apple RWTS routines ($0A00 to $0FAC), which are the much more complex state machine access routines that allow us to read and write to the disk. They are the warm loader ($0FAD to $0FFC), which will use RWTS to load the remainder of CP/M; the cold loader and Z-80 finder ($1000 to $1188), which is our next module; and finally the IOCB ($1200 to $13FF), which will be moved down to memory pages $02 and $03 for final use by the system. The IOCB is loaded high because loading at memory page $03 is difficult (remember this page is used to store the byte translation table the ROM uses when reading the disk). At the completion of this nine-sector load, the boot sector branches to the cold loader and Z-80 finder at $1000.

**Clean-up Time.** The cold loader has several functions, but its primary one is to identify the system hardware. This is accomplished by examining the contents of the 256-byte page of memory allocated to each slot. Before that process begins, however, some housekeeping items are taken care of. First, any RAM board in the system is set to a ROM-read, RAM-write condition. This means that when the processor attempts to read memory in the $D000 to $FFFF range, it will read the ROMs on the Apple motherboard; but when attempts to write to this area of memory are made, it will write into the RAM on the RAM board. In this way, the routines in the Apple Monitor ROM can be executed by this and subsequent modules, but, at the same time, if a RAM board is present in the Apple, necessary CP/M system data can be written to that RAM.

Next, the drive motor is turned off since no further reads will be done using the boot ROM and future disk access will be through RWTS, which will control the drive motor itself. The third step is to initialize two temporary track-pointer values in page $00. These will be used by RWTS so that no annoying recalibration of the heads will need to be made on the first disk access while RWTS locates the head positions. The fourth housekeeping function is to call three subroutines in the Monitor ROM at $FB2F, $FE93, and $FE89, in that order. These three routines set and initialize text mode, set output to the video screen (PR#0), and set input from the keyboard (IN#0), respectively. Fifth, the stack pointer is initialized to its unused value, clearing any previous items on the stack. At this point, everything has essentially been brought to the power-on/reset condition. Finally, the boot slot value, which has been stored in the [X] register, is checked to ensure that CP/M was booted from slot 6, since SoftCard CP/M makes assumptions based on that premise. If not, the message *must boot from slot six* is printed to the forty-column screen via the Monitor COUT subroutine at $FDED and, when complete, the loader branches to the general Monitor entry point at $FF65. If booting did occur from slot 6, the loader continues its process.

**Z-80 Shuffle.** The loader's next step is to perform several memory moves. The first of these is to move a short Z-80 program originally loaded from $1169 to $1176 to address $1000, overwriting the first part of the loader itself. This, of course, is a 6502 address, as are all those we've mentioned so far, but it is significant since $1000 to the 6502 is the Z-80 address 0000H. (Note: To avoid confusion, we will continue to show all 6502 addresses in the form $xxxx and all Z-80 addresses in the form *xxxx*H.)

When activated for the first time, the Z-80 always branches immediately to location 0000H. This program is preparation for turning on the Z-80 and provides the means for the Z-80 to recover when activated. The next two memory moves bring the IOCB down to its final location as we saw earlier, and once the moves have been completed, three variables are initialized.

The first of these variables is the IOCB variable at $03B8, which represents the number of disk drives in the system. It is initialized to 0 since the module will go on to identify dynamically the number of drives (by finding the number of controllers and multiplying by two) as part of its hardware identification process. The second variable is a page $00 temporary location we'll call ADDR1 which is used in the identification process, as we'll see later. It is also set to 0. Finally, the third variable, also a page $00 temporary, we'll call FLAG1; this variable is used as an indicator of the presence of the Z-80. It, too, will be discussed later, but for now it is initialized to 255.

**Ready, Steady, Go!** Now that everything is in the correct place and variables have been initialized, the loader begins execution of its Z-80 finder portion. This general-purpose routine not only finds the Z-80 but also identifies all recognizable peripheral cards. The finder starts at slot 7 and proceeds downward, examining each slot in the same way by systematically inspecting various parts of the 256-byte page of addresses allocated as I/O area for that slot.

First, the ADDR1 variable and the byte following it (ADDR1+1), which are used together as a two-byte address vector, are set to the value of the first byte of the I/O area. Since ADDR1 has already been set to 00, the finder simply sets ADDR1+1 to the page number of the slot, that is, $C*n*.

Its next step is to alter itself, changing the address field of a memory-write instruction to the same address that is contained in ADDR1 and ADDR1+1. The finder then attempts to execute the memory-write instruction. If the card in the slot is a SoftCard, this write will activate the Z-80, causing it to execute the Z-80 routine we placed at 0000H earlier. That routine performs two functions: first, it changes FLAG1 from 255 to 00; and second, it uses the value at ADDR1+1 to perform another write to the SoftCard to return control to the 6502, which will pick up where it left off in the finder code. At that point, the finder loads the value at FLAG1, and if it is 00 rather than 255, it knows it found the SoftCard and goes on to place the SoftCard's slot address in two places where the system will use it as part of normal operation. If FLAG1 is still 255, then the Z-80 code did not get executed and therefore the card in that slot is not a SoftCard. If the SoftCard was found, the finder code goes on to the next slot; if not, it continues its examination.

**What's the Slot Got?** The next step is to determine whether the slot holds any card that contains ROM. This is done by two successive reads of all 256 bytes in the slot's I/O space, creating a checksum of the values. A checksum is simply some mathematical process performed on a series of values that results in a single number. In this case, each byte is added to a running total, creating a two-byte sum.

Doing a checksum of two sets of numbers will show within reasonable limits if the individual values are identical. If a ROM is there in the slot, both reads will find the same string of values and create the same checksum value. If no ROM is present, the values will change from one read to the next and create a different checksum. You can verify this by using DDT to read an empty slot's I/O space several times. If no ROM is found, a value of 00 is placed in the SLTTYP table at $03B9 plus the slot number in the IOCB. If ROM is found, the identification continues.

The next step involves the protocol established by Apple Computer for peripheral card manufacturers. It involves looking at the two bytes at $C*n*05 and $C*n*07 and comparing them to a table of values for various types of cards. Disk controllers have values of $03 and $3C respectively, Apple Comm cards have values of $18 and $38, Apple parallel printer cards have values of $48 and $48, and, finally, Apple serial cards have values of $38 and $18.

Although all of these cards contain firmware (ROM-based software) that can be used to control them, knowing which card you're dealing with makes it possible to operate the card independent of its firmware by accessing the control addresses for its hardware components—just as RWTS operates the disk controller without using the firmware in the boot ROM. Other manufacturers' cards that use the same control addresses and hardware and therefore can be manipulated in the same way as Apple cards can be recognized as one of the types we've been discussing by following this protocol.

**Just Your Type.** In a revision to the standard (which occurred after the SoftCard system was written), Apple added an additional category, called a Type 6 firmware card. This card type was designed to give more freedom to peripheral manufacturers whose cards do not fit the categories we've just mentioned or for cards that do fit the categories but have a different control address structure or hardware components. Identifying a card as a Type 6 opens up methods by which system software can identify entry points in the card's firmware to branch to and from that perform the card's various functions. A Type 6 card looks, at $C*n*05 and $C*n*07, just like a serial card. Separating the two cards requires examining further all serial cards at $C*n*0B.

Since SoftCard preceded this new standard, it doesn't recognize the Type 6 card, and since the two bytes examined show the card to be a serial card, SoftCard CP/M will attempt to deal with it as a serial card. A new version of SoftCard CP/M scheduled for release soon corrects this problem, but for now, Type 6 cards such as the Epson 100 parallel printer interface will be identified incorrectly and will not work. In the case of most standard cards, however, the finder code will properly identify them and set the bytes in the SLTTYP table in the IOCB to the proper values.

**Missing in Action.** If after examination of all the slots no SoftCard is found, the message "can't find Z-80 SoftCard" is printed to the forty-column screen via COUT and the finder branches to the Apple Monitor. If the SoftCard is located, however, the finder code goes on to its next task and initializes a series of vectors in memory. These vectors occupy the last sixteen bytes of memory page $03 in the IOCB. We discussed these in an earlier column, referring to them as the Apple break, reset, Applesoft ampersand, Monitor control-Y, nonmaskable interrupt, and interrupt request vectors. Together with three vectors in high memory, $FFFA to $FFFF, they provide the means to recover from such things as interrupts and the pressing of the reset key. In the Apple II, these events cause an automatic (and nonstoppable) branch by the 6502 to the address contained in $FFFA and $FFFB (nonmaskable interrupt); $FFFC and $FFFD (reset); or $FFFE and $FFFF (interrupt request).

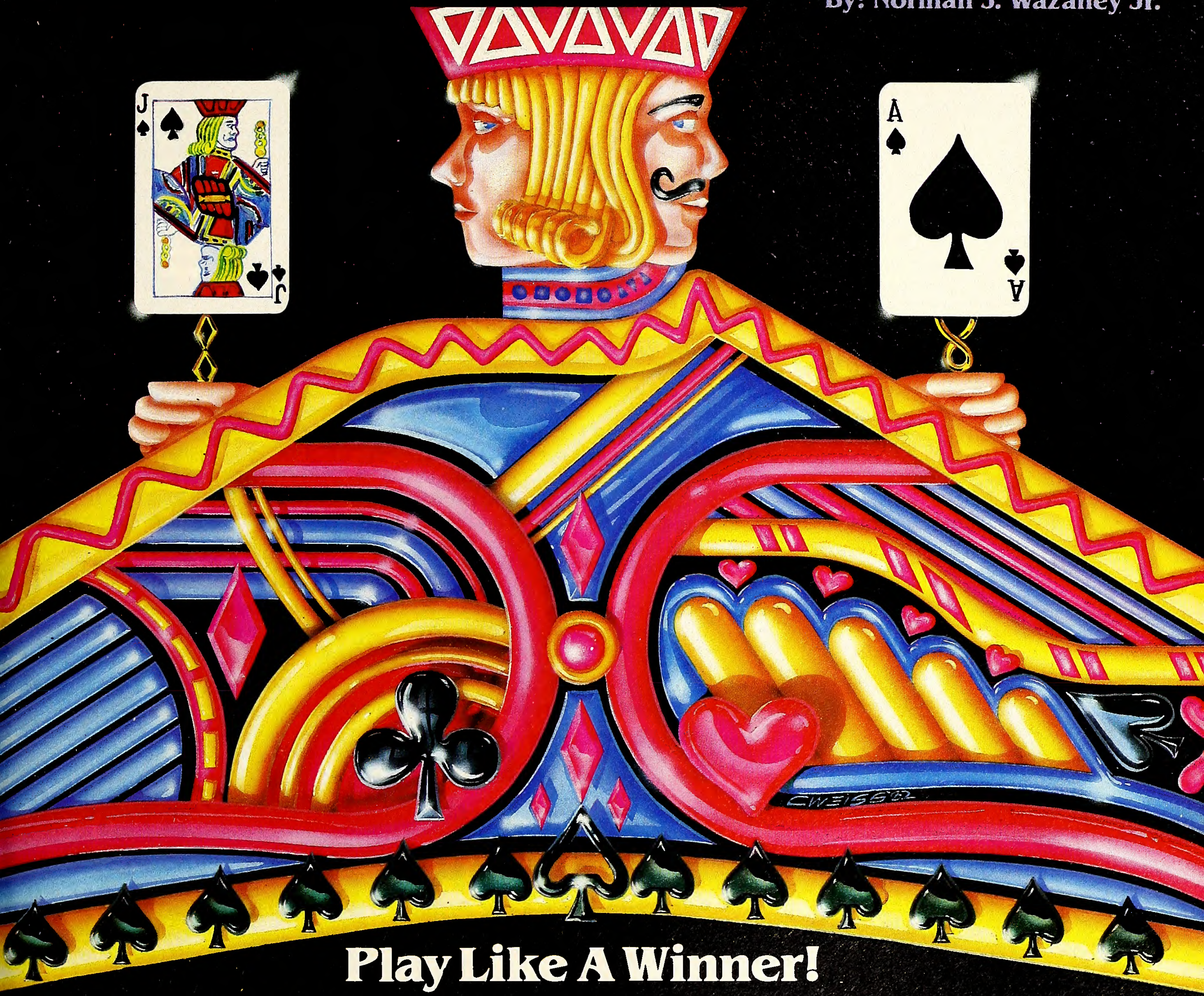
Since these addresses are in the Monitor ROM and cannot be changed, and since at the time not all future requirements were known, Apple Computer implemented the vectors in the most flexible way possible. This meant setting up a separate set of vectors in low memory that could be changed and arranging the high memory vectors so that the 6502 would wind up at these low memory vectors eventually. In this way, no matter what was created in the future—DOS, CP/M, Pascal, and so on—a means was left open to allow the 6502 to wind up at the proper point to recover the system, as when the reset key is pressed in DOS and the system reinitializes itself, leaving DOS active and in control.

**Low Commotion.** The vectors just moved by the finder code are the low memory vectors for CP/M, and they point to the subroutine caller loop in the IOCB. In this way, whenever a reset is encountered, for example, the Z-80 will have been turned off and the 6502 will have been enabled and directed to the reset vector in high memory. After executing some Monitor code, the 6502 is eventually directed to the reset vector in low memory and through this to the Z-80 reactivation side of the subroutine caller loop. Executing this loop, the 6502 restarts the Z-80, which immediately branches to 0000H and executes the code found there. If CP/M has already been completely booted, this code will be a JumP to the Z-80 warm boot code to reinitialize and recover the system. Since we have not yet fully booted CP/M and 0000H still contains the Z-80 recovery program stored there by the cold loader, the finder code now places a Z-80 JumP instruction and the address of the Z-80 cold boot code at $1000 (0000H) for later execution.

**Call the Whole Gang In.** Obviously, the next step is to get the CCP, BDOS, and the rest of the BIOS into memory, especially since the BIOS contains the Z-80 cold boot code! This is done by the finder code calling

# BLACK JACK STRATEGY II

By: Norman J. Wazaney Jr.



## Play Like A Winner!

This complete system provides simple, step-by-step instructions to help you raise your level of play regardless of your present skill. Features include a SIMULATOR for testing betting and playing strategies, a TUTOR to teach the strategies and a GAME with all the options available at the casinos. Another outstanding feature is a Strategy Table Compiler for ease of entering and visualizing your playing and betting strategies.

The system is able to simulate millions of hands and provides better insight into computing odds and house percentages. You may be surprised to discover how many of your pet theories are less than accurate.

Black Jack Strategy runs on 48K Apple II, Applesoft in ROM.
3.3 DOS/One Disk Drive Printer Optional

OUR SOFTWARE IS UNPROTECTED PERMITTING CONVENIENT BACK-UP.

The comprehensive documentation is packaged in a handsome, easel-backed binder. Available now for only $69.95.

Since winning is always more fun than losing, be fair to yourself and get Black Jack Strategy before your next visit to the casino (where $69.95 will not go very far).

To order by phone, call 800-526-9042 and use your Visa or MasterCard. All shipments made the same day in which orders are received. To order by mail, add $1 for shipping charges and send your check to:

**Soft images**
DIVISION OF DECISION SYSTEMS INC.
200 Route 17, Mahwah, N.J. 07430.

the 6502 warm loader subroutine at $0FAD. This loader is the last module we'll examine this month. But first we'll complete the finder code.

For now, assume that the warm loader has executed and that all of CP/M has been loaded from disk. The finder now patches the warm loader code itself, so that in future warm boots the warm loader will load only the CCP and BDOS (twenty-two sectors) and not the high memory BIOS portion (an additional six sectors). After performing this small step, the finder places the same restart vector address (the Z-80 side of the subroutine caller loop) into the last six addresses of the RAM board if one is installed, and these become the new high memory restart vectors.

At this point, then, a system restart or interrupt will be handled properly no matter how it occurs. The finder code is now complete, and it will branch to a point in the caller loop that properly shuts down the 6502 (saving registers and status) and starts the Z-80. Next month we'll pick up the Z-80 boot code from this point, but for now we'll go on to cover the last 6502 module.

The warm loader is similar in some ways to the boot sector, since its primary function is to read a set number of sectors from the system tracks into memory. Using RWTS, however, involves dealing with a table called the I/O block or IOB, not to be confused with SoftCard's IOCB area. The IOB is a concept that Apple has used with RWTS since the first thirteen-sector DOS, and it is the way that RWTS keeps track of all needed information about the current disk access it is performing.

**With a Twist.** SoftCard's IOB is structured differently than that of standard DOS, and SoftCard's version of RWTS has been altered to handle those differences. Made up of twelve bytes beginning at $03E0 in the IOCB, this table has entries for the following items in the order shown:

1. current track
2. current sector
3. current volume (unused)
4. last volume (unused)
5. current drive (1 or 2)
6. last drive
7. current slot (times 16)
8. last slot (times 16)
9. buffer address (two bytes)
10. error code
11. command (read, write, and so on)

If the proper values are placed in the proper entries, RWTS can be told which track and sector to read or write to and where in memory to get it from or write it to. RWTS will also place an error byte in entry 10 that signals the calling routine that there was an error and, if it's identifiable, what the error was. Command codes are 00 for seek track and sector, 01 for read, 02 for write, and 04 for format.

Using RWTS and the table, the warm loader begins by placing $E400 into the buffer address entry for 56K CP/M ($B400 for 44K) and track 0, sector 11, slot $60 into their appropriate entries. Then the warm loader calls RWTS at address $0E03. Decrementing and keeping track of the number of sectors remaining to load, the warm loader increments track, sector, and buffer address in the IOB table, repetitively calling RWTS until the entire system has been loaded. Once done, the loader places $0800 into the buffer address entry, since this will be the standard disk buffer in all future reads and writes. It then returns to the finder code as we saw above and is patched by the finder code to load only twenty-two sectors, CCP and BDOS only, when it is called in the future by the Z-80 warm boot code. In future calls, the warm loader is accessed via a special vector at $0E00 that contains a 6502 JumP instruction with the address $0FAD.

This completes our in-depth examination of the 6502 boot routines. Although direct use of this information is limited, it does provide a much clearer picture of the means by which the system becomes operational. In addition, certain areas or routines we have discussed, such as RWTS and the IOB table, can be accessed by your own programs. Until next month. . . .


SINGLES' NIGHT AT MOLLY'S
By Norman J. Wazaney Jr.
MOLLY'S
OUR SOFTWARE IS UNPROTECTED PERMITTING CONVENIENT BACK-UP.

# What a Turkey!

# An Apple Keeps Track of the Flock

## BY ANDREW CHRISTIE

*The holiday meal about to weigh down your groaning board did not while away a leisurely existence ambling about a barnyard pecking at corn feed and chasing the farmer's dog. There is no longer such a thing as a turkey in the straw; that material long ago proved insufficiently absorbent as litter and is no longer used. Times change, and our hallowed traditions now depend on automation—and in some cases, a powerful little computer—to keep running smoothly.*

Every four weeks, a tractor trailer from the Land O' Lakes hatchery in Foley, Minnesota, pulls up to Dick Christiansen's turkey farm on Route One in Newell, Iowa, with a new flock of five thousand toms. Dick's son Mike drives out to the Newell Co-Op Elevator and delivers the feed order to mill man Don Piercy. While Piercy grinds the feed, the tiny toms are quickly deposited in the brood pens that will serve as giant incubators until they are moved into one of seven containment buildings—their home for the next twenty weeks and the rest of their lives.

*Call him Sam. He's a white tom. He was separated out from the hens at the hatchery, desnooded to prevent infection, debeaked to discourage him from picking at or devouring his fellows during moments of stress, and the inside two toes on each foot have been clipped to reduce the possibility of injury, preserving him in the optimal grade.*

*Sam arrives at the farm and is put into a brooder ring with four hundred of his peers. The floor of the ring is covered with three inches of wood shavings. The brooder hover stove, a circular metal affair resembling the saucer from* The Day the Earth Stood Still, *hangs from the ceiling on a safety chain and radiates a comforting ninety-five degrees of welcoming heat to the apprehensive little poults. Five long feeder troughs extend out from the center of the ring, alternating with water jugs. Things look okay. Sam relaxes. The poults spread out across the ring, each giving the other his own space. Sam is content. Everyone commences to peep.*

The Christiansen family farm, like most American farms, is more than a hundred years old and is a hereditary business. It went into the turkey management business in 1954, but Dick Christiansen still raises corn and soybeans, products nicely complementary to the farm's main product.

The first Apple arrived in 1978, when it was perceived that there might be a certain fiscal advantage to knowing the exact costs involved in putting a flock of five thousand turkeys on the market every month. Suddenly the Iowa Office Supplies Center became vital to the Christiansen farm. With no computer experience, Dick Christiansen bought an Apple system and hired commercial programmer Kim Catania—"He didn't know anything about turkeys and I didn't know anything about computers"—to write the farm's first program, *Turkey Production Records Systems*. Christiansen was then able to see a weekly summary of feed to each flock, keying in average sample weights for cost of production per pound.

*There was tension in Sam's ring. The water jug and feeder troughs had been gradually removed until there were none left. Many of the young toms were starting to panic, piling up under the brooder stove and attempting the forbidden feather picking. Other toms felt that the automatic fountains that had been noticed in the immediate vicinity of the last water jugs, as well as the automatic feed line inside the periphery of the ring, could sustain life. Many of the toms didn't believe it; they resisted changing from the old ways. Life had seemed so much simpler in the old days the week before. Some pined away and died as starveouts, but the majority of the flock rose to meet the challenge of the new technology and thrived.*

# PLAY THE SYSTEM THAT MADE KEN USTON THE WORLD'S WINNINGEST BLACKJACK PLAYER.

**Ken Uston,** familiar to millions as the blackjack expert featured on CBS's "60 Minutes," has won a reputation as the world's foremost blackjack player. Now Intelligent Statements brings Ken Uston's casino-proven blackjack system to your home computer.

A$

A$

## PLAYS LIKE THE REAL THING

**Ken Uston's Professional Blackjack**™ is the most complete and realistic blackjack game you can buy. You'll meet the same playing opportunities that you'd face at a real blackjack table — at your choice of over 70 Nevada and Atlantic City casinos, each with its own set of rules and variations. Or you can create your *own* casino, manipulating sixteen different game parameters to produce *39,813,120* different playing environments. You can select the number of decks in the shoe, vary the dealing speed, choose and name competing players, set their wallet sizes and much more. And all your data is accurately displayed, so you can play the strategy you like and get the feedback you need **to win.**

J$

Ken Uston's PROFE$$IONAL BLACKJACK™

J$

## WINS LIKE THE REAL THING

**Ken Uston's Professional Blackjack** is the most thorough and authoritative teaching system you can buy. Now you can learn all of Ken Uston's computer-optimized card-counting strategies, from basic to advanced levels. Menu-driven interactive practice drills — augmented by superb documentation — lead you through each skill level. At any point you can choose to see accurate running counts, continuous statistical evaluations, discard deck totals and instructional prompts, complete with sound effects. So you develop and refine the skills you need **to win big.**

## DON'T LOSE OUT. START WINNING NOW.

Start winning at blackjack. Send $89.95 (plus $2.00 for shipping and handling) to:

**Ken Uston's**
**Professional Blackjack**
DEPT. ST 11
Intelligent Statements, Inc.
Box 600, Holmes, PA 19043.

(North Carolina residents add 4% sales tax). Make checks payable to Intelligent Statements, Inc. Major credit cards accepted over 24-hour toll-free lines at **1-800-345-8112** (PA residents call 1-800-662-2444). Be sure to specify your home computer brand.

**IBM PC* requirements:**
48K RAM, disk drive, PC DOS,* 80-character display. Color and monochrome versions supplied with each package.

**APPLE II** requirements:**
(DOS 3.3)
48K RAM, disk drive, 40-character display.

Display shown is actual photograph of IBM PC color version. Other versions vary with hardware limitations.

Versions for CP/M**, Atari** and TRS-80** should be available before Christmas.

intelligent statements™

GROWN-UP GAMEWARE™

*It was the light they loved, and the light loved them. By the magic of its endless candlepower, the electric days could last up to eighteen hours. "Night" itself did not plunge them into darkness but was symbolized by a dim light that allowed them to see and kept them calm. Their growth accelerated, their feathers became thick and white, and they began to exhibit the full rotundity of form that is the traditional mark of success in life.*

---

In poultry management, feed represents 60 percent of total production cost. Dick Christiansen uses a complete turkey feed. The Land O' Lakes feed programs on which Christiansen and most other turkey farmers depend permit selection of ingredients in the manufacture of the feed, allowing consideration of ingredient costs plus the costs of grinding, mixing, and hauling the feed. A complete feed program delivers feed with ingredients mixed to the farmer's specifications; a base feed program gives the farmer a formula of protein/fat starter, to which he adds his own soybean meal or corn, manufacturing the feed himself. Complete feed is more what you might call an assembled feed; basic, of course, is slower.

**How Do You Soar with Eagles . . .** No successful turkey grower works in isolation. Merlin Sletton is the Christiansen's turkey production specialist. Dealer support in the world of poultry management is just as important as it is in some other fields. He monitors the feed ingredients and flock condition and prescribes necessary medications.

At Iowa State University, extension poultry specialist Bill Owens labors to discover more economical feed formulations and alternative ingredients, working from the basic equation of one pound of meat produced for every three pounds of feed. His job is to find ways to bring one end of that equation down, or the other end up, or both. He must work within Food and Drug Administration established guidelines regarding permissible drug use and combinations with species and age of bird. Added fat comes in handy for increasing the energy concentration of the ration, improving palatability, reducing dust and feed wastage, and improving the "finish" of birds.

**. . . When You're Working with Turkeys?** Dick Christiansen is one of twenty-six Iowa turkey growers on the board of the Iowa Turkey Federation, a commodity organization for the state turkey industry that organizes educational meetings and research and keeps consumers informed with new recipes and articles on the correct preparation of turkeys. Through the National Turkey Federation in Reston, Virginia, it keeps Iowans informed of legislative developments. According to ITF executive secretary Carolyn J. Taylor, "Every time a processing plant opens its doors, it has to meet a whole set of FDA rules and regulations, and we have to keep our people informed of pending legislation. The government can even tell you how much water you can use at the plant. (In processing, each turkey must undergo a "chill cycle" in which it is submerged in chilled water for two to three hours.)

---

*It was time to leave the brooder ring for the real world of a confinement growing unit. At eight weeks, Sam was entitled to two feet of floor space, but eventually he would grow to command a full four square feet. He considered himself lucky. His cousin Dan had been a range turkey. One day it had rained. Before the workers could get Dan to a range shelter, he had looked up to see what rain was and drowned. Then there was little Mary Jo, a white range hen who had met up with a migrating water fowl and contracted Avian Influenza.*

*Not that life in the confinement unit was all corn and soybeans. Some of the toms had grown too fast and succumbed to aortic rupture. The turkey serviceman came and ordered a pound of copper sulfate per ton of feed. At a farm across the way, respiratory disease had claimed a dozen, and the litters, walls, and ceiling were sprayed with calcium proprionate mold inhibitor and Poul-Dine was put in the water. A minor Bluecomb outbreak made some of them depressed and morbid. Others were felled by Round Heart or Water Bellies. Life went on.*

---

**Turkey Tribulations.** Through the Veterinary Medical Research Institute, the Iowa Turkey Federation has funded $130,000 in turkey disease research since 1972. Carolyn Taylor is philosophical about the matter.

# Not all business
# And we've got the

As you know, one picture is worth a few thousand numbers.

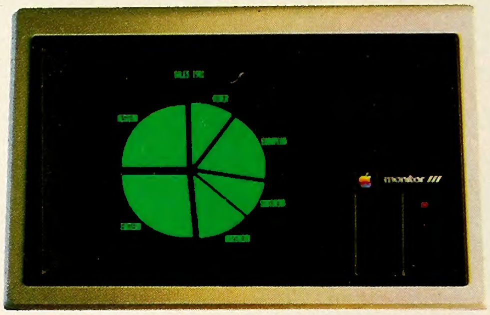
As you may not know, Apple® Business Graphics software can generate more types of pictures, in more colors, using more data than any other graphics package.

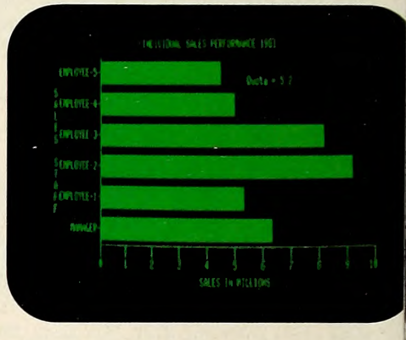
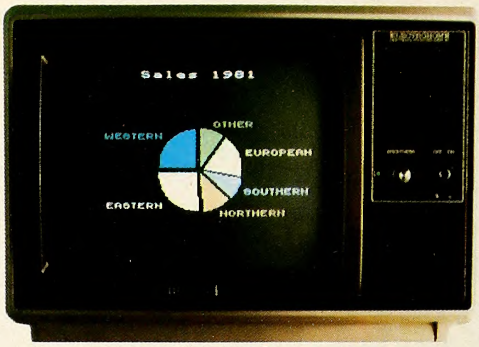
So you not only get the usual bar graphs and pie charts. You also get unusual bar graphs and pie charts. Sophisticated line and area graphs. Even scattergrams. All teamed with extremely useful and powerful features—exploded views, unlimited overlays, floating titles and more.

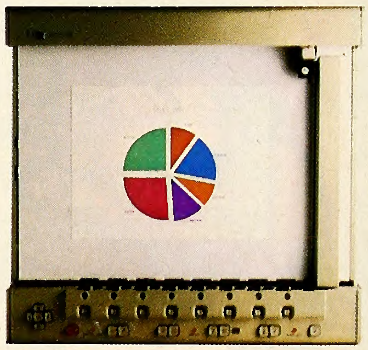
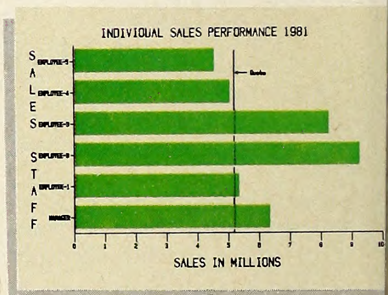
| | Apple | VisiTrend/ VisiPlot | pfsGraph |
|---|---|---|---|
| *Graph Types* | | | |
| Line | Yes | Yes | Yes |
| Vertical Bar | Yes | Yes | Yes |
| Horizontal Bar | Yes | No | No |
| Side-by-side Bar | Up to 4 | 2 | 4 |
| Pie | Yes | Yes | Yes |
| Partial Pie | Yes | No | No |
| Scattergram | Yes | Yes | No |
| Curve Fitting | 5 Kinds | 1 | None |
| *Data Points (Max.)* | 3500+ | 645 | 36 |
| *Plotter Compatible* | Virtually Any | None | H-P7470A Only |
| *Compatible File Types* | Pascal BASIC VisiCalc | BASIC VisiCalc | pfs VisiCalc |
| *Math Functions* | Yes | Yes | No |
| *Available Colors* | 6 | 4 | 4 |

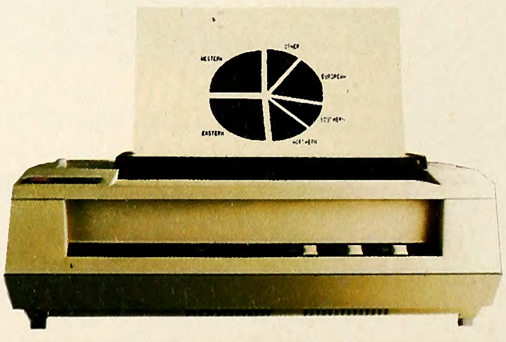
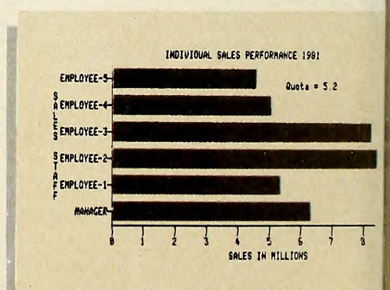
*Apple Business Graphics is available for both the Apple II and Apple III.*

Equally important, with our graphics package you'll find more ways to see what you're doing. On the monitor of your choice. And on virtually any printer or plotter on the market.

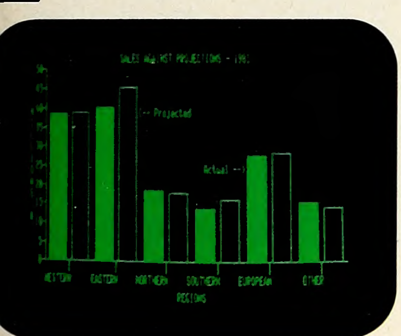


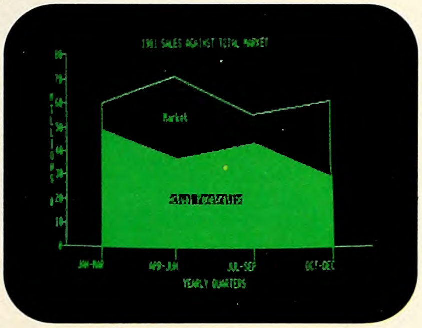



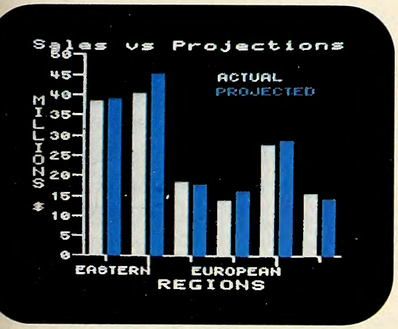



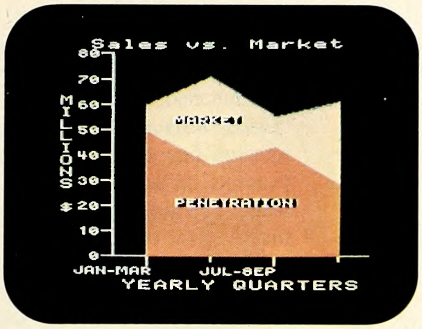



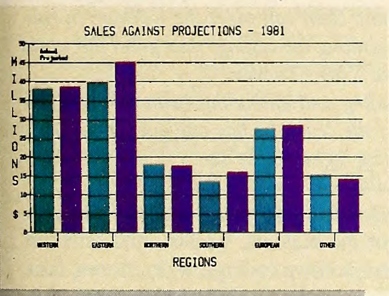
# graphics are alike. pictures to prove it.

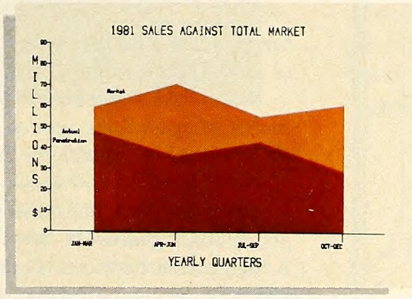


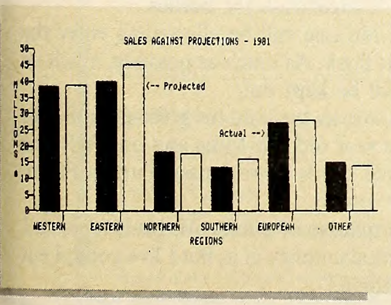



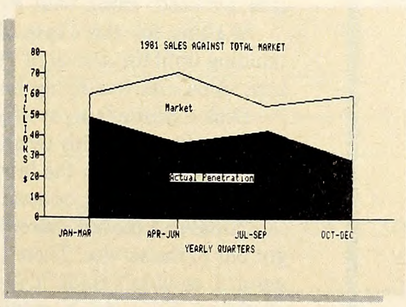



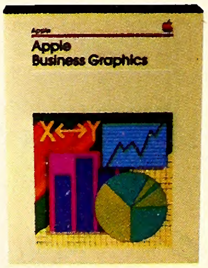

Even on transparencies and slides (by combining Apple Business Graphics with packages like Screen Director™ and Target Image Maker™).

All of which makes for more presentable presentations. And more revealing market analyses, forecasts, budgets, stock trends, business plans or customer demographics.

Or the information of your choice from the files of your choice. Be it VisiCalc®, Pascal, DIF or BASIC.

We could easily tell you more.

But we'd rather show you more. In person. At any of our over 1300 full-support dealers (they also offer a vast library of other quality software distributed by Apple for Apples).




So pay one a visit. And find out how easy it is to turn a sea of data into data you can see.

apple®

The most personal software.

Call **(800) 538-9696** for the location of the authorized Apple dealer nearest you, or for information regarding corporate purchases through our National Account Program. In California **(800) 662-9238.** Or write Apple Computer Inc., Advertising and Promotion Dept., 20525 Mariani Ave., Cupertino, CA 95014.

Screen Director is a trademark of Business Professional Software, Inc. Target Image Maker is a trademark of Comshare Target Software. VisiCalc is a registered trademark of VisiCorp.

"Turkeys are always plagued with some kind of disease problem. Right now the institute is working on Newcastle Disease and Avian Influenza. A while back we had a lot of trouble with Coryza. We're getting so we can pinpoint them, but you get one thing licked and something else always crops up. What it amounts to is that the turkeys get themselves sick. If someone walks into the room, if they get excited, if a germ gets into one of these disease-free buildings. . . . You've got to remember that these birds are bred for meat, not brains. They're susceptible."

The second Christiansen farm program was a farm accounting system. Now there were profit-and-loss statements at the end of the month and for the year to date, and at year's end, a complete cash flow printout for all farm operations, business and personal. A balance sheet presented completed assets against liabilities and equity, causing much approval from the vicinity of Citizens First National, bankers enjoying such things as they do.

As it was for so many others, the advent of *VisiCalc* was manna from heaven for the Christiansen Apple. Plugging in an additional twenty-four cents per bushel of corn and subtracting a dollar ten per ton of soybean meal, then projecting a change in the selling price per pound of turkey in a what-if program involving thirty-two feed items, did indeed seem like magic. Every two weeks, the Land O' Lakes feeding program, with the cost of all ingredients, gets plugged into *VisiCalc* for the least cost ration to feed proportions. The Christiansens now have about three years of data and are able to plot it with a data plotting system, creating separate graphs with overlays for cost per bushel of corn, turkey prices (a record low of \$2.87 in June 1980; a record high of \$4.95 by October), and monthly rainfall with a ten year average. The payoff is efficiency.

"Because the economy is what it is," says Dick Christiansen, "our profit picture looks poor, but it would look a lot poorer without the Apple." There was a bumper crop of turkeys last year, depressing the market for eighteen months. Breeders accordingly cut back, producers are producing less, and now turkeys are in short supply; in higher demand, at higher prices.

The turkey's higher visibility is certainly due in large part to the tireless efforts of the National Turkey Federation, who, while not having the economic clout of their organizational bedfellows who promote the edibility of eggs or the sophistication bestowed on the consumer by the drinking of milk, try harder. It is through their efforts that turkey (low calorie, high protein, low cholesterol) has become an accepted ingredient in the nation's frankfurters, as well as establishing itself in the deli section as a luncheon meat in its own right. Early this year, during the critical "turkey fatigue" period following the holiday season, the NTF offered retailers free radio advertising time if they would use it exclusively to promote turkey products sold by their stores. The effort paid off, to the relief of growers and processors, offsetting the potentially grim effects of the seasonal turkey depression combined with the production surplus.

Dick Christiansen (left) goes over farm reports with turkey production specialist Merlin Sletton.

---

*On Sam's kill date he is herded into the turkey loader when it pulls up to the building and pokes its nether end through the door. He finds himself briefly on a conveyor belt, and is plucked out at the top of the loader and installed in a crate with several other turkeys, then deposited in the Dil-Mar Foods truck below with many other crates. The flock makes its way to the processing plant at Storm Lake, where Sam will meet the workers in rubber aprons and the long hooks moving along the ceiling, then greet the world at large, shrink-wrapped and labeled "Mr. Turkey."*

*But, in the meantime, in the loading and unloading, with his head immobilized against one wooden slat of the crate, he has one eye turned up to the sky, and he regards the surprising light of the sun.*

---

After the flock has gone, all movable equipment—feeders, waterers, and brooder guards—is taken out of the building, washed, and sanitized. A high-pressure sprayer is used to wash down ceilings, walls, floors, fans, and any remaining equipment. The building is allowed to dry, then building, floor, and equipment are sprayed with disinfectant. Next, they are sprayed with Environ, a phenolic germicidal detergent, to take care of pathogenic bacteria, virus, and fungi. Finally, poultry spray and larvacide are used, along with Ronnel Concentrate for beetles.

**Waiting for the Class of '83.** No one will be allowed to enter the building until the arrival of the next flock. As much as possible, it will be kept open and free-flying birds will be kept out.

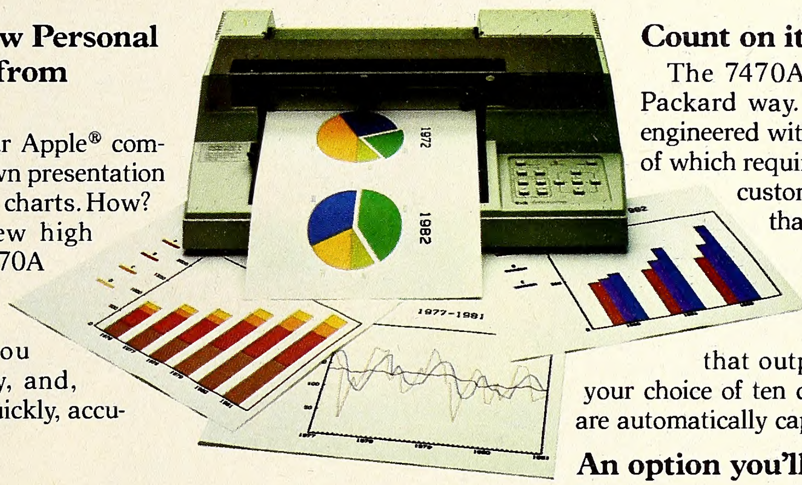
Dick Christiansen, at fifty-one, works the land his great-grandfather worked, along with his twenty-five year old son Mike and one full-time man. They make up the labor force of the Christiansen Turkey Farm. Dick Christiansen is content.

"I haven't moved more than a quarter of a mile off this land since I got out of the service. There's not much money in it, but I live good; I eat good. I wouldn't trade it for anything."

Some things don't change.

# One Apple® and $1,550 can make a lot of pies.

## And charts. And graphs.



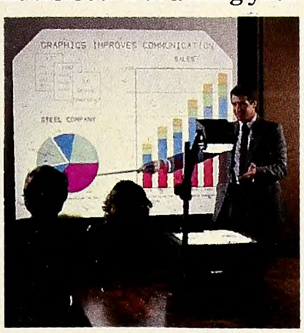

### Introducing the New Personal Computer Plotter from Hewlett-Packard

Now you can use your Apple® computer to generate your own presentation charts, graphs, and pie charts. How? Simply add on the new high quality, low cost HP 7470A Personal Computer Plotter.

The 7470A helps you save time, save money, and, lets you communicate quickly, accurately and effectively.

### The eye is faster.

Data, when visualized graphically, becomes information fast. Charts and bar graphs can make any presentation clearer and more readily understood. But asking your staff to produce the graphics for your next presentation doesn't ensure accuracy or artistic talent. And going to outside suppliers can be costly. Combined with your Apple® computer, the new HP 7470A plotter does the communicating for you. Quickly. Logically. And with off-the-shelf software.




### Fast and pretty.

The 7470A gives you high plotting speed with excellent line quality... faster than any competitive small plotter. On top of all that, it comes in an attractive design package that looks nice on your desk. And it does it for only $1,550. (U.S.A. domestic suggested retail price.)

### Count on it.

The 7470A is built the Hewlett-Packard way. To last. Designed and engineered with only a few parts, none of which require adjustment. And with customized integrated circuits that ensure reliability.

### Pen Pals.

The HP 7470A has two single-pen stables that output multi-color plots in your choice of ten coordinated colors. Pens are automatically capped and stored.

### An option you'll want, too.

For only $95, you can also get a 17057 Overhead Transparency Kit that turns your plots into transparencies for overhead projectors. For "I need it tomorrow at 9:00 A.M!" meetings, it's a necessity.

### Start plotting your next presentation today. Clip and mail the coupon below. Now.

Mail the coupon below and we'll send you—absolutely free—a sample plot, a more detailed brochure, and a sample overhead transparency.

Then... stop in at your nearest Hewlett-Packard Dealer. See the HP 7470A in action. Once you see it demonstrated you'll find a hundred ways to make your own Apple® pies. And charts. And graphs.

*When performance must be measured by results*

HEWLETT PACKARD

Seeing is believing. Send me a sample plot, an overhead transparency, and more detailed information.

Name __________ Title __________

Company __________

Address __________

City, State & Zip __________

Phone Number (      ) __________

Send to: Hewlett-Packard, 16399 W. Bernardo Drive, San Diego, CA 92127—Attn: Nancy Carter



11203 ST/X

# ]RUN THANKSGIVING 0.1982

## BY ANDREW CHRISTIE

As the fourth Thursday in November bears down upon us once again, it's time to take a long, critical look at the infrastructure of our traditional holiday system. We must ask: can Thanksgiving make it in the information age? Though there have been modifications over the years in food preparation hardware and technique, these are essentially minor changes; its basic holiday technology has remained unchanged for 361 years.

More than any other holiday, the American Thanksgiving Day is an intriguing amalgam of pagan celebration, rendering homage to the spirits of successful agricultural endeavor, and the European Pietistic mode. Thanksgiving's design structure is dissimilar to that of the standard modern holiday, bypassing, as it does, the grid overlay of church tradition on a locus of preexistent native ceremonial data—the favored method of fifteenth and sixteenth-century Catholic field engineers faced with the problem of integrating Christian programming with that of a possibly incompatible host culture, sometimes requiring brute force technique. It is, rather, uniquely indigenous. But it is not a new invention. Several thousand man-years of basic myth development went into Thanksgiving's current streamlined civil/religious holiday configuration.

The first known version to have been run successfully was designed by Israeli agrarian engineers in an Exodus emulation mode. Designated the Feast of the Tabernacles (SUCCOTH I), its elegantly simple four-function design provided for harvest-thanks, folk dance addressing, forebear history recall, and full display of the fruits of harvest, serving as a model for many future systems. The documentation, in the book of Nehemiah, remains a marvel of user-friendly clarity:

> Go forth unto the mount, and fetch olive branches, and branches of wild olive, and myrtle branches, and palm branches, and branches of thick trees, to make booths. . . . So the people went forth and brought them, and made themselves booths, every one upon the roof of his house, and in their courts, and in the courts of the house of God, and in the broad place of the water gate . . . and there was very great gladness.

A major later development was the Greek Thesmophoria, or feast of Demeter, goddess of the harvest. This was configured for Athenian married women in the month of November, an important early retrofitting of agricultural harvest effect with human fertility. Two wealthy women were chosen to supervise the sacred meal task management. On day one of the feast, while the general populace executed the mirth/rejoicing routine, the women proceeded sequentially to the promontory of Colias and initiated a seventy-two hour Thanksgiving celebration function in the temple of Demeter. Their return poked a public celebration for the same time period, beginning in solemnity before branching to an orgy sequence and livestock execution. The input values of a fruitful goddess included poppy arrays and ears of corn, a basket of fruit, and a little pig.

The Roman Cerelia, or feast of Ceres, added little to the borrowed Greek Thanksgiving technology, essentially moving back the time frame to October 4 and introducing a fasting subroutine among the common people, who offered Ceres a sow and initial harvest cuttings. Most significant to the evolution of the present-day version, sports and field procession event scheduling were added to the feast routine, prefiguring the modern NCAA/Macy's parade sequence.

**The Pilgrim Process.** By the seventeenth century, the basic harvest/fertility/thanks components were in place, requiring only consolidation and modulation to be formed into the final Thanksgiving procedure. At this time, the Puritans, a high-mobility English group, during a ten-year stay in Leyden, Holland, had occasion to effect a study of the Netherlands' annual thanksgiving celebration of delivery from Spanish rule. The Puritans did not care for the sloppy holiday programming technique that had jammed the English event calendar with feast days, but they were attracted to the efficient holiday pathways of the Dutch, which also demonstrated the viability of religious linking loader implementation.

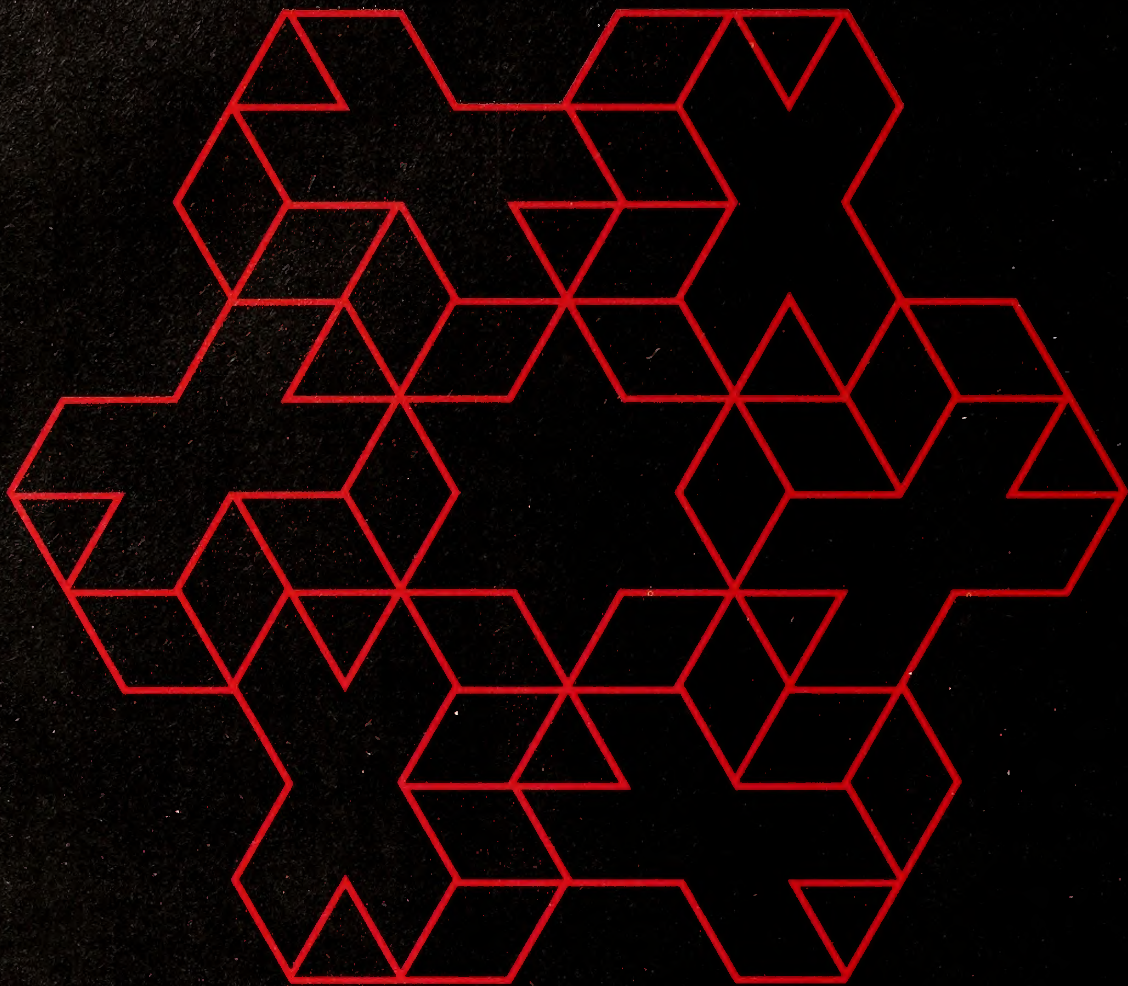
Eventually establishing a base in America, the Pilgrims made an initial New World harvest test in October 1621. It provided enough raw material for team leader William Bradford to command a three-day feast and celebration based on the English harvest home model. Though this first New World holiday model was deemed a success, it proved difficult to repeat, research in subsequent years yielding frequent disappointments marked by recurrent crop failures. Droughts and inadequate harvest, however, increased the Thanksgiving celebratory factor in an inverse ratio whenever a measure of agrarian success was achieved, and so the various Massachusetts groups celebrated their harvests at random for the next fifty years.

**Federal Aid.** Though this time is considered the Thanksgiving golden age by some, with participants freely sharing festivity techniques and thankfulness theory, it is clear that there could have been no advances in the field as long as the holiday base remained a loose, decentralized aggregate of local independents.

In this regard, the 1789 Thanksgiving Proclamation of George Washington must be considered a landmark in Thanksgiving development; the first proposal for a standard format, universal celebrating system:

> "Now therefore, I do recommend and assign Thursday, the twenty-sixth day of November next, to be devoted by the people of these States to the service of that great and glorious Being, who is the Beneficent Author of all the good that was, that is, or that will be; that we may then all unite in rendering unto Him our sincere and humble thanks for His kind care and protection of the people of this country, previous to their becoming a nation; for the signal and manifold mercies, and the favorable interpositions of His providence, in the course and conclusion of the late war; for the great degree of tranquility, union and plenty, which we have since enjoyed; for the peaceable and rational manner in which we have been enabled to establish Constitutions of Government for our safety and happiness, and particularly the national one now lately instituted; for the civil and religious liberty with which we are blessed, and the means we have of acquiring and diffusing useful knowledge; and, in general, for all the great and various favors, which He has been pleased to confer upon us."

# Accounting Plus II™



## Plus...

Software Dimensions announces three new modules for Apple II* users with Accounting Plus II—each designed with the same high standards and human engineering factors that have made Accounting Plus II so popular.

With **Invoices Plus** you can inventory items just by entering part numbers; easily invoice non-standard items and miscellaneous charges; place user-defined comment lines on the invoice; get twice as many line items per invoice; automatically generate multiple pages; specify individual invoice numbers or assign them automatically—and more.

**G L Plus** expands your financial reporting capabilities, including changes in your financial position, offering even more flexibility for your General Ledger module. Typical reporting features include Budget/Prior Months' Listing, Income Statements, Balance Sheets, Budget Variance Report, Comparative Reports, Source and Use of Funds.

**Labels Plus** offers a versatile label package that lets you maintain one or more mailing lists with up to 1,000 names each, as well as to access existing Vendor and Customer files easily; add, edit and delete names; sort and select; print mailing labels.

You won't find this kind of flexibility and sophistication in any other Apple II accounting system. To find out more, contact your Accounting Plus II dealer.

Software Dimensions Inc., 6371 Auburn Blvd., Citrus Heights, CA 95610. 916/722/8000

*Registered trademark of Apple Computer Inc. Accounting Plus is a trademark of Software Dimensions Inc.

That just about covered it. (Note the founding father's typical great foresight in giving thanks for "the means we have of acquiring and diffusing knowledge.") But the ex-colonies proved to be backsliders, and Thanksgiving soon lost what ground it had gained in the market to the major organized forces of Christmas and Easter, falling back to its previous undated regional level, where it stayed for many years.

In the end, it was only the tireless efforts of Thanksgiving booster Sara Josepha Hale, editor of *Godey's Lady's Book*, that succeeded in marshaling available technology to produce the Thanksgiving product in the form we know it today. Through her editorializing, and by her personally prevailing upon President Lincoln, a reproclamation was issued declaring the fourth Thursday in November as the official date of observation. Each president since then has performed the dynamic memory refresh; all state governors have received the protocol and relayed it, naming the same day, completing the subroutine and improving program modularity.

Thus did Thanksgiving Day become the only religious festival operating system celebrated in the United States by authority of the civil government. A program interrupt occurred in 1939 when FDR tried to retrofit it to the third Thursday in November, causing widespread chaos. Thanksgiving crashes occurred regularly, and all the states slipped back into their random habits again. It took a joint resolution of Congress in 1941 to perform the final debugging process: "That the fourth Thursday in November in each year after the year 1941 be known as Thanksgiving Day, and it is hereby made a legal public holiday to all intents and purposes."

**Overseas Competition.** We can reasonably conclude that our Thanksgiving system is now basically sound and performs its designated functions with maximum efficiency and minimal required energy output. As the old folk proverb goes, "If it works, don't fix it." But how does our fundamental system stack up on the international market? (See figure 1.)

In Canada, Thanksgiving is the second Monday in October, incrementing from the harvest-home ceremony bus of the Sunday preceding. Outside North America, Thanksgiving formats are somewhat less . . . sedate.

In Peru, the harvest is in May and the celebration is the Invention of the Cross, the Christian program update for the Incan Ayllihuay, a nocturnal bash for the life-giving forces input to the soil of the earth from the night skies. The cross was everywhere in the early sixteenth century, replacing the native home gods as preferred hearth-protection mnemonics. In the mountains, on the eve of Cruzvelacuy, bonfires are burned-in around all Christian processing units, which are enhanced with votive candles, lights, ornaments, and flowers.

Christianity hit Finland in 1200 A.D. and the unpronounceable Midsummer's Eve celebration and fertility feast, in which the fishermen gave thanks to the rivers and lakes for their bulk storage and retrieval capacity, was moved to the Saturday closest to the June 24 feast day of Saint John the Baptist. Dwelling units decorated with cut branches effect symbolic new-life simulation. The Midsummer's Eve *kokko* (bonfire) requires a long and tedious wood-gathering process, as it must burn at least until midnight along all the lake shores and the sea coast. Theoretically, if a girl bathes her face in the dew concurrent with the midsummer sun power-up sequence, no additional supplementary beauty enhancement will be required. As she is homeward bound, if she retrieves flower arrays of maximum fragrance potential and tucks them under her pillow, she will dream of the man she is to marry. This innovative design apparently accounts for the high correlation of marriages performed on Midsummer's Day to the number of engagements contracted around the bonfire the evening before.

The rest of Europe is generally based on standard Ingathering Feast techniques and need not be elaborated on here.

Liberian Thanksgiving is configured on technology exported by a boatload of American colonial engineers who interfaced with the African coast in November 1820 and held on-board Thanksgiving services in recognition of same. Two years later, colonists had the ceremony up and running as a national Liberian holiday after successful battles against the Dey, Mamba, and Vai tribes. At church services, funds are prompted to flow to areas of lesser concentration, followed by a jump command to feasting/entertainment with costume/exhibit array enhancement.

On Taiwan, the Chinese celebrate the Festival of the Harvest Moon on the fifteenth day of the eighth moon of the lunar year, usually late August to early September. On the night of the Moon Festival, Yueh Lao, the old man in the moon, links designated couples together with invisible red silk thread. (How do they know it's red? Hush, child.) Moon cakes are the high-order element of the festival. These were field-tested during the Yuan Dynasty (1368 A.D.) when the Chinese were attempting a program for the deletion of the alien Mongol rulers. A senior bakery analyst hit upon the idea of concealing *delete* message units in round-cake dedicated registers. The cakes were exchanged as gifts, and commands were successfully entered without authorities breaking the code protection. The revolution was successful, and the round, flat cakes filled with sweet, starchy filling became referred to in the jargon as moon cakes.

Meanwhile, across the broad Pacific, the descendants of the Puritans sit grimly before electronic oracles, where the images of the designated celebrants combine in colorful gridiron formations. Belts are loosened, and the uncles snore in the parlor in the rich and heavy afternoon, breathing like dolphins. I made a snowman and my brother knocked my snowman down and I knocked my brother down and then we played *Snack Attack*.

Though the variables are always changing, the values remain constant.

*Harvest home! Harvest home!*
*We've plowed, we've sowed,*
*We've reaped, we've mowed,*
*We've brought home every load.*
*We are too hip, hip, hip, harvest home!*

| | Government Proclamation | Merrymaking | Feasting | Family Gathering | Tree Branches | Bonfires | Offering to Gods |
|---|---|---|---|---|---|---|---|
| Thesmophoria (Greece) | no | very heavy | yes | no | no | no | yes |
| Cruzvelacuy (Peru) | no | heavy | no | no | no | yes | no |
| Juhannusaatto (Finland) | no | moderate | yes | yes | yes | yes | no |
| Thanksgiving (U.S./Can.) | yes | light | yes | yes | no | no | no |
| Thanksgiving (Liberia) | yes | light | yes | yes | no | no | no |
| Succoth (Israel) | no | moderate | yes | yes | yes | no | no |
| Moon Festival (China) | no | light | yes | yes | no | no | yes |

Figure 1. Thanksgiving comparison chart.

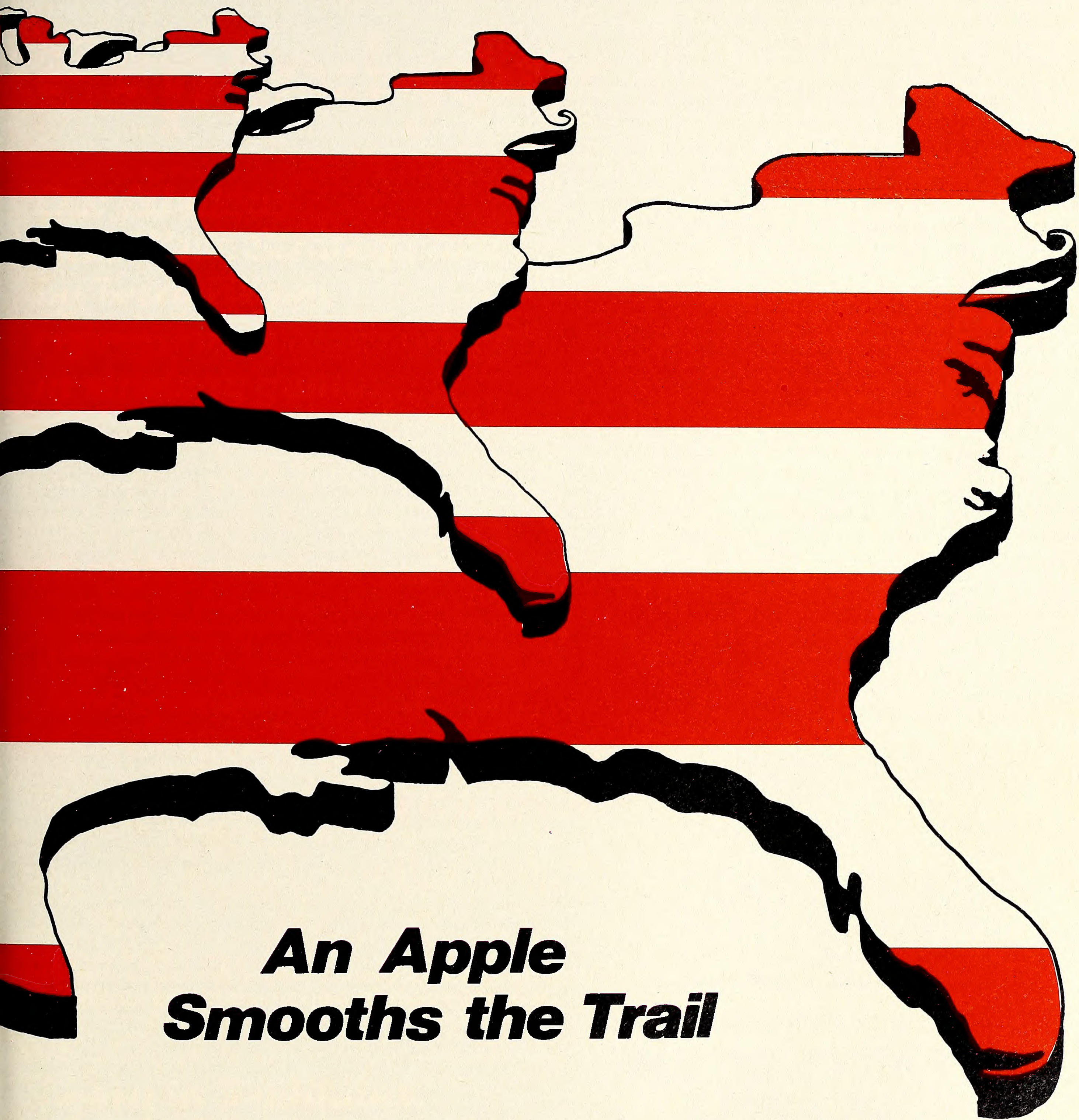
# CAMPAIGN 82

## BY LINDA A. MERRILL AND DAVID HUNTER

The history of American politics is peppered with sound and fury, war and peace. Bloody, Wagnerian battles have decided many an election, lightened only by the court jesting of a Will Rogers or a Rich Little. The great dream of Jefferson and our founding fathers survives in the elemental clash between donkey and elephant. Controlled by mysterious forces, like the movement of continental plates, Republicans and Democrats grind against each other while mountain ranges rise and fall, lakes dry up, and coastlines shrink.

Our most powerful chieftains meet in a great domed hall on the Hill. The Daleys and the Rockefellers leave local strongholds for the battleground named after our greatest warrior, Washington. They wield mighty clubs and lead loyal legions of supporters. Here the will of the common people decides, except when an assassin takes that right away. And then there is great sorrow, sorrow only a stricken people can know.

November has been chosen for the month of the greatest battles. As the autumn leaves fall, so do the hopes of many bright and young-at-



# An Apple Smooths the Trail

heart candidates. It's the way of things.

**Put Up Your Dukes.** Like the drive against the Germans in World War II, it's the campaign that wins an election. Republican and Democratic candidates usually have to survive a primary bout in June. Only the politically strongest and smartest survive. There are many skirmishes and minor clashes before this time, and more fierce confrontations follow all the way up to the final election.

This year's gubernatorial election in California is a tough one. The major candidates are battling for one of the prize gems in the empire, the proving ground for our current chief executive. There can only be one winner. With all the big guns used in this type of campaign, the presence of an Apple among the support troops on one side is worth an investigation.

Election campaigns, like wars, are only made possible with money. All those thirty-second television spots, billboards, posters, bumper stickers, fliers, radio commercials, and buttons cost bucks. The two main contenders for governor in California have each required more than two million dollars to wage media-intensive campaigns.

In an effort to keep the political game aboveboard, the Political Reform Act (PRA) was passed in 1974 by the voters of California. The PRA requires all political candidates to report every monetary donation of more than a hundred dollars. With the total amount of donations running into millions of dollars and the total number of donors adding up to thousands of people, this reporting process can quickly escalate into an enormous task.

Faced with such a responsibility, in this case for Republican candidate George Deukmejian's campaign for governor of California, Pat Formby, campaign treasurer, decided to use an Apple II Plus to help with the campaign's Schedule A reporting forms. Six times a year, these reports must be filed. The time of filing is called a *threshold.*

**The Searcher.** Shortly after he made this decision, Formby realized he would have to find someone who was knowledgeable about both computer programming and the complicated Schedule A reporting. If that person could design a program to meet his accounting requirements for the duration of the campaign, then Formby's computer could do the job for him. James Lorenz turned out to be just the right person for the job.

Lorenz, a software designer with Orion Business Systems in San Diego, California, had run for the office of California State Assemblyman in 1980. During his campaign, Lorenz had tracked his Schedule A reporting manually. When Formby contacted him about creating a program to do the job, Lorenz recognized immediately the enormity of the assignment and the timesaving and laborsaving potential if he could design such a program.

To satisfy the PRA requirements, the program has to accumulate donations so that as soon as a supporter's total dollar amount contributed exceeds one hundred dollars, that person is included in the Schedule A listing. The report must include the date of the contribution and the person's name, address, occupations, and, under certain circumstances, employer.

Lorenz designed the program to use Cobol on CP/M and contracted Fritz Sands to do the programming; thus Formby's Apple had to be configured to the task. Lorenz added a Z-80 card, an eighty-column card, a 16K RAM card, a twenty-megabyte Corvus hard disk, and two printer cards. A Qume and a high-speed TI810 were the chosen printers. The Apple system could then accommodate both Formby's Schedule A reporting and Sands's complex array of interactive programs.

A problem developed almost immediately. With all the cards that Lorenz had added, the inside of the Apple became so hot that the computer malfunctioned after a very short period of operation. To solve the problem, Lorenz cut a hole in the computer's floorplate near the speaker, moved the speaker over, and inserted a twenty-cubic-foot-per-minute muffin fan over the hole. Then he masked the ducts so that the air would flow past the slotboards, over the motherboards, through the keyboard, and out the front left. Now the Apple ran at room temperature with no overheating problems. "It's loud, but functional. We call it the Orion Turbo," says Lorenz.

**True Grit.** Lorenz is quick to point out that the Schedule A is not just a matter of who, when, and how much. One of the most valuable aspects of Schedule A reporting is the information it provides the campaigners for tracking donor records. For example, it's financially helpful to know if a multiple donor is contributing an increasing or decreasing amount of money with increasing or decreasing frequency. If either the amount or the frequency factor is increasing, that donor is a political supporter who deserves some special courting. Lorenz's program tracks this kind of information.

With the reported information on a computer database, it's also possible to get to it from whatever point you want. For example, you can get a geographical breakdown if you want to know what area you are getting the most donations from. Or it can tell you how many donors have contributed a given amount of money. Or it can tell you the amount in total donations on a given day.

One of the features that fully uses the computer's capacity to save labor time is what Lorenz calls a self-teaching zip code directory. When the computer operator enters the address of a contributor, the computer encodes in its memory the city and state that correspond to the zip code that was entered. The next time that an address has to be entered with that city and state, all the operator has to enter is the five-digit zip code. When the computer sees that code, it will automatically call up the city and state for that zip code without requiring the operator to rekey those characters.

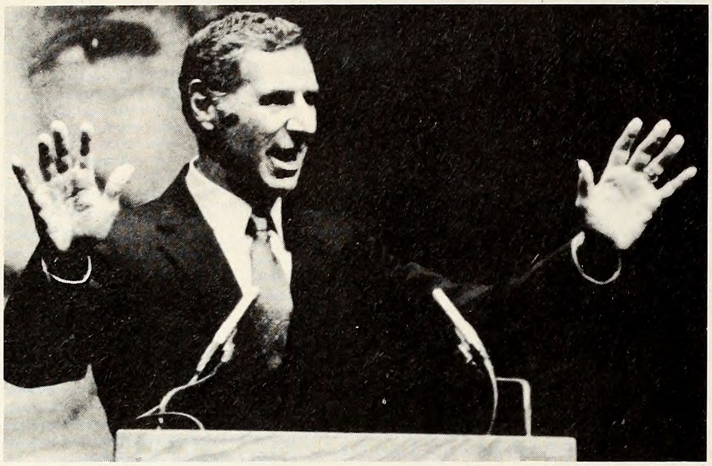
According to Lorenz, the average input per day per operator is four hundred donors. An average city and state combination is about ten characters long, so Lorenz figures that this built-in zip code directory is saving the computer operator up to four thousand characters a day.

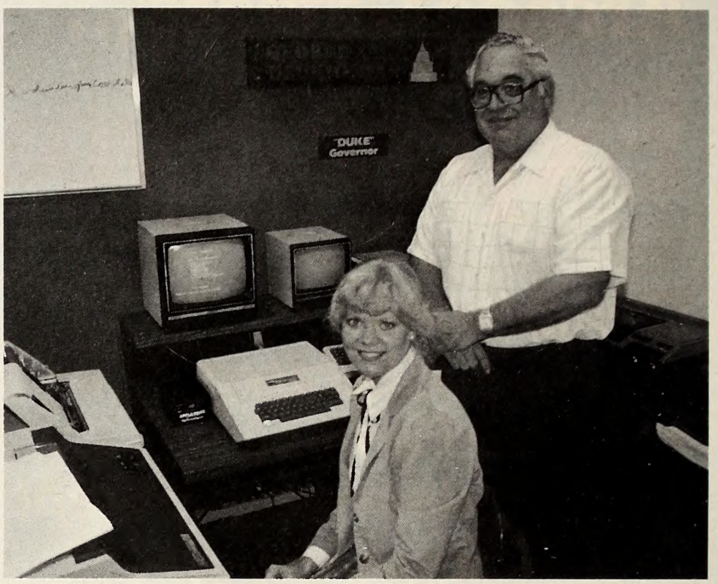
Since he is the campaign treasurer, Formby is held personally responsible for all campaign dollars. So when Formby asked Lorenz to design this program, one of Formby's requirements was that there be an audit control system built into it. Lorenz was happy to oblige.

**Stagecoach.** When Formby receives a group of contribution checks for the campaign, the checks are broken down into batches. Each batch is assigned a batch control number. Then when the computer operator enters a donation into the Schedule A database, the batch control number that was assigned to that contribution check is entered in the donor's file along with the other required information. In this way, an audit trail is created.

This batch control system gives Formby the ability to keep in his computer database the information that he must also store in his hard copy files. With the audit trail, Formby can reconstruct an entire batch of checks if, for some reason, the hard copy file that contains the xerographic copies of the checks and deposit slips for that batch is misplaced.

For example, if batch x is missing from his file drawer, Formby can ask the computer to reconstruct batch x. The computer will search every record on the database and pull the records that have batch control number x in them. When it pulls those records, the computer notes the do-






Above: California Republican gubernatorial candidate George Deukmejian. Right: Lianne Porter, Pat Formby, and Lorenz's Apple/Corvus accounting system, a winning combination.

nors' names and the contribution amounts that were entered for batch number x. At the end of the search, Formby has batch x reconstructed. In this way, the audit trail can serve as a file back-up system in the event that the paper records are misplaced or destroyed.

Lorenz's program also includes a way to handle contribution checks that bounce. If a donor writes a check that later is returned to Formby because of insufficient funds, the computer operator can go into the database and alter the batch to show that a bad check was received. The record of the donation is not deleted from the database, but the amount of the returned contribution check is subtracted from the Schedule A report. The contribution will not show on the Schedule A, but it will show on the audit trail. In this way, a person cannot make a donation with a bad check, claim a tax deduction, and get away with both the dollar amount of the returned check and the deduction.

Lorenz admits that a software designer without an intimate knowledge of the Political Reform Act and its legal ramifications would not have been able to create a very satisfactory software program. The fact that he had run for office himself made him uniquely qualified to design the program. Lorenz is proud of his creation. Having done manually in 1980 what Formby's Apple II Plus now does, he can truly appreciate his program's success.

**The Quiet Man.** As for Formby's reaction to Lorenz's program, he couldn't be more pleased. "The project has definitely been a success. I am happy that I chose my Apple computer to take care of this task for me," says Formby. "Although we can use it for blitz mailings, the program is primarily used as a reporting and fund-raising tool. Because it includes dollar amounts, the database is much more useful than a straight mailing list.

"The candidate doesn't know where the money's coming from. He's too busy. We have a list now of twelve thousand to fifteen thousand names."

Formby also points out that the database provides easy access to information needed for sending out thank-yous in response to every contribution. In addition, the data can be retrieved easily if necessary to respond to donors' tax claims.

Lianne Porter is the computer operator who has been working on the Apple for the campaign. According to Porter, "The program is saving time and labor for the campaign. It's simple to input data into the computer and easy to look up in the database if you need to find a certain piece of information.

"It's easy, but you still have to catch yourself from making mistakes," Porter adds. "I work on the Apple about six hours a day. We get enough money in each day to keep me busy."

Formby is a CPA who's "kinda retired." Working out of a two story office building across from a racetrack in Los Alamitos, California, Formby is volunteering his time as treasurer for Deukmejian's campaign. Formby has performed his present duties for Deukmejian before, in races for the state assembly, senate, and attorney general.

"The old way was to hand post everything in a general ledger book with carbon paper. If you took the information to an outside computer house, the preparations were intense and the turnaround time forever."

**Reap the Wild Wind.** The day may come when people vote from the comfort of their homes. The telecommunications revolution promises this and more. In the future, political campaigns will rely heavily on interactive videotex systems for polling. The voice of the people will be heard louder and faster than ever before.

Still, it's going to be difficult to serve up a thousand-dollar-a-plate dinner through a monitor. The traditional forms of political parlay and maneuvering will never completely disappear. The wars will just escalate, bringing in more participants and creating more casualties.

It's the way of things.


SHEILA NEEDS
YOUR HELP
BEFORE IT'S TOO LATE
SHEILA
Written by Brian Fitzgerald
H.A.L. LABS
For use on the Apple II, Apple III, or Franklin Ace 100
It's only been an hour since he locked her in the tower . . .
Leap into the mythical, hi-resolution world of SHEILA . . . it's the real-time adventure saga that's littered with traps, puzzles, magic spells and various monsters. You control a sophisticated arcade-like arsenal of bombs, spears, teleport pills, magical devices, and spells to free SHEILA from the forbidden tower — and save the world from eternal Hell.
SHEILA is the brand new fantasy game that combines full-color graphics and subtle strategy better than any videogame you've ever played. So get set for endless hours of fun and surprises from SHEILA . . . simple to play but a real challenge to win! Ask for SHEILA by name today at your favorite software dealer, or order direct.
SHEILA is available for the Apple II, Apple III, or Franklin Ace 100.
© Copyright 1982. Dealer inquiries invited.
If ordered direct send check or money order for $25.00 plus $1.50 postage (Calif. residents add 6% sales tax).
Written by Brian Fitzgerald
H.A.L. LABS
4074 Midland Road, Suite 23, Riverside, Calif. 92505 (714) 359-8480
Apple is a registered trademark of Apple Computer, Inc. Franklin Ace is a trademark of Franklin Computer Corporation.

# Explore the Frontiers of Intelligence

## THE MIND OF MAN

**Chess**, **Checkers**, and **Odin** (as played by U.S. Othello Association rules) are classic games of the intellect. They evolved over the centuries as a way to understand complex situations and achieve mastery in action.

Now, by interacting with the intelligence embodied in these programs, you can participate in the exciting challenge of the **Mind of Man.**

## FOR ALL GENERATIONS— A NEW GENERATION OF INTERACTIVE INTELLIGENCE

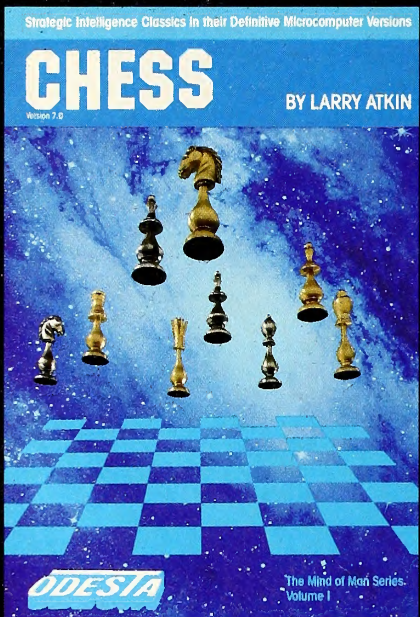
**Chess**, **Checkers**, and **Odin** are unique—in playing strength and in what they let **you** do. You can try out any idea—and even get ideas from the programs themselves. Plus, it is easy and fun to play the games and use their many features:

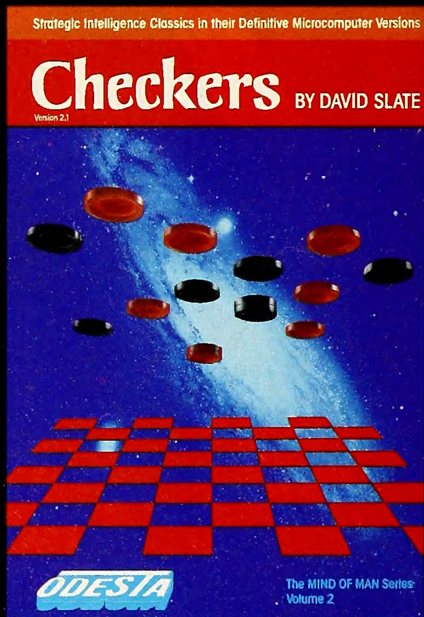
• Different levels of play, from beginner to expert • Advice on best move • Take back and replay moves • Auto and manual modes • Instant replay of games • "Change" feature adds or subtracts pieces • Sophisticated opening libraries • Clear graphics • Instructions include chapter on skillful play.

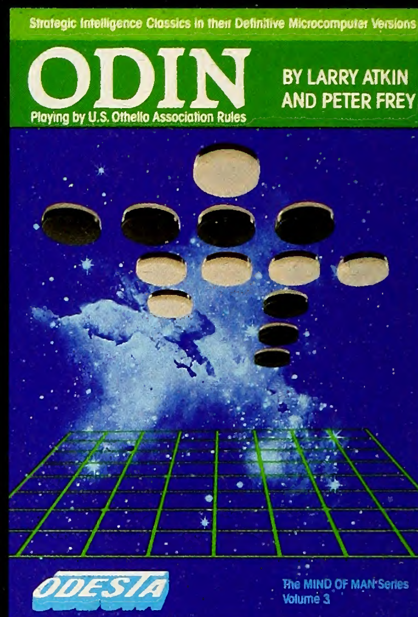
## THE PEOPLE BEHIND THE PROGRAMS

**LARRY ATKIN AND DAVID SLATE**
Authors of the Northwestern University 4.7 Computer Chess program; Winners of the World Computer Chess Tournament, 1977-1980; Winners of 8 North American Computer Chess Championships between 1970 and 1979; Two of the world's authorities on machine intelligence.

**PETER FREY**
Professor at Northwestern University, teaching courses in Psychology and Computer Science. Editor of and contributor to the definitive text on computer chess: **Chess Skill in Man and Machine.** One of the U.S. Othello Association's top-ranked players.




## CHALLENGES THE PROFICIENT; INSTRUCTS THE BEGINNER.

A new microcomputer standard for what many consider to be the ultimate game of the intellect. In addition to its superior playing strength, **Chess** from Odesta introduces a new generation of interactive intelligence—with 27 cursor-controlled features, including:

• Advice and prediction of best moves • Save games to disk • Graphic illustration of attacks and defenses • Variations of blind-fold chess • 17 levels of play, including postal and mate-finder modes • Enactment of over 30 classic human and computer chess games • Plus—an opening library of over 7000 moves.

For those who want the best.

## YOU'VE NEVER KNOWN CHECKERS LIKE THIS!

Learn the complexities of this surprisingly sophisticated strategy game by interacting with **Checkers'** 24 user features:

• Play against 16 levels of difficulty • Watch **Checkers** play against itself—one level against another • Switch to "Give-away" mode, where the object is to make your opponent take **your** pieces • Watch the **Checkers** movie—an instant replay of a whole game • For those interested in the inner-workings of "programs that think", adjust 58 program parameters, so that **you** can experiment with the way **Checkers** itself thinks, and how it plays.

An ideal introduction to artificial intelligence.

## WHY ARE 20 MILLION PEOPLE PLAYING THIS GAME?

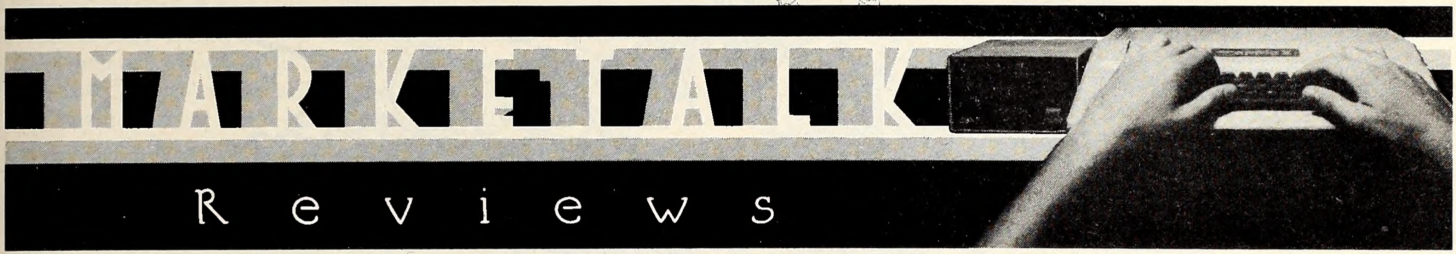
A classic board game, where the object is to entrap your opponent's pieces—but only at the right time. The rules of play are simple and the game fast. At your disposal are the full range of features found in all of Odesta's **Mind of Man** series. You will need them, along with your keenest insight and deepest perception, to master the secret of **Odin** (playing by U.S. Othello Association rules). Join the growing number of strategists exploring the subtleties of a game that may change the way you think.

ODESTA
930 Pitner
Evanston, IL 60202
(U.S.A.)

Chess: $69.95
Checkers: $49.95
Odin: $49.95

See your local software dealer, or order (Mastercard or Visa):
800-323-5423
(in Illinois, call 312-328-7101)

For **Apple II**, **Apple II Plus 48K** disk systems, and **Atari 48K** disk systems. Odin is also available for **TRS-80 Model 1 & 3 32K** disk systems.




*Unless otherwise noted, all products can be assumed to run on either Apple II, with 48K, ROM Applesoft, and one disk drive. The requirement for ROM Applesoft can be met by RAM Applesoft in a language card. Many Apple II programs will run on the Apple III in the emulator mode.*

**Arcade Machine.** By Chris Jochumson and Doug Carlston. Somewhere trumpets blare, crowds of hobbyists cheer, and game manufacturers cringe. *Arcade Machine*, after an eleven-month delay, has finally arrived. Although this program does not perform black magic at the wave of a joystick, it certainly shines alone in territory no one else has walked. *Arcade Machine* produces (with some help from you) exciting, animated, graphic games. These games can have as polished a professional look as many of the games you see in the stores.

If you've ever been enjoying a game but wished it could do this instead of that, you'll thoroughly enjoy custom-designing your own games to suit your moods and tastes—and the *Arcade Machine* requires just enough of you that you'll feel immense satisfaction and pride in your accomplishment. And therein lies the other side of the coin.

Nirvana is not quite at hand. Even though you need no programming knowledge, using the *Arcade Machine* requires that you learn quite a bit about the way graphic games are constructed. In fact, you'll soon come to appreciate the enormous skill and effort that the authors of home-arcade games put into them. This program handles at least nine-tenths of the work for you, yet the remaining tenth will take several sessions to learn and longer to master. It is not that any of these concepts or procedures are difficult in themselves but that there are so many of them. Still, *Arcade Machine* strives to make things as easy as possible for you to learn; the manual even contains layout sheets for practice.

The main menu contains a shape editor to determine what your pieces are going to look like, a path creator to implement how they are going to move on the screen, a game options section, a level options selector (up to five), a background/title creator (here is where you get to put *your* name), a load/save game function, a run-game mode for testing as you are created, and, finally, a create–game-disk feature for making copies of your finished product.

Each of these sections contains other submenus, where even more incredible options are considered. One of the nicest features is the option to make any game a two-player game. Shapes and backgrounds from different games can be merged into new games, allowing for updating and improvement in case you tire of your old games.

You needn't be able to draw well to use the *Arcade Machine*'s graphics. For example, tell the computer where you want the center to be and where the outside should go, then hit O and, presto, a circle. Triangles and rectangles are similarly easy. Another command draws them and fills them in with color at the same time. Color on the Apple is not a simple matter. When dots are placed side by side, they look white. Only when they're staggered do you see color on the screen. Different combinations of the basic colors yield different colors. This can take some getting used to when designing your character shapes.

To show the range of the *Arcade Machine* and possibly to provide prototypes, the back of the disk contains five games generated by the *Arcade Machine*. A couple of them, like *Pizza Man*, are quite remarkable.

Naturally *Arcade Machine* does have some limitations; for instance, it doesn't provide for designing maze games. But it is the manual itself that is the program's biggest limitation. Perhaps Broderbund will produce a supplementary manual to clarify some of the more obscure points. (Hint, hint.)

Broderbund is encouraging amateur game makers by sponsoring a contest. Starting January 1983 and ending in June, Broderbund will award two hundred dollars' worth of prizes to the best game design produced with the *Arcade Machine*. The monthly winners will compete for a fifteen-hundred-dollar grand prize. Who knows, the game you create may even be good enough to publish! If not, at least you'll have the fun of testing your game-designing instincts against the pros.

Using the *Arcade Machine* isn't a cinch. But it's a lead-pipe cinch you'll never get more for your money—or more for your effort. RRA

*Arcade Machine*, by Chris Jochumson and Doug Carlston, Broderbund Software (1938 Fourth Street, San Rafael, CA 94901; 415-456-6424). $44.95.

**The Mask of the Sun.** By Chris Anson, Alan Clark, Larry Franks, and Margaret Anson. The pioneers who opened up the Midwest and West of our country lived primitively compared to those who remained in the established cities. New frontiers often are characterized by primitive ways, yet they are the great breakthroughs that lead to progress.

In *The Mask of the Sun*, a new company shares with us major breakthroughs in adventure graphics; but, if we are to appreciate most this welcome into a new frontier, we must accept some compromises. It's well worth it.

When you play *Mask*, you will never see a picture fill with color on screen. You move through fully colored graphic scenes; if once in a while the color bleeds a little, you're not apt to care. Not only because of the ready fill, but because move is the precise word.

In *Mask*, when you say "Go west" or "Left," you don't flash to another place, you move there—watching the scenery go by on the way. This is most exciting and effective when you're approaching something significant and can see it in the distance. The fact that most of this action takes place in the rather gamely tunnels of old pyramids does not take away from the importance of the concept.

On top of this, individual scenes are animated. A statue transforms before your eyes; an animal attacks; a carved face talks; an island submerges and arises in molten lava. Often you must interact with the animation. Like in *Escape from Rungistan*, a prompt appears during the animated sequence and you must act quickly to avoid dire consequences.

All these fine effects are plenty to make up for substandard drawing on the graphics. Fortunately, they're not called on for this in *The Mask of the Sun*. The pictures are full-bodied, well done, detailed illustrations, surpassed in the world of adventure only by Penguin's new *Transylvania*.

Sound effects occur occasionally, like the animation. When they occur they are appropriate and add to the fun.

Now for the meat—the adventure itself. It's a good one. The puzzles vary in difficulty; some appear quite obvious. Don't be overly confident about these, though. The great Russian playwright Anton Chekhov once said, "Never hang a gun in the first act if it isn't going to go off in the third." A puzzle easily solved, an apparently innocuous statement casually made in the beginning of *Mask* may well be the sole key to a crucial puzzle toward the end. This is absolutely delightful. Let nothing escape you; count nothing irrelevant.

The people at Ultrasoft claim an ambition for and a dedication to creating the finest software possible. Their willingness and effort to implement suggestions from beta testers attests to their claim. As their first release, *The Mask of the Sun* is vivid evidence. It's not perfect, but neither was the aircraft at Kitty Hawk. A year from now we may look at an old graphics adventure and say, as we watch the colors fill, "How come they released it before it was finished?"

We anticipate great pleasure and more surprises following the progress of this innovative company.

*The Mask of the Sun*, by Chris Anson, Alan Clark, Larry Franks, and Margaret Anson, Ultrasoft (24001 S.E. 103rd Street, Issaquah, WA 98027; 206-392-1351). $39.95. MCT

**Seafox.** By Ed Hobbs. The first few minutes of this fast-paced arcade

# BE CHALLENGED

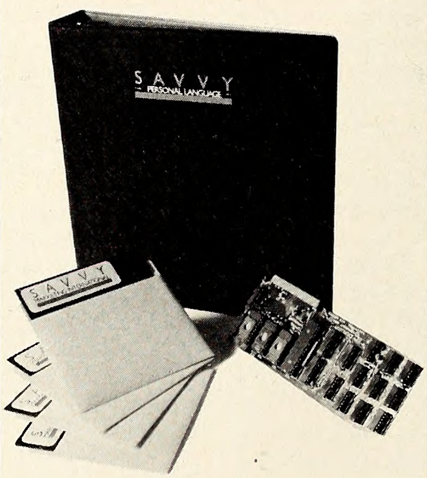
If you are a talented micro computer specialist, you follow the field closely, you know what's been done and what's expected. —

But you've never seen anything like

## SAVVY™

Nobody has.

- A CO-PROCESSOR for the Apple II, with 64-Megabyte Associative Memory, 26 Decimal Digits of Precision, and Hardware Security.
- AN OPERATING SYSTEM that automatically Loads, Links, Overlays, and Executes Application Programs.
- A DATABASE MANAGER that automatically Blocks, Allocates, Opens, and Closes Data Sets.
- MACHINE INTELLIGENCE that automatically Resolves: Program Names, Item Names, Data Types, File Structures, Folder Names, and Instructions to the Robot Programmer™.
- A COMPILER that Produces Self-Loading, Self-Relocating, Serially Reusable Code.
- A PROGRAMMING LANGUAGE that uses Plain-tex Language and Machine Intelligence.
- AN ASSOCIATIVE NETWORK in which Machine Intelligence, Operating System, Compiler, Database Manager, Programming Language, and Application Programs are all interconnected in the 64-Meg Associative Memory.




Even the developers of this technology haven't seen the end of the possibilities. Accept the challenge of working with SAVVY, and it can become whatever you make it.

**Your innovative applications that explore the power of SAVVY will be used in future publicized promotions, with credit given to the developer.**

To own SAVVY for $950.00 contact your local Apple dealer.

To accept the challenge, contact Excalibur, the developers of SAVVY, for detailed information on this remarkable system.

Excalibur Technologies Corp.
Albuquerque, New Mexico
(505) 242-3333

**SAVVY Marketing International** markets and distributes SAVVY the Personal Language™ System.

TM - SAVVY, Robot Programmer: Excalibur Technologies Corp.

TM - Personal Language is a trademark of SAVVY Marketing International.

**Excalibur**
TECHNOLOGIES CORPORATION

game might remind you of *Star Blazer*, but in an underwater scenario. Then the many differences and unique features become apparent. *Seafox* is a submarine whose duty it is to sink enemy shipping. Needless to say, the enemy frowns on this activity and exerts tremendous effort to stop the *Seafox*. The sea is full of enemy subs that try to ram you. On advanced levels these subs also fire torpedoes at you. On the surface, merchant ships cruise slowly by, protected by a screen of faster moving Red Cross ships. (Judging by the nonstop flow of these hospital ships, our side must be trouncing the enemy on other fronts.)

The *Seafox* has two weapons—torpedoes to fire forward and Polaris missiles to fire upward. One really nice feature, which *Star Blazer* didn't offer, is your ability to shoot in both directions at the same time. You don't have to wait to rise to a certain level before you can fire the Polaris missiles. You have, however, a limited amount of ammunition. Supplies and gas—the *Seafox*'s tank depletes rapidly—arrive regularly via a supply ship that comes along the bottom and are released on the back of a dolphin. The *Seafox* must intercept the package before giant clams steal it or enemy submarines destroy it. If you shoot the dolphin, an indestructible giant orange shark roars from the sidelines and gobbles up the *Seafox*.

In *Seafox*, the Red Cross has realized the vulnerability of its ships. The hospital ships now have antimissile shielding on their hulls; any missile striking one is immediately defected downward at the sub firing it. The returning missile is easy to avoid; the penalty resides in the time lost as it slowly makes its way to the bottom. Since no missile can be fired in a single direction until the first shot detonates or disappears, you can miss opportunities. Sinking all the enemy merchant ships completes a level.

On the second level, enemy destroyers join the surface parade, dropping depth charges that explode at the level you occupied when they were discharged. You can turn this peril to your advantage by maneuvering so that the charge will explode on enemy subs. As the underwater arena becomes crowded, the enemy depth charges are as likely to blow up their own subs as yours.

On the third level, the enemy subs add torpedo fire to the fray, and on the fourth, they discharge magnetic mines. The fifth, final level? You tell us.

With *Seafox*, Broderbund is cementing its current leadership in the ratings: this game is highly addictive and a good challenge. RRA

*Seafox*, by Ed Hobbs, Broderbund Software (1938 Fourth Street, San Rafael, CA 94901; 415-456-6424). $34.95.

**Starcross.** By Dave Lebling and Marc Blank. If it takes a minimum of two instances to form a proposition, then *Starcross*, adventurous Infocom's first foray into science fiction and second departure from the dungeons of *Zork*, enables the proposition that this young company is one of remarkable versatility and apparently unerring ability to implement fresh and fitting approaches to multifarious subjects. *Deadline*, the you-solve-it mystery, was, of course, instance one.

*Starcross* resembles *Deadline* and the *Zork*s only in the benevolent intelligence that is a hallmark of Infocom's work. In every other regard, it serves its own identity—that of a sci-fi situation and story. The scope, the structure, the language, the events, the requirements on the player are all dictated by the genre and its time and setting.

Where the *Zork*s are set in the fantasy world of Dark Ages magic and myth, where *Deadline* is set in the very mundane world of the Britishlike parlor murder mystery, *Starcross* is set in the no less fantastic and no more obvious world of the possible future.

As the game begins, you find yourself on a one-person spaceship, searching for black holes. With you are only your smart-aleck computer and your mass detector. Amazingly, when the mass detector proffers you a pictorial map of nearby space, you the real person can reach in your *Starcross* package and pull out the exact map in four glossy colors. You'll need it to direct the computer as to the course to take to your destination (why on earth didn't you pop for the gadget that interfaces the mass detector directly to that cantankerous computer?).

But instead of a black hole, you find an asteroid-size spaceship, the kind that might be devised to transport entire civilizations to new horizons when their own corner of the universe is becoming uninhabitable.

What you find there and what you do with it all are the meat of this outstanding adventure. As with so many excellent adventures, determining your purpose is an integral part of the plot.

*Starcross* is painted in brilliant colors with ultra-high-resolution detail; as usual, Infocom has created these splendid graphics entirely with words. This is a text, or, as the authors prefer, and appropriately so, a prose adventure.

No puzzle in *Starcross* is illogical, although many require good imagination and truly innovative thinking. Picture facing an apparent dead end. Nothing you'd expect to work solves this puzzle. You mull the problems and all the pieces a long time, and finally an idea comes. It's complex, it's far out, it's got to be out of the question—no one else would have thought of that—but it's thoroughly logical. Just for kicks, you try it, expecting another dead end. Instead, the program understands, responds—you weren't crazy; those guys actually think like you. But wait—they've taken it even farther; you'll have to stretch your mind even more. Like most good stretches, it feels delicious.

Infocom's adventure vocabulary continues to expand. The famous *Zork* parser allows you to speak to the Apple as though it were a person. When it doesn't understand, it makes perfectly clear what it doesn't understand. No guess-the-right-synonym games here.

In the realm of science-fiction adventure for the micro, only *Cyborg* holds a candle to *Starcross*. Both deserve places in the computer sci-fi hall of fame. A year younger and benefiting from Infocom's extraordinary technical tools, *Starcross* is the smoother, the more playable; *Cyborg* still wears the crown for plot.

No adventurer should pass up *Starcross*. And, if you've never ventured into this kind of game, *Starcross* is a good place to start. MCT

*Starcross*, by Dave Lebling and Marc Blank, Infocom (55 Wheeler Street, Cambridge, MA 02138; 617-492-1031). $39.95.

**Sensible Speller.** By Charles Hartley. *Sensible Speller* easily merits a place in the running for best program of 1982, based on five features.

First, it is the most comprehensive dictionary yet available for the Apple. Not being satisfied with its earlier, field-leading thirty-one thousand words, Sensible Software has brought the entire Random House Dictionary, Concise Edition, to the rescue of bad spellers everywhere.

This rather staggering vocabulary of seventy-five thousand words resides on two disks. There is also room to add ten thousand words of your own choice.

Second, this spelling verification program works with most word processing packages available for the Apple. The company is putting out four versions of *Sensible Speller*. The DOS 3.3 version works in conjunction with *Screen Writer II*, *Apple Writer I* and *II*, *Apple PIE*, *Magic Window*, *Executive Secretary*, *Text Editor*, and *Word Power*. It will handle any word processor that generates standard DOS 3.3 text or binary files. There are also Pascal, CP/M, and *Super-Text* versions available.

Third, the program does much more than verify spelling. It tabulates the total number of words used in the entire text and the number of unique words (counting only the first use of every word). The printer option prints alphabetic lists of both, with the number of times each word is used printed next to the word. This is a tremendous boon to the professional writer or editor, making it easy to spot overusage of words and phrases. Each potentially misspelled word is screened "within context" (showing several words before and after the problem word in the manuscript), and the program provides for wild card search of the entire dictionary to locate the correct spelling.

Fourth, the dictionary is upgradable through the program. New words can be flagged during the verification process for later automatic addition to the dictionary proper. While the main program is protected (a back-up copy is included), the two dictionary disks are not, so the frequently used main files can be backed up for safety. When the dictionary adds the new words, it automatically re-sorts all files onto a new disk, adding the new words in at their proper place.

Fifth, the package is special. The looseleaf manual has a special pouch on the inside front cover holding a hardback copy of the complete Random House Dictionary used on the disks. This provides valuable support for the program by affording easy access while writing, thus cutting down on the verification time. Special plastic pages hold the four disks that come with the program, so everything is kept neat and orderly.



Yet, however good this program, it doesn't eliminate the need for human editing. As the manual notes, the sentence "Dead ewe right too hymn?" will be verified without question. Also, errors in syntax will not be picked up by the computer.

*Sensible Speller* is a topnotch professional package that belongs on your desk if you're involved in word processing. RRA

*Sensible Speller*, by Charles Hartley, Sensible Software (6619 Perham Drive, West Bloomfield, MI 48033; 313-399-8877). $125.

**Honinbo Warrior.** By Mark Watson. There's a lot of game software out there—programs that play some kind of game against you; programs whose purpose is to be an "opponent on disk" whenever you want to play. There are programs to play you at chess and checkers, bridge and blackjack, tick-tack-toe and twenty questions—not to mention those pushy young upstarts like *Pac-Man* and *Wizardry*. But in all this abundance, one game is missing: a game that is older than chess, and at least as popular. There is no program that will let you play go with your Apple—until now.

In case you're not familiar with go, it's a Chinese/Japanese board game, somewhat similar to chess. As in chess, the pieces used by the two players represent units of two armies in combat, and each player's task is to position his pieces so they can capture the enemy—but there the resemblance ends. Go is to chess as a diagramless crossword puzzle is to a normal one: the structure develops during the game.

For this and other reasons, it's difficult even to describe the game to a computer in a useful way, let alone ask the computer to evaluate possible moves. The "value" of a particular move at a particular point in a particular game depends on two and only two things: the relative location of each stone played so far—probably about a hundred of them—and the relative location of every stone yet to be played (another hundred or so).

How do you tell a computer about that?

The result is, if you're a go player, that you either find "soft people" (flesh-and-blood humans) to play with, or you go without. Which can be difficult, for a beginner.

But that, as they say, was yesterday. *Honinbo Warrior* is no superhero, but he knows the moves. Now you can learn the basic patterns of play and experiment with them to your heart's content, with a tireless sparring partner. Now you can play go with your Apple!

It won't be a master-level game. Honinbo-san is what you might call a well-trained novice: he knows how to place his stones properly, but he falls into traps a lot. However, from a beginner's viewpoint, that's not at all bad—you want to be able to beat the teacher every so often. It's very satisfying to see the message, "Honinbo resigns," even after a fairly easy game.

On booting the disk you will see a title page, which is followed (press any key) by several pages of introduction and instructions. Finally you will come to a go board, with the columns lettered A through N across the bottom and the rows numbered 1 through D (hex) up the side.

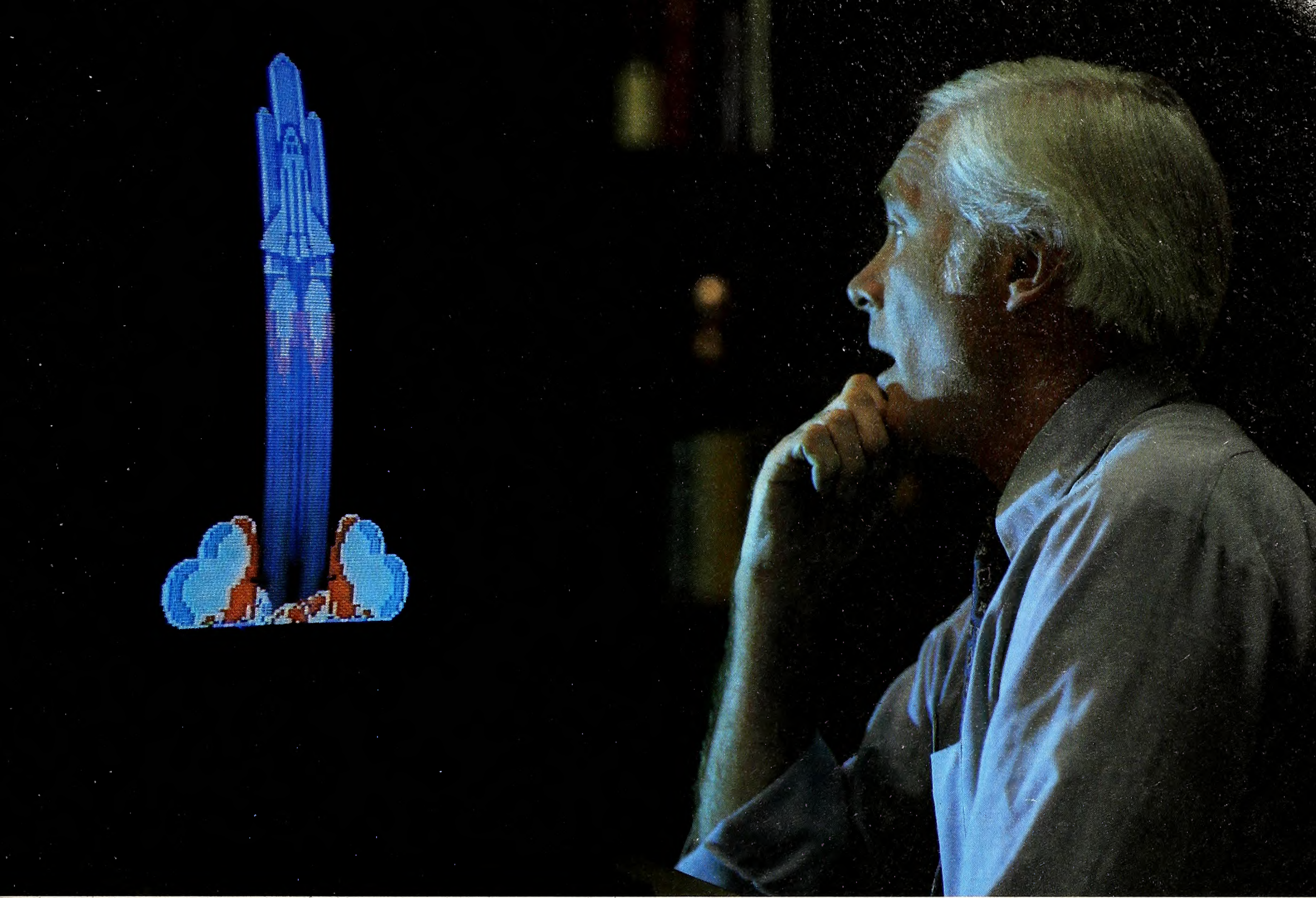
That's smaller than a normal go board (nineteen by nineteen) and not much use for heavy exercises in strategy; but it gives plenty of room to practice your local tactics—you can have four or five attacks going at once—and it's a size the Apple can handle in reasonable time.

A moment after you see the board, Honinbo will announce his move: the message "C4 Honinbo" will be displayed, and a black stone (actually a blue circle) will appear on the board at the intersection of vertical line C and horizontal line 4 (sometimes Honinbo will open on C3 or D4 instead, but he always uses one of the classic corner openings).

Then you will hear a short musical phrase from the speaker, reminiscent of a battlefield trumpet-call, which you will soon learn to recognize as Honinbo-san's way of saying, "Your move!"

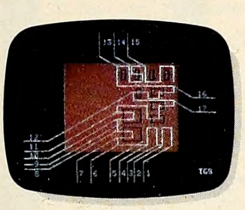
Type in the column and row designations of the point you want to play on. Your move will not be entered until you hit return—if you have second thoughts, press the back-arrow two or three times and start over.

When you press return, your move will be listed under Honinbo's, and a white stone will appear on the board at the designated point. Honinbo-san will think about it for a while and then announce his next move as before. Frequently he will also add a little comment, such as "I want more territory!" or "You think you can invade?" or—after you've pulled off something particularly neat—"Smallpoint!"



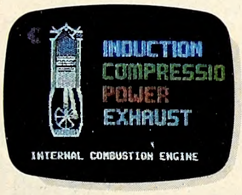
# THE GRAPHIC SOLUTION

Solve your toughest communication problems with the Graphic Solution™, a sophisticated, new graphics package from Accent Software.

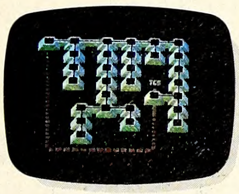


With precise, multi-speed ANIMATION create captivating sales presentations and product demonstrations that will both intrigue and inform your clients and customers. Watch their reactions; you'll see your messages getting through.

Develop educational materials and training aids that MIX TEXT AND GRAPHICS on the screen, breathing new life into abstract, hard-to-grasp concepts. Mix programs too. Images can be displayed on backgrounds loaded from any of your other programs. Construct custom TYPEFACES AND TYPESIZES to balance the visual elements.




Tired of run-of-the-mill business graphics? Change standard charts and graphs into colorful THREE DIMENSIONAL PERSPECTIVES. Add text and animate the data to show the relative rates of change for your most important information. Like cash flow projections. Or revenue estimates.



Plot flowcharts, time and motion studies, industrial process flows with COLOR-CODED ELEMENTS highlighting critical paths. Animate the sequences to show how flows actually progress.

Work with live action? Prepare film and videotape storyboards using the unique FRAME-BY-FRAME graphic sequencer that lets you create and animate a video story before shooting.

Whatever your graphic communication demands—in the business world, the arts, industry, education—The Graphic Solution™ at $149.95 has the answer. Take a hard look at The Graphic Solution. You'll like what you see.


The Graphic Solution requires a 48K Apple II with ROM Applesoft and DOS 3.3.

See your local dealer or send $10.00 for a demonstration diskette to:

**ACCENT SOFTWARE, INC.**

3750 Wright Place, Palo Alto, Calif. 94306 Telephone 415-856-6505

# Sign Me Up!

As an Apple owner, you're entitled to a free trial subscription to Softalk. If you've never received Softalk, merely fill out this card and mail it in. If you have received Softalk in the past, or are receiving it now, you can use this card to renew or to register a change of address.

☐ I'm a new Apple owner.

☐ I'm using this card to renew my subscription. Please find enclosed $24 for a one-year subscription.

Apple Serial Number

Name Street Address

City State Zip

☐ Above is my new address. Below is my old address:

Name Street Address

City State Zip

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

# BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 673 NO. HOLLYWOOD, CA

POSTAGE WILL BE PAID BY ADDRESSEE

SOFTALK
P.O. Box 60
North Hollywood, California 91603

The system takes care of saying "atari!" (a go word similar to "check!" in chess) for both sides and uses another musical phrase, this one with overtones of panic. It also manages captures, removing the dead and lining them up beside the board. In other words, it does pretty much what you'd expect of a go-playing program.

And it includes a neat personal touch. Your name is written permanently into the program when you buy it, so you are always listed automatically as Honinbo-san's opponent.

This is a byproduct of the author's elegant approach to the matter of copy-protection. You can make as many copies of the disk as you like—but they will all have your name on them! Whenever one of your moves is listed during a game, it is credited to the buyer of the disk. And the program includes several hidden validity checks, so unauthorized tampering will crash the system. (Note: for this and other reasons, *Honinbo Warrior* is not distributed through retail stores. It is only available direct from the author.)

So much for the good stuff. It must be said that the program is not perfect—in fact, it's not finished. This is version II, and III is expected early next year (updates will be available, at a nominal cost, to purchasers of II).

There is a small bug in the program. When you capture a stone by playing on the edge of the board, the system sometimes gets confused and redraws the captured stone on the point it was taken from, still surrounded by your men. However, the fault seems to be in the display only: the system logic knows that the black stone is not really there, and it will let you play a stone on that point if you choose to.

Another problem is that there is no manual for the program. You get a disk and nothing else. However, system operation is so simple that you don't really need a book with this one—except, of course, a good book on go, which you get on your own.

Or, to make a long story short:

Once upon a time there was a man who traveled a great distance to see a horse which could, it was said, walk upright on its hind legs. The horse performed as advertised, but with much wobbling and little grace. When asked about this, the man replied: "How well it is done is not the issue. What is remarkable is that it should be done at all!" JR

*Honinbo Warrior*, by Mark Watson (535 Mar Vista, Solana Beach, CA 92075). $21.

**Star Maze.** By Gordon Eastman, based on an original game design by Robert Woodhead. From a technical standpoint *Star Maze* is an incredible accomplishment. Utilizing eighteen hi-res colors, it features multidirectional scrolling. As the position of your ship changes, the speed at which the screen ripples is something to behold. The object of the game is to recover nine power jewels randomly scattered around a large maze. This is not your ordinary puzzle, however, as the maze is constructed of passageways through and around giant multicolored marshmallows of varying shapes and sizes. Anything that hits these marshmallows bounces off them at ninety degrees, unless your ship is going slowly; then it sinks partway into the wall before bouncing back. Shots fired from your spaceship carom off the walls. It is also possible to be going up a narrow corridor and, by angling your shots, lay down an incredible crossfire in front of you. Your ship has unlimited shots and an auto-fire button. If you are surrounded by aliens, you can either jump around the maze in hyperspace at the cost of fuel, or you can use one of your few antimatter bombs to wipe out every alien on the screen. When the jewels are recovered, they must be returned to your mothership for credit. Refueling is also possible at the mothership, but it is a rather tricky maneuver matching speeds.

Speed control is really the crux of this game. Although there are nine varieties of alien enemies that have to be fought (some of whom mutate upon being hit), the focus of the game is on controlling the speed and direction of your ship through the maze in search of the power jewels. This is very difficult to do, as the slightest thrust in the wrong direction can amplify your velocity and make the ship very difficult to slow down and control. The speed of the ship goes from 0 to about 980 and is measured along each component of the x, y axes. To stop and change direction means nullifying all of the component vectors. To make things more fun, the ship has limited fuel that can only be replaced at the mothership, if you pass through the ship at a speed of under 100. The fuel gauge on this ship resembles that of an old Chevy: for the first half of the gauge it goes really slowly, but the last half goes really fast. Being annihilated for running out of fuel is the most common cause of destruction in this game.

Capturing all nine of the jewels gains you the next level. There are sixteen different mazes, and winning your way through all earns you the title of ace. Unfortunately, the game lacks a level indicator, so the player has a difficult time keeping track of how many more levels he has to go to win the title. The successful player will have to have a very delicate sense of control to tame the ship, but then it can be a really fun game to play. RRA

*Star Maze*, by Gordon Eastman, Sir-tech Software (6 Main Street, Ogdensburg, NY 13669; 315-393-6633). $34.95.

**Prism.** By Mark James Capella and Ronald N. Roberts. Everyone loves a good fairy tale, and, when you combine one with a fabulous treasure hunt, the result is a very exciting product. *Prism* is the first program in International Software Marketing's Storydisk line. The tale is woven around the adventures of a young boy, Hubert, as he strives to recover the three stolen "keys of color." These keys, each representing a primary color, activate the mystical prism. The prism functions as the source of all the color in the world, and with its deactivation, the whole world is drained of color. The story is told in text and thoroughly illustrated in hi-res panels with some very slick animation.

After many heroic episodes, Hubert manages momentarily to retrieve the three keys from the evil Lord Grane of Yolsva, but at the last moment he loses them and they are scattered across the face of the earth.

Herein lies the treasure hunt. ISM has actually hidden three real keys somewhere in the continental United States. The yellow key is ten karat gold with a three-quarter carat topaz on the handle. The red key is fourteen karat gold with a three-quarter carat ruby, and the blue key is eighteen karat gold with a three-quarter carat blue-white diamond. The first person who physically finds one of these keys is the winner of that key. The story contains clues to the location of each of these keys. Some of the clues are readily apparent; others are diabolically hidden within the story. There are meanings upon meanings, enough to keep puzzle solvers going for quite a while.

Only when the three keys are physically united with the prism, which resides at ISM, will the conclusion to the *Prism* story unfold.

The reading level is geared for ten-year-olds to twelve-year-olds, and younger children enjoy watching the pictures and having the story read to them. With the treasure hunt thrown in, this can be fun for the whole family. RRA

*Prism*, by Mark James Capella and Ronald N. Roberts, International Software Marketing (Suite 421, University Building, 120 East Washington Street, Syracuse, NY 13202; 315-474-3400). $19.95.

**Odin.** By Larry Atkin and Peter Frey. Its origins are obscure, a condition probably contributed to by the several name changes it has undergone over the years of its existence. It has been called *Annexation, Reversi*, and most recently, *Othello*, and has undergone changes in rules as often as changes in name. The most recent incarnation has given it both renewed popularity and a degree of standardization; tournament play is now regulated by the United States Othello Association.

For those unfamiliar with the board game, the rules are simple. The playing field is an eight-by-eight square field. The playing pieces are small plastic disks, black on one side and white on the other. One player plays black and the other white, and in the initial setup the four innermost squares are covered by one piece each. The pieces on two opposite corners of this innermost square have the white side up, and on the other two pieces the black side is up.

The white player goes first, placing a piece white side up on any square where, with other white pieces, it will surround one or more black pieces. The word "surround" is being used here in a two dimensional sense. Any number of black pieces in a horizontal, vertical, or diagonal line may be surrounded by two—and only two—white pieces positioned at the ends of the line, one of which must be the piece just placed. The placement of a single piece may cause opposing pieces to be surrounded in any or all of eight directions around it. The black pieces surrounded are then flipped over to become white pieces. The black player then does the same thing, attempting to surround and flip the white player's pieces. The players take turns until either the board is filled with pieces or neither player can move, at which point the player with the most pieces on the board is the winner.

See? Simple. Actually, it is much easier to understand with a board in front of you, or, in the case of *Odin*, a computer screen. Odesta has developed a program that not only plays a devastating game of *Othello*, but also helps you get the hang of it as you go along. It is like playing against a brilliant opponent who is also a patient teacher.

All input is accepted through a single game paddle, or the arrows and the return key if you don't have paddles. Turning the paddle dial or pushing the arrow keys moves a cursor on the screen through the board positions of all your possible moves and a menu of other options that are designed to provide maximum flexibility. Pushing the paddle button or the return key either accepts that position as your move or selects that option. This is an excellent method of input. It doesn't stand between you and the game.

The menu options available are designed to help you study the game. *Change* lets you change any pieces on the board. This is useful for setting up hypothetical situations and studying how to play them.

*Expect* shows you what the computer thinks is your best move, based on the look-ahead analysis it has made on its previous turn. *Scores* shows you the relative strengths of all the moves open to you. This assessment, unlike the one in the *Expect* option, is not based on a long-range analysis, but on the immediate number of positions to be gained or lost by the moves. Experience will show that the move that turns the most pieces is not necessarily the best move to make. The player who seems in the lead in the middle of a game of *Othello* may be thoroughly trounced by the end. This is a lesson that *Odin* is particularly adept at teaching.

Move is an interesting command. It tells the computer to take over your position in the game and allows you to play its position. This is an act of desperation for the student. It can often teach that even a hopeless situation has some escape. It can also teach that you have no chance against *Odin* in the higher levels of play. Which brings us to another unique feature of *Odin*. It allows you to choose from among fifteen levels, and to change the level at any time during the game.

*Restore* will take back the last move, allowing you to go again. *Replay* shows you the whole game as it progressed from beginning to end. Both of these are useful as learning tools.

The care that went into the programming of *Odin* is matched by the care that went into the preparation of the manual. It is clean and well organized, with diagrams to help explain the fine points of the game. Each option is fully described in a logical sequence. An appendix discusses the history of the game and some observations on the strategy behind it. Experts can't seem to agree on the best strategy for *Othello*. This is probably because experts suffer from human frailties. Most of the observations on strategy in the manual seem to come from the way the computer plays the game. With its being able to look ahead as far as twelve moves into the future, the computer is as close to infallible as an *Othello* player can be.

That fact can make *Odin* a frustrating opponent. It is not recommended that you give up human opponents for *Odin*. We doubt that the human ego can survive that kind of abuse. What *Odin* can provide for the beginner and expert alike is a way of learning new strategies and keeping in practice for your favorite flesh-and-blood opponent. And when that match occurs, *Odin* has an option for supervising a two-player game as well. DD

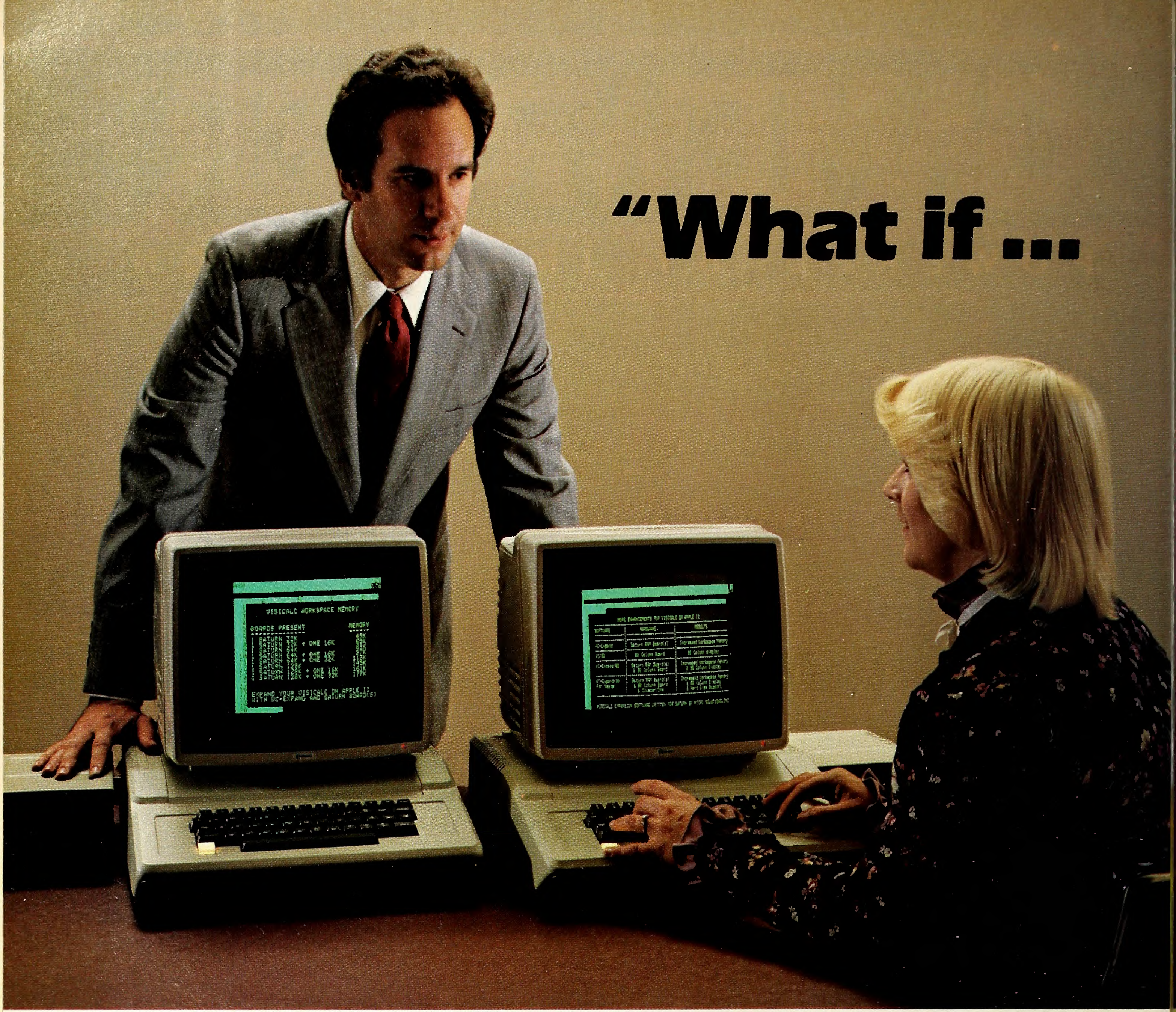
*Odin*, by Larry Atkin and Peter Frey, Odesta (930 Pitner, Evanston, IL 60202; 312-328-3211). $49.95.

**B & B Speaker.** By Bill Burwell and Bob Benjamin. Anyone in any doubt as to the generation of the name of this product or company may regress to the contest on page 2 for training.

Burwell and Benjamin have produced for the marketplace what occasional hobbyists have been doing for their own systems for a long time; and for those of us for whom the words *tweeter* and *solder* might well be synonymous, the product is welcome.

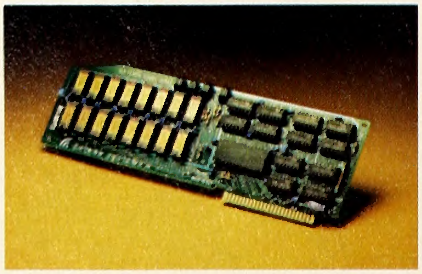
Housed in a modern eight-by-eight brown wood box, the B & B speaker plugs directly into your Apple and intercepts all sound before it makes it to the beeper.

Then, because the B & B has a magic dial, you can turn all your Ap-




# I run out of memory?"

Most people *do* run out of memory with only 18K VisiCalc* workspace. But you can expand your Apple II* to 177K VisiCalc memory! You can also get 80-column display, lower case letters, and hard disk support—all without buying a bigger computer.



The Saturn expansion system for VisiCalc consists of a preboot diskette, one or more plug-in RAM boards, and an optional 80-column display board. You can put the Saturn boards in any slot. And with all that memory, our software lets you save files on more than one diskette.

Each Saturn RAM board includes additional software for other programming applications. So your BASIC, PASCAL, and CP/M programs get an extra bonus.

Ask your computer dealer for more details about the Saturn memory expansion systems. See how much bigger and better your models can become!

*VisiCalc is a registered trademark of VisiCorp. Apple II is a registered trademark of Apple Computers, Inc.

SATURN SYSTEMS INC.

P.O. Box 8050
3990 Varsity Drive
Ann Arbor, MI 48107

1 (313) 973-8422

ple's sound off if you like and you can adjust the volume over a considerable range when you have it on.

That's it. It doesn't affect the quality of sound noticeably. But then its price doesn't affect your wallet particularly noticeably either.

All the B & B does is add a whole bunch of convenience.

If you play games at night and don't want to disturb the family (or don't want your husband to know how late you stayed up playing *Serpentine*); if you want to play *Microgammon* during lunch at your desk but don't want to be a bad influence on co-workers (or don't want them to insist on a turn); if you live under an airport runway and do want to hear Cavalier's latest game ditty—then B & B is the ticket. MCT

*B & B Speaker*, by Bill Burwell and Bob Benjamin, B & B Microproducts (14711 Lull Street, Van Nuys, CA 91405; 213-899-7332). $34.95.

**Spy's Demise.** By Alan Zeldin. There is no such thing as an Everyman video game. Rarely on the hi-res screen will you see glorified the life of Joe Normal. Your character in Penguin's latest entry in this larger-than-life medium is an adventurer, but he is also a loner. He is the secret agent, the international spy; unable to trust anyone, unable to confide in anyone. Not even the elevator boy.

How's that for a role model? In *Spy's Demise*, every move you make is at risk of bodily harm. This game requires daring and quick reflexes. As a nice change from the typical arcade format, it doesn't require you to kill anything. That doesn't mean that you won't get killed. You will. Frequently.

Your mission, should you decide to accept it, is to get to the top of a building while enemy spies attempt to crush you with elevators. This may seem a bit odd, but to get to the top of a building, you have to run your figure, a thoroughly suspicious-looking chap wearing a trench coat and dark glasses and stealthily carrying a briefcase, from one side of each floor to the other. At the end of each floor is a lift that takes you up a level.

Meanwhile, an evil cadre of even more Sam Spadeish types constantly ride a set of elevators up and down the building at varying speeds. They never leave the elevators, but if one happens to intersect your path as you run by, you vanish in a very smoky explosion (what was in that briefcase?). This traumatic event is accompanied by the opening bars of "Secret Agent Man."

The points you get for each level depend entirely on how quickly you can reach your goal. The seconds tick away on a digital display at the top of the screen, but don't let them pressure you. Try to go too fast and you won't make it at all. On top of that, there are flashing dots that appear for bonus points. Do you risk your neck to go back to get them, or do you just try to survive? Think fast; the meter is running.

As an added bonus, when you complete each of nine screens part of a coded message will appear. Penguin is offering a *Spy's Demise* T-shirt to the first person from each state to solve the cryptogram. This may actually take a while. Not that cryptograms are very difficult, but completing nine levels in *Spy's Demise* will take a James Bond of video games. DD

*Spy's Demise*, by Alan Zeldin, Penguin Software (830 Fourth Avenue, Geneva, IL 60134; 312-232-1984). $29.95.

**Horse Racing Classic.** By Dennis Tazumi. "I've got the horse right here, it's name is Paul Revere. . . ." Yes, guys and dolls. It's race time. The flag is up and "They're runnnnnnning!"

The horses with early foot have taken the lead, and the horses that are strong in the lane bide their time while saving ground along the rail. At least it certainly looks like J. O. Tobin and Ack Ack taking the lead, with Stymie, War Admiral, and Spectacular Bid closely bunched up tight, while Vigors and Exceller trail the field.

Around the first turn, Tobin and Ack Ack push each other for the lead, Stymie saves ground on the rail, while Spectacular Bid is forced wide. Vigors and Exceller trail.

Down the backstretch, J. O. Tobin and Ack Ack set furious fractions. Stymie closes along the rail, War Admiral goes wide to overhaul


Lovers
OR
S·T·R·A·N·G·E·R·S

the leaders, and Spectacular Bid regains the ground he lost. Vigors and Exceller trail.

Into the far turn, the pace is telling on the leaders, but J. O. Tobin continues to race neck and neck with Ack Ack. Stymie is looking for room along the rail, War Admiral is forced wide, and Spectacular Bid has moved up to challenge. Vigors and Exceller are starting their moves and are closing ground rapidly.

Into the stretch, they're four across the track. Stymie found room along the rail and has moved into first. Ack Ack is inside J. O. Tobin and has him by a head. War Admiral's in the middle of the track in full stride and Spectacular Bid is looking for an opening. Vigors and Exceller are closing fast.

Past the eighth pole, Stymie has the lead. War Admiral's moved into second. Ack Ack holds third, but the pace has gotten to J. O. Tobin, who's veered out. Spectacular Bid and Exceller make for the space left by J. O. Tobin, while Vigors goes way wide in an attempt to circle the field.

As they come to the finish, it's Stymie and War Admiral neck and neck. Exceller has Spectacular Bid by a nose as both gain ground. Vigors is in high gear on the outside as Ack Ack and J. O. Tobin drop back.

At the finish, Vigors gets up for the win. Spectacular Bid gets his nose to wire ahead of War Admiral as Stymie falters to fourth and Exceller is fifth. Ack Ack and J. O. Tobin trail.

*Horse Racing Classic* comes as near as anything yet to capturing the excitement of the racetrack on an Apple II. The folks from British Columbia have pulled off a programming tour de force to pack a disk this full. You get one hundred horses available to run on a nine-race card. There's an official track program that announces the entrants in the next race along with the weights they are carrying and the odds.

There's a racing form that includes data on the odds, jockey, trainer, number of times in the money, and charts for the last two races run by each horse. You're provided with one hundred dollars pocket money to bet as you will as the card progresses.

The races themselves are fully animated, although Bill Budge won't lose any sleep over the techniques used. And the results are automatically updated to the horses' files at the end of the race. There are a lot of niceties that make this program a pleasure.

But is it a good simulation of a day at the races? Absolutely not. The fault lies not so much with the programmers or their choices as with the 6502. The Apple II, with 48K of RAM and three or four screen refreshes a second, just can't do justice to something as complicated as a seven-horse race. On the other hand, imagine these Canadians loose with an Apple IV! They're three-fourths of the way to a superior program, given the speed and memory we're led to believe the Apple IV will have.

A cynical reviewer could have a lot of fun with the flaws. The horses round the curves sideways, thus appearing ready to bolt into the infield or through the fence back to the barn. The players do not have the option of controlling which horses will enter which races. The horses do run true to their electronic racing form, but any resemblance between that form and its real-life counterpart is purely coincidental.

These things take on less importance in the middle of a race, when you're urging Secretariat to make his move while the person next to you is pleading for Foolish Pleasure to hold on. *Horse Racing Classic* may be less than perfect, but punters everywhere will love it.

It's the real McCoy, and let me tell you, boys; you'll have a ton of fun backing Equipoise. Can do. Can do. On your Apple II you can do. ART

*Horse Racing Classic*, by Dennis Tazumi, Tazumi Software International (8 North Grosvenor Avenue, Burnaby, British Columbia, Canada V5B 1J2; 604-294-2891). $36.95.

**The Cosmic Balance.** By Paul Murray. Paul Murray's *Warp Factor* is among the best of the space battle games for the Apple. It's challenging and exciting, with a wide variety of spaceship types and scenarios to keep it interesting for either two players or one player against the computer. But despite its popularity, it has one glaring drawback that it shares with most games of its type: it is painfully slow.

This isn't too surprising, as a really good space battle simulation has to do a lot of complicated calculations just to monitor the game. Playing the game requires that the computer make moves in a logical way. Even a beginner could easily beat a computer that was just tossing around random numbers, so the computer as player takes even more time.

Now we have *The Cosmic Balance* by the same author, a game that is not only in the same genre but seems to use a lot of the same conventions as well. Is this Warp Factor II, The Wrath of Murray? The first scenario involves two starships fighting a battle to the death. Their names are—you guessed it—*Enterprise* and *Reliant*.

So why has Strategic Simulations released a game that is so similar to *The Warp Factor*? Well, *The Cosmic Balance* is a member of their new Rapid Fire line, and as that implies, it plays much faster than its predecessor. There is actually no time lapse between the moment the player finishes giving orders and the beginning of battle display. This is not the only change or improvement over the earlier game, but in view of how slow most space simulations are, it is surely the most impressive.

For those who are unfamiliar with this sort of game, it plays like this: certain victory conditions—most commonly, your survival and the enemy's demise—are assigned at the beginning of the game. You are given a ship or a fleet of ships with which to accomplish your objective. Here lies one of the differences of *The Cosmic Balance*. In this game you must actually design your ship before you start the game (except in the first scenario, which allows the newcomer to gain some experience as a commander before donning the role of designer).

Designing a ship of the complexity required would seem to be an extremely difficult task. You have only one resource to limit what you put into a ship: space (the final frontier). The manual lists all the things you might want to put into the ship—engines, drives, weapon systems, and so on—and how much space each occupies. All you have to do is select a component from the menu and the computer will tell you how much space is left. If you run out of space and still need to add something crucial, you'll find that it is as easy to remove a component as it is to add one. All the information is efficiently gathered on one screen, so you aren't required to remember what you have done already; it's all there in front of you.

The great thing about being able to design your own ships is that you can specialize them for a given purpose. In the Planetary Raid scenario, you must attack a planet that has very strong defenses. However, you will find that it is better defended against guided "seeker" missiles than against siege phasers. Solution: design a ship with no seekers and use the space to augment your power supply.

If you are playing against the computer, after designing your own ship you must also design your opponent's. If you think it will ruin the surprise to play against a ship you just designed, the manual suggests that you make several and select among them randomly. The recommended method of selection: rolling dice, of all things.

Now you're ready to begin. Movement is simultaneous. In the two-player game, player one enters orders secretly, then player two goes, and then the computer executes all the orders. In the solitaire version, the computer seems to be making its decisions at the same time as the player. Orders for charging shields, charging and firing weapons, and making course and speed changes can be given in any order. All the orders are displayed on one screen at all times (except when you request a hi-res tactical display, which is necessary for determining the relative positions of the enemy ships), so it is easy to see what you've done already and rescind any bad commands.

Like its predecessor, *The Cosmic Balance* has only a partial allegiance to the laws of physics. It takes as much energy to decelerate your ship as it does to accelerate, which is as it should be. However, it takes no energy at all to turn. All you have to do is point the ship's nose in another direction and all forward momentum is deflected—which is fine if you're an airplane, but the same rules should not apply in a frictionless environment. Admittedly this won't annoy most people, and it doesn't really detract from the fun of this game, but there are a few hard-core science-fiction fanatics out there who are dying to see somebody get it right.

*The Cosmic Balance*, by Paul Murray, Strategic Simulations (465 Fairchild Drive, Suite 108, Mountain View, CA 94043; 415-964-1353). $39.95. DD

**Money Munchers.** By Bob Bishop. Maze games are the Chinese food of computer arcade games. Even before the advent of that pie chart refugee, *Pac-Man*, maze games were common microcomputer fare. And there have been some memorable meals, so to speak, but more often than not one is hungry again in an hour.

Yet maze games are something of a constant in any computer gamer's diet. And a pleasurable one, even if only in a fleeting way. Most publishers serve up some maze game variant, but the biggest eater in town these days seems to be DataMost. And they got Master Chef Bob Bishop to prepare their latest offering.

The name says it rather succinctly. Each maze in *Money Munchers* is carpeted with dollars. You are the kid in the candy shop. Actually, you're a small fellow who seems to swim through the mazes, raking in the dough just as fast as he can. There are the Money Munchers, of course, who are also intent on reaping the pecuniary harvest. They wander about scarfing up money whenever they chance upon it, paying little heed to anything else in the maze. Should one latch onto you instead of a greenback—well, bye-bye. When the loot is all gone, colorful kaleidoscopic patterns entertain you while the next maze is readied. If you survive a level, you gain an extra life.

The next two mazes also contain spiders—lethal ones, naturally. But they do not pursue you either; they merely patrol patches of money. You could simply avoid them, but it would be such a shame to leave all that nice cash to the ambling Munchers.

By the time you reach the fourth level, a truly inimical presence is felt. There are these snakes, and they hunt you. No matter where you are, the closest serpent will home in on you, and the others all begin to wriggle toward you. They, too, can be avoided, but only with a great deal of effort and persistence. Unfortunately, the Money Munchers and the spiders are still there, making it quite crowded.

And just in case you get any silly notions about figuring out the patterns to run in the mazes, chef Bishop has prepared a true surprise: You never see the same maze twice. For these are random mazes, thousands of them!

Like most good maze games, *Money Munchers* is quite addictive. While it does not have the distinctiveness of some recent variations on the eternal formula, it is a lot of fun. The denizens of the mazes are clearly identifiable and quite nicely animated, while the money carpet changes



only in area, never in design. Green is a constant throughout the game.

DataMost is becoming quite reliable for this particular genre. *Money Munchers* is the sixth game they've released that is played in mazes. It has its own identity and easily stands apart from the others. It is a lot of fun to play.

But, as with most maze games, one is left feeling hungry after an hour or so. DA

*Money Munchers*, by Bob Bishop, DataMost (9748 Cozycroft Avenue, Chatsworth, CA 91311; 213-709-1202). $29.95.

**Watson.** By Bill Sefton and Mark Pump. As Holmes had his trusted assistant, so does the *Inspector* from Omega Microware have its companion. *Watson* is a sophisticated disk sleuthing utility that requires the *Inspector* to be resident in memory to work. More than just an add-on, *Watson* actually piggybacks the *Inspector*. The program is unique because it is available on either a plug-in chip (replacing the current D8 EPROM) or, for those who own language cards, on a disk. There are twelve new functions that *Watson* will do.

One of the most important features is the ability to reconstruct VTOC for a blown disk. This allows the VTOC from the bad disk to be read, and then *Watson* rebuilds the track bit map from the catalog track. Another function provides for hex ASCII disassembling, which is a great aid for programmers. All track and sector lists can be displayed, thus aiding the reconstruction of a blown catalog. Sectors can be locked in or freed and extra space cleared by killing DOS on tracks 1 and 2.

If you have ever made a copy of an important file disk and wondered, in the back of your mind, whether all of it copied, then the disk verification function will be much appreciated. It compares the sector checksums of each disk to make sure they are the same. Another very important utility for programmers is the *File Follower*. This program greatly aids in debugging your own programs. It allows a file to be scanned sector by sector automatically. It is this bloodhound talent that really earns this package its title. It's certainly not for the average user, but geared for the sophisticated progrmmer who needs such utilities. RRA

*Watson*, by Bill Sefton and Mark Pump, Omega Microware (222 South Riverside Plaza, Chicago, IL 60606; 312-648-4844). $49.95.

**Dnieper River Line.** By Bruce A. Ketchledge and Gary Sipes. Avalon Hill has for many, many years been the premier war game company in the United States. Many people's first contact with strategic games was over an Avalon Hill board game like *Tactics II*, *Gettysburg*, or *D-Day*. To date, their efforts in the computer field have been directed primarily at the TRS-80 cassette market, with a few programs translated for the Apple. Now, however, the company is directly addressing the Apple market also.

*Dnieper River Line* is certainly Avalon Hill's best effort yet to translate their vast board experience onto the computer. The lack of graphics is partially compensated by the enclosed playing map. The 8 x 11 map is broken into a 12 x 12 grid. Accompanying the map are 352 standard cardboard counters. Each counter represents a different division.

The scenario of the game is the Soviet crossing of the Dnieper River in late 1943. The computer plays the Russians and you play the German officer commanding the Army Group South. There are six objective areas on the map. At the start, the computer will secretly select three of these areas as objectives. If the computer achieves those objectives, it wins. The German commander is given only fifteen initial units to deploy along the river, but he can lay down mine fields and deploy garrison troops in the towns. The Russian side has a random mix of twenty-two possible units deployed secretly across the river.

The game moves very fast. The German commander is always straining to catch glimpses of the enemy forces and to determine the Russians' objectives. The computer, with words alone, does a credible job of showing garbled communications and disjointed visualization as the Russians lay down heavy artillery and smoke barrages. The terrain makes rapid deployment of reserve forces difficult, and, seemingly at the worst possible moment, one of your key divisions may not be able to move due to mechanical breakdowns. Erratic off-map artillery can provide miraculous support or be unavailable for hours.

Mastery of this game will take quite a while, as it is very difficult for the German side even to get a draw. *Dnieper River Line* is not for the beginning strategy player. It is more like a tough, middle-game, chess problem. If Avalon Hill ever develops the ability to achieve complex color graphics on the Apple, they would certainly become one of the most formidable forces in microcomputer games. RRA

*Dnieper River Line*, by Bruce A. Ketchledge and Gary Sipes, Microcomputer Games, a division of the Avalon Hill Game Company (4517 Harford Road, Baltimore, MD 21214; 301-254-5300). $30.

**Ali Baba and the Forty Thieves.** By Stuart Smith. One of the famous stories from the Arabian nights is retold in delightful hi-res color as a role-playing fantasy game. You are the famous Ali Baba and the sultan has asked you to rescue his daughter Princess Buddir Al-Buddoor, who has been kidnapped. Ali Baba can call on a variety of allies to assist him in overcoming ferocious tigers (great image), bandits, magical swords, and a large variety of monsters—more than one hundred at last count. Each ally and monster has its own strength allotment, weapons, and armor. There is even a deadly dragon with incredible statistics. If this does not sound hard enough, the monsters have a reincarnation ratio, which, luckily, you can adjust for difficulty.

Trading posts exist where better weapons and armor can be bought. Until he can equip himself better, it is very hard for Ali Baba to survive. The game is divided into dungeons and outside areas. Each section has a different feel to it. Gold can be found throughout both sections, in chests. Unfortunately, some of the chests are booby-trapped, and most are guarded by monsters. The trapped chests inflict damage to the person opening the chest. When a character's strength goes down to zero, he dies. Resurrection is occasionally possible, but it means starting over at Ali Baba's house.

One of the finest parts of the game is its music. The computer plays *Scheherazade* very well, at many different spots in the program. One of these places is when you find a mystical rune. These runes (to musical accompaniment) convey special secret messages that can be of great aid in playing the game. However, these runes also have a nasty habit of exploding after they are read.

On your wanderings you might free two other captives, Abdalla and Morgiana, who are so grateful that they follow you and assist you in your quest. Beware of moving statues and collapsing tunnels. If you are very virtuous, a unicorn may aid you in fighting the monsters. Things are not what they seem in many parts of the dungeon, so explore very cautiously.

The author of this program has included a private challenge to each person playing it. The player must use only Ali Baba (no allies) and not strike out at a single monster, even in defense. Then when he rescues the princess and returns her to the sultan, a special message comes on the screen with the author's personal congratulations on achieving a nonviolent solution to the game.

Since the game plays very easily, children can enjoy this as an introductory fantasy role-playing game. In fact, whole groups can play at once, each person being a different ally of Ali Baba and aiding him on his quest. This very different hi-res adventure should be part of every game player's collection. RRA

*Ali Baba and the Forty Thieves*, by Stuart Smith, Quality Software (6660 Reseda Boulevard, Suite 105, Reseda, CA 91335; 213-344-6599). $32.95.

**Battlesight.** By Major C. Hanselmann, D. Lubar, V. Bauman, J. Anderson, and E. Wolcott. Finally a serious tank-to-tank warfare game has been developed. It's the first day of World War III. The Warsaw Pact forces have crossed the West German border in full force. Your job is to try to stop them, or delay them as long as possible, with your force of five M60A3 tanks dug in on the high ground overlooking the Russian avenue of approach. Your intelligence team delivers a profile of the opposing forces. They consist mainly of the Soviet T-62 tanks, which are medium tanks but hold the world's record for firing the fastest tank rounds. There are also, scattered throughout the attack force, the heavier armored T-64 and T-72 tanks which are more accurate at greater distances.

The Soviets have also deployed some of their Hind-A missile-launching helicopters, which are very accurate at long range. The enemy is using low-flying aircraft to try to spot out your position, so that their forces can zero in on your tank force. Fortunately, the Soviet forces are approaching in a spearhead formation; thus each wave consists of slightly more tanks than the previous wave. The game is over if all five members of your tank force was destroyed, or if you let five of the enemy tanks slip by.

The movement of the tank and its operations are what makes this game unique. Tremendous attention to detail has been incorporated into *Battlesight*. Both the turret and the treads are movable. Taken into consideration are the speed of the individual tanks on both sides, the time to reload, the speed of the tank rounds and the missiles in flight, the turret traversal speed, the rate of fire, the kill probability at varying distances, and the firing accuracy while stationary or moving. Far shots have a lower probability of hitting than close ones, but if you wait too long the reload time and the turning speed of your turret will delay your shooting another enemy, enabling him to escape.

Designed for the TG joystick or its equivalent, *Battlesight* gives a very realistic feel to the handling of a tank under battle conditions. The low-flying observation planes are shot down with the tank's machine gun, fired by hitting the space bar. Your tank can also suffer varying degrees of damage before it is fully destroyed. The turret can get damaged, or the treads blown off, immobilizing the tank. You even have the option to repair the tank, although you are defenseless during that time. Surviving successive waves with the same tank can increase the efficiency rating of its crew, thus bettering their odds on hitting a far target and shortening the reload time.

When you feel you have become really proficient at this game, then you can add the option of night warfare for a really fantastic challenge. Every other wave makes its attack under cover of darkness. The only time you can get a quick glimpse of the advancing force is from the flash of their muzzles when they fire, or from a quick magnesium flare. The player has to develop a keen sense of distance and rate of tank speed in order to fire accurately under these conditions.

*Battlesight* even provides a lesson on strategy for the player. It is appropriately called "Helpful Hints for Hopeful Heroes." At the end of the game a detailed profile is given showing tanks destroyed, shots fired, crew efficiency, and overall tank performance.

This is Versa's first action game, and it is certainly hoped that this fine team will produce other high-quality games. *Battlesight* is a superb modern tank warfare simulation. RRA

*Battlesight*, by Major C. Hanselmann, D. Lubar, V. Bauman, J. Anderson, and E. Wolcott, Versa Computing (3541 Old Conejo Road, Suite 104, Newbury Park, CA 91320; 805-498-1956). $39.95.

**Warp Destroyer.** By Thomas Ball and Eric Varsanyi. It's nice how some games naturally go together. *Zorks I* and *II*, *Wizardry* and *Knight of Diamonds*, and the *S.A.G.A. Series*, for instance. And after a few rounds of *Warp Destroyer*, you'll feel like playing another Piccadilly game: *Suicide*, for instance.

You'll be having a ball shooting up probots and enemy craft while producing enough noise to make punk rock sound like Mantovani. Then it will hit you: "What's the point of all this, anyway?"

*Warp Destroyer* features some fine graphics, luscious and noisy sound effects, and a story line that justifies all the seemingly pointless shooting and bombing that goes on. According to the instructions, it is the year 3526, and your lifetime enemies, the Zalbians, have broken the Neutrality Pact of 3103 (after four hundred years, isn't it time for a new pact?). Your mission is to recapture your twelve planets from these imperialist expansionists.

You must go through five phases to rescue each planet. The first requires you to keep a small cross lined up in your crosshairs while you go through subspace. If you let it drift too far, you have to start over. The next three phases involve shooting at mines, fighters, and probes, before finally making your assault on the Zalbian base. Positioning the crosshairs is a breeze with a joystick; don't even attempt to use paddles or the keyboard unless you're a glutton for frustration.

The hard part is making a hit. *Warp* doesn't afford you the luxury of light-speed laser blasts or ion cannons. Instead, you get missiles that look more like diminishing rectangles than anything else as they fade into the distance and take what seems like forever to find their targets. Firing them is fun, though. You are allowed to fire several shots at a time, so many that the screen will fill with flying rectangles obstructing your view of the targets.

And obstruct they will, because the only way to hit anything is to fire

**More Apple II owners choose Hayes Micromodem II than any other modem in the world.** Compare these features before you buy. You should. It's your money. Thousands of other Apple II owners have already compared, considered, and are now communicating — all over the U.S.A. — with Micromodem II. The best modem for the Apple II. The most modem for your money.


**A complete data communication system.** Micromodem II is not "base priced" plus necessary "options." It's a complete, high-performance data communication system. The printed circuit board fits — quickly and easily — into your Apple II, eliminating the need for a serial interface card. And the Microcoupler™ (included) connects the Apple II directly to a standard modular telephone jack. Auto-dial and -answer features are built in. Operation can be full or half duplex, with a transmission rate of 300 bps. And it's Bell 103 compatible and FCC approved.

**Now there's Hayes Terminal Program, too!** Developed by Hayes specifically for Micromodem II, this new Terminal Program allows you to access all the great features of your modem in a matter of seconds.

With it, you can use your CP/M,® DOS 3.3 or Pascal formatted disks to create, send, receive, list and delete files. Hayes Terminal Program is a complete, stand-alone disk.

And because it's menu driven, you can choose from a wide variety of options to set your communication parameters — as well as change hardware configuration — directly from the keyboard. It even allows you to generate ASCII characters that are normally not available from Apple keyboards, further extending your capabilities. Incoming data can be printed (on serial or parallel printers) as it's displayed on your screen.

**Micromodem II is available with or without the Terminal Program.** Buy your modem by itself; or optionally packaged with the Terminal Program disk and user manual at extra cost. The software is also sold separately, for those who already own a Micromodem II.

**If you're ready to communicate with other computers,** to access information utilities, time-sharing systems, or use bulletin boards, then you're ready for Micromodem II. Come on. Compare. Consider. Then buy.

Micromodem II is already the best-selling modem for the Apple II. And Hayes' new Terminal Program makes it better than ever. Available at computer stores nationwide.

Hayes

**Hayes Microcomputer Products, Inc.**
5835 Peachtree Corners East
Norcross, Georgia 30092 (404) 449-8791

like crazy and hope that your missiles happen to explode as your target drifts by them. If you're good at predicting where randomly moving objects will be a half second later, you'll do fine.

The attack on the Zalbian base is faintly amusing at best. After firing through gaps in the base's moving shield and spreading several shots evenly across the base, you must hit the return key to detonate them. Boom! Now what? Those crazy Zalbians seem to go on living as though nothing had happened. Boom! Boom again! Still nothing. Unfortunately, the instructions don't give the slightest hint on how "evenly distributed" your shots should be to destroy the base successfully. Maybe if you keep shooting, they won't have time to eat and you can starve them to death.

There are twelve levels, one for each planet you rescue, each more difficult than the last. If you like good graphics and strong sound effects, if you like shooting at things and have a tireless button-pushing finger, then *Warp Destroyer* is for you. If you enjoy the feeling of achievement and success for valiant efforts, look further. MTV

*Warp Destroyer*, by Thomas Ball and Eric Varsanyi, Piccadilly Software (89 Summit Avenue, Summit, NJ 07901; 201-277-1020). $29.95.

**Early Elementary I. By Steven Shotwell.** Lo and behold: a teaching program in which the manual starts out by specifying exactly what the program intends to teach! This program is aimed at students in preschool through second grade, and will teach them to recognize and name fifteen colors, the numbers from one through twelve, and seven geometric shapes. As children work with the program, their eye-hand coordination will improve as well.

Yes, this program is set up so that a child can run it alone. Once the system has been started, all the child has to do is press any key to indicate the answer.

The basic format is this: a geometric shape (for example) is displayed on the upper half of the screen while a series of different shapes is displayed, one after another, on the lower half. When the matching shape appears, the child may press any key to indicate the match.

Correct answers are rewarded with a short musical phrase from the speaker and a message on the screen like "Great, John!" or "Super, Marsha!" (assuming the student typed in one of those names when prompted at the beginning) followed by a smiling face that winks. Wrong answers get a frowning or sorrowful face and no music.

The program offers four different kinds of drills: Shape Match (described above); Color Match, which works similarly with blocks of different colors; and two forms of number behavior—Number Drill and Count the Shapes. In Number Drill, a number word ("three" or "seven" or the like) in *large* characters fills the top half of the screen, and a sequence of numerals, also *large*, flashes by on the lower half for matching. In Count the Shapes, a group of of objects (squares, circles, and the like) appears in the top half, and a sequence of numerals is displayed below: the child must pick the numeral that matches the quantity of shapes displayed.

In all of these modes, the system is always "busy": something is happening all the time. This, plus the simple but effective animated graphics and the music, make it seem more like a game than a teaching device. Clearly the intent was to make the system fun for the kids to play with—if you don't tell them it's good for them, they'll probably love it!

The system was obviously designed by a teacher with an orderly mind. It not only scores each run, it automatically records the score in a class record file under the student's name. Class records are only accessible through the teacher management file, which is locked by a password.

Other options available through the teacher management file include turning the music on and off, changing the presentation speed of the answer choices, changing the number of questions in a drill, and the like.

This is a very limited program in certain ways: it has a limited audience and limited objectives. But that's not necessarily a disadvantage; tight limits permit a very sharp focus. The author knew exactly what he wanted to do, and he seems to have done it very well. If you're trying to teach these behaviors to a child in this age group, then this program could save you a lot of time! JR

*Early Elementary I*, by Steven Shotwell, Compu-Tations (Box 502, Troy, MI 48099). $29.95.

**Voyager I.** By William Volk. *Voyager I* represents Avalon Hill's first effort in the realm of computer graphics. Do not be put off by the picture on the box, which features the much cruder TRS-80 style graphics. The Apple version features the 3-D graphics perspective that *Wizardry* has made famous.

The game revolves around your attempt to single-handedly destroy an alien spaceship from the inside. The *Voyager I* was built by a world of robotic warriors bent on galactic conquest. You have fought your way aboard and are trying either to capture the ship by killing all the robots or, failing that, to blow the ship up. The game is a combination arcade game and role-playing game in real time. There are four levels to the ship, each containing thirty-six rooms.

A rapidly discharging laser rifle is your only weapon. Luckily, there are a few fully charged laser rifles lying around the ship waiting to be picked up. The screen shows the 3-D view of the ship's interior and three status thermometers, graduated from 0 to 100. One reflects your strength, another the charge left in your laser, and the third the percentage of generator power left to run the ship. Strength points are lost with movement and as a result of battle with the robots. If your strength goes to zero while you're fighting, you're dead.

The game is self-mapping. Areas that you've explored remain on the map until an entire level is revealed or until you change levels. When you go up or down the elevators to other levels, any part already explored is shown to you. Unfortunately, *Voyager I* suffers from a severe playing problem—it's impossible to lose! Every time you're killed, not only are you resurrected but you retain all your previous mapping. Any robots or generators you destroyed before your death remain destroyed. For those who prefer perseverance to skill development, it is always possible to win. Also, a special non-prize goes to the first person who can explain how a single rifle, held by one person, can fire, from the bottom corners of the screen, twin shots that converge on the center!

Despite these problems, the game is fast-paced and enjoyable. The robots shoot fast and are hard to kill. A late suspense factor comes when you destroy the last generator. You have sixty seconds to reach a hidden shuttle craft and escape before the ship explodes. For a first effort in color graphics, Avalon Hill has done well. RRA

*Voyager I*, by William Volk, Microcomputer Games/Avalon Hill (4517 Harford Road, Baltimore, MD 21214; 301-254-5300). $25.

**Oil Rig.** By Kevin Bagley. J. R. Ewing may find that he has competition. Now anyone can be as loathsome and greedy as he is.

No matter how much they may curse OPEC and Exxon when prices go up at the pump in the real world, players of *Oil Rig* will pray for shortages when all their assets are in petroleum products and the government is regulating them from here to Sunday. You laugh now, but when you are itching to sell your first-born child to pay for that drilling platform before the price goes up another ten thousand dollars, and you suddenly realize that your last shred of humanity has long since disappeared, don't say we didn't warn you!

The game is played on what looks like a real-time *VisiCalc* spreadsheet. The display shows constantly fluctuating prices on six commodities from crude to refined oil and eight kinds of equipment necessary in the oil business. The player starts armed only with one thousand dollars and his own business acumen. That amount of money is just exactly enough to prospect for oil once, but it won't begin to cover the expense of drilling for any oil that is discovered. So where is one supposed to get the kind of money necessary for the big time? Even J. R.'s senile old grandmother could tell you that: buy low, sell high.

For this reason it is best to start out dealing in commodities, which begin well within the lower price ranges (ten to one hundred dollars per unit) and fluctuate from there to as high as one thousand dollars a unit based on such capricious events as OPEC price hikes (or reductions), government regulations (or "recommendations"), and the old standby, inflation.

Each item on the display has a letter from A to N assigned to it, so business transactions are as easy as typing B F 10, which means buy (B) ten units of petroleum products (F). The display then changes to show how many of the item you now own and the current value of your holdings, constantly updated based on fluctuating prices. When you have enough cash on hand to cover the costs of drilling, as well as the losses that could be incurred by a dry well, the high-stakes wheeling and dealing can begin.

Drilling is handled on the hi-res screen using paddles. A black rectangle moves across the screen, underground. You control the position of a drilling rig on the surface. When you push the button, a black line moves down from the drill. If it intersects the moving oil field, you get some income. While this may not be too realistic a simulation of the process of drilling for oil, it is welcome as an occasional change of pace from the market activity that takes up most of the game time.

The game takes some time to grow on you. In the first few rounds of play you will lose for reasons you don't understand. Having totally realistic expectations of the way that commodities, oil wells, the Arabs, and the government behave in the real world won't be an immediate help here. Things go by their own rules in *Oil Rig*. For instance, inflation can make your prices drop (what?). Or an even better example: a refinery that cost you less than a thousand dollars (really) can blow up even before you start to use it, costing you $15,000. Ha, ha; you lose.

Taxation is based not on your income, but on your cash on hand. If you are holding more than a hundred thousand dollars, you are liable to be taxed right into bankruptcy without warning. Ha, ha; you lose again. Need a tax dodge? Sink a few hundred thousand into a refinery. Wait a minute, didn't a refinery cost only one thousand dollars last year? Sho' 'nuff. It's a jungle out there, Bobby.

*Oil Rig* is surely not the most accurate attempt at a simulation we have seen, but as we said, it can grow on you. Learn to play by Bagley's rules instead of what common sense tells you is right, and you may have some fun. DD

*Oil Rig*, by Kevin Bagley, Computer Programs Unlimited (9710 24th Avenue S.E., Everett, WA 98204; 206-337-5888). $29.95.

**The Linguist.** By Robert C. Clardy and Charles J. Fleishman. This is not a program for everybody. If you're not working with a foreign language—learning, teaching, writing, translating, or whatever—then *The Linguist* probably will not help you. But if you are, and especially if you have to deal with a language that uses a special typeface such as Russian, Hebrew, or Japanese, this might be just the program you need.

*The Linguist* is a highly specialized file manager: it creates, edits, and (if desired) drills you on foreign language dictionaries.

So what's new and exciting about that? Anybody who's into computers has seen file management programs before. Right; but how many file managers have you seen that come with six different typefaces built in? This one not only manages files but also displays the words of each language in the correct character set for that language, both in upper and lower case. It also includes all the characters used by the International Phonetic Alphabet, the Trager-Smith phonemes, and other commonly used phonetic symbols.

Typefaces and special characters required are provided for twenty different languages: all the familiar European ones, the others mentioned earlier, and even Latin and Esperanto.

Or, if you're into Tengwar, or Sanskrit, or High Martian, that's all right too: the system will accept additional typefaces designed by the user (though you will need *Higher Text*, another program from the same publisher, to design them with).

When you're working in a foreign typeface—for example, adding words to your Greek/English file—the Apple's keyboard is automatically modified by the program into a "foreign language typewriter": for Greek, the A key prints the character Alpha, B prints Beta, and so on. ("Prints" refers to a display on the screen, of course, not the printer; you'd need a dot-matrix printer with a special graphics driver to handle this output.) The manual provides several pages of charts showing the various keyboard layouts.

The system offers three modes of operation, with the same group of activities in each mode. The translator mode matches a word or short phrase (up to fifteen characters) in one language with a word or short phrase in another; the definer mode matches a word in one language with a definition (a phrase up to forty characters long) in another; and the phrasebook mode matches a phrase (forty characters) in one language with the equivalent phrase in another.

In each mode, you have the following options: you can sort the list of pairs alphabetically, in either language, and then scan the list; you can "look up" a word or definition—input one half of a pair and get the other half (English to foreign or foreign to English); you can add to or edit the dictionary; or you can have the system drill you either on the entire dictionary or on a selected subset of it.

In a drill, the system will display one half of a pair, randomly chosen from the specified list, and wait for you to supply the other half. You can get a hint—the first letter of the missing word—by typing a question mark. The system displays a running tally of your answers.

Note: word lists and dictionaries are not provided with the system except for a few short demonstration lists. None of the above functions (except "add more words") will operate until you input a dictionary. And each mode requires its own list; you can't get a definition from a list of word pairs, for example.

There is also a slight problem with the manual: it is not as clear, or as easy to read, as one might wish. The necessary information is all there, but it is organized to explain what the system does instead of how to use the system. However, this will only trouble you in the beginning, if at all; once you're familiar with the system, it runs very smoothly.

The bad news is that you will have to create all your own data files, like dictionaries and such. The good news is that here is a tool you can do it with, no matter what typeface you need to work in. JR

*The Linguist*, by Robert C. Clardy and Charles J. Fleishman, Synergistic Software (5221 120th Avenue S.E., Bellevue, WA 98006; 206-226-3216). $40.

**Magic Spells.** By Leslie M. Grimm. *Magic Spells* is a delightful children's program. It's from the same team that currently creates the Learning Company games that are inspiring much excitement in the educational world. The programming and design work is done by Leslie Grimm, but all the hi-res and lo-res graphics are done by the very talented Corinne Grimm, age ten.

This is a fun educational program for children of elementary school age and above. Merlapple, the Wizard of Spells, welcomes you to the Kingdom of Magic Spells. Inside the castle are many chests of gold and jewels. Each chest can only be opened if you know the correct spell. These spells are anagrams, words that students must unscramble. There are eleven different lists, each twenty words long. If students give the correct spell, they receive all the treasure in the chest; if they miss, then the lurking Spelling Demon takes a share. The more tries required to get the correct answer, the more the Demon takes.

The game has a good self-help feature: every correct letter in the correct place is acknowledged. So if the scrambled word is *isucrc*, and the first guess is *scrius*, the program will show your answer as "??r?us." The question marks indicate which letters are still wrong. (Got the answer yet?) Delightful graphics and musical reinforcement make this a game that children come back to time after time.

*Magic Spells* is not a program that a child will master and then be bored by. It's designed to grow with the child. There is a separate segment of the program, called *Spells Writer*, that enables parents and teachers to program new word lists. Each list is twenty words long and can contain words up to fifteen letters long—so even high school students striving to increase their vocabulary can use the programs.

*Magic Spells* would be appropriate in every family library and should be seriously considered for any school using Apples. RRA

*Magic Spells*, by Leslie M. Grimm, Special Delivery Software/Apple Computer, (20525 Mariani Avenue, Cupertino, CA 95014; 408-996-1010). $45.

**Wiz-Mate Megaman.** By Louis Simmons and Dick Hodgkins. **Wizmaker Wizardry Character Editor.** By Ron Richards. *Wizardry* is a huge game; it takes a long time to develop characters strong enough to take on to *Knight of Diamonds*, and even longer to build ones who'll survive in the new environs.

But building characters is 90 percent of the game in *Wizardry*. Few things in gaming are as gratifying as seeing your *Wizardry* characters gain levels with the attendant improvement in attributes and mastery of new spells. One of the activities that keeps *Wizardry* players glued to the chair for one more, and just one more, expedition into the maze is the realization that a character is within a successful expedition's booty of earning a new level.

Conversely, it takes guts and confidence to change a carefully developed character's class from mage to samurai, from priest to bishop, from fighter to lord. Yet you know the character deserves the higher class with its greater long-range capacities. But making the change means casting your character back to level one and lowering his attributes to puny proportions. Only his hit points and knowledge of some spells are retained. But you do it, and you begin building him again. You know it was worthwhile when your former fighter begins to master spells, when your former mage adds priest spells to her repertoire, and when your new bishop can identify the amazing magical items Boltec would have charged hundreds of thousands of gold pieces to inspect.

All this is bypassed with the *Wizardry* oriented utilities, or can be.

Then what's the point of playing, you may ask. Well, each *Wizardry* scenario has a plot and puzzles that must be mastered to win the scenario. But playing *Wizardry* for those alone is to make a great, full game a mere fantasy game; it isn't even role-playing anymore, because the roles are divorced from the player—are invincible characters produced by megamachines.

Appropriately, then, the first *Wizardry* oriented utility to hit the market is called *Wiz-Mate Megaman*. With it you can make a full roster's worth of supercharacters: all will be level 768, know most and sometimes all spells with maximum casting ability, have between seven hundred and twelve hundred hit points, and be equipped with such goodies as lords' garb, rods of flame, and healing rings (which make curing spells obsolete). Visiting Adventurers' Inn with these monstrosities becomes a necessary bore to refresh spells. The number of experience points needed for anyone to rise another level is so enormous that the computer must delay several seconds to announce it; none requires less than hundreds of millions, most require billions.

The Megamen are various and are the combinations chosen by the people at Wiz-Mate. There are good, neutral, and evil; all classes. There's even a neutral ninja. And they're named, but of course you can change that. That's all you can change. Nothing else you do with the characters makes a dent in their attributes or abilities. They're dead hulks, machines for mapping, weapons for killing monsters for the satisfaction of saying you got to the end.

But if you got to the end using Megamen, that's all you did. You certainly didn't win.

*Megaman* has one excellent utility. It allows you to print out your roster and each of your characters with all attributes, possessions, and spells.

*Wizmaker* is a different story. *Wizmaker* allows you to alter already made characters in any number of ways. You can make megacharacters with it if you choose; the one thing it won't give them that *Megaman* will is possessions. You must still earn those.

But *Wizmaker* will also allow you to do as little as you like. So you can restore the level that the lifestealers sucked away, for example. It's still cheating, but not quite as game-destructive.

To those not of the cheating bent, *Wizmaker* can be a palatable utility. *Wizardry* itself allows you to back up your characters, but all must return to back-up status if one is to be restored. With *Wizmaker*, you can restore just the character who needs it. You can take characters to more than one scenario simultaneously—lending your youngster your main party's leader to help her get her band started, for example—without removing the character from your disk.

And, if you're a person who keeps regenerating a character at the beginning until you randomly get seventeen or more points to distribute instead of the normal seven to ten, you can settle for seven and doctor the resulting character to reflect the extra ten points you otherwise would have spent five or ten minutes rolling for. (If you're like this reviewer, though, you'll keep rolling; this still feels like cheating.)

Neither of these utilities' publishers has any connection with Sir-tech and the real *Wizardry*. The Sir-techians believe, as we do, that using instantly created superheroes to play causes much of the value of their games to be overlooked.

Also, people out to make an independent profit on someone else's success are precariously balancing on a contradiction in terms. MCT

*Wiz-Mate Megaman*, by Louis Simmons and Dick Hodgkins, Computer Configurations (Box 1711, Austin, TX 78767; 512-472-3716). $34.95. *Wizmaker Wizardry Character Editor*, by Ron Richards, ARS Publications (3710 Pacific Avenue, Suite 16, Venice, CA 90291; 213-396-9303). $19.95.

**Lazer Maze.** By James D. Spain. If you ever wake up one morning and find yourself about to fight for your life in an alien arena, then you will be glad you had this training. *Lazer Maze* is a very different game and one of the most intriguing ones to come from the prolific Avant-Garde stable. The game challenges your mental agility and quickness of thought, rather than the blinding reflexes that are required for playing most arcade games.

The arena is filled with varying amounts of laser baffles, which reflect an incoming laser beam at right angles. Around the edge of the arena are numbers indicating position. You stand on one number and calculate the number of the point of exit for your laser beam as it ricochets around the arena. When you have determined the exit point for the beam, you enter this number. The computer then places your alien opponent on the exit number that you have indicated, and your laser gun is fired. If the beam does indeed exit where you have forecast, then the alien is destroyed and you are awarded points. The points earned are determined by the complexity of the path and the time it took you to enter the destination number. The complexity factor refers to the number of reflections within the maze, such as a four-bounce shot or a thirty-five bounce shot. For the more intricate shots, large bonuses are awarded.

This is all well and good, but what happens when you miss the alien? Well, then the alien gets to lob a nuclear grenade over the maze at you. And the alien never misses! So it certainly pays to be accurate, even more than it does to figure the exit point quickly. There is also a large bonus given at the end of each round if you have had no misses.

The rounds are divided into skill levels, starting with novice and working up to master level. The novice level has only twenty baffles in the arena, while the master level has seventy. Unfortunately, the game does not have a cumulative scoring system or a limited amount of men on your side. Basically, you are playing each round against a previous high score while trying to exceed it. This is not easy, as the mazes generated are random, so the highest possible score for each maze is quite different and not necessarily higher than any of the previous mazes. Once the player reaches the master level, he must keep playing seventy mirror mazes one after another in the hope that one will come along that will allow higher scoring combinations. RRA

*Lazer Maze*, by James D. Spain, Avant-Garde Creations (Box 30160, Eugene, OR 97403; 503-345-3043). $29.95.

**Math Strategy** and **Spelling Strategy**. By Robert B. Dilts. These two unusual educational programs, developed by Behavioral Engineering (Scotts Valley, CA), are built on principles that have been discovered in the field of neurolinguistic programming. Studies have correlated sensory representation and the way people organize their experiences; these "sensory cuing mechanisms" relate eye movement to learning and remembering. The theory underlying this discovery asserts that you remember best when you can picture what you're trying to recall in your "mind's eye." This is purported to be a spot up and to the right or left of normal forward eyesight.

In developing this training, computers are placed in front of students with the monitors placed in raised positions either to the left for right-handed students or to the right for left-handed students. Students are asked to look up at the monitor and visualize the questions, then either to repeat back the problem, to solve the problem, or to enter the problem backward, all while not looking at the original problem. To aid students, eyes on the screen are looking up in the direction the students need to focus. Also, students can select one of eight colors and all the correct answers will be shown in that color, reinforcing the technique. Two levels of expertise are provided, and additional lessons can be created by the parent or teacher.

There appears to be disagreement among educators about the value of this technique. Some school systems might feel that learning this technique would interfere with the traditional learning habits in which the children are presently being instructed. Parents would be well-advised to check with their children's teachers before purchasing either of these highly innovative programs. RRA

*Math Strategy* and *Spelling Strategy*, by Robert B. Dilts, Special Delivery Software/Apple Computer (20525 Mariani Avenue, Cupertino, CA 95014; 408-996-1010). Each, $45.

**Frazzle!** By J. C. Nolan. *Frazzle!* introduces the fine technique of pattern mine-laying. Your ship is inside a force field and is being attacked by beasts from another dimension. Only six of these creatures can attack per level. Your job is to eliminate them before they touch your ship, or before your ship comes in contact with the force field. You cannot shoot at these beasts; all you can do is lay down energy probes similar to space mines and lure the beasts into contact with the probe, thus destroying them. Unfortunately, sixteen is the maximum number of probes allowed at one time within the force field. Any contact with your own probe will destroy you also, so it can get rather crowded within that force field very quickly, with maneuvering space at a premium.

There are five levels to overcome. On the first level, the instant the six beasts are neutralized, the whole group is regenerated, and each beast has six lives. On levels two and three, each beast resurrects instantly. The fourth and fifth levels introduce killer beasts who dive straight at your ship like kamikazes. These never completely die, but are reborn time after time until they finally overwhelm your ship. *Frazzle!* can be played by one person or two, competing to see who is the better sower of mines against the hordes of beasts. The game is unlike the regular arcades and a good choice for young players. RRA

*Frazzle!*, by J. C. Nolan, Muse (347 North Charles Street, Baltimore, MD 21201; 301-659-7212). $24.95.

**Death Race '82.** By Don Fudge. Once again, Don Fudge (compiler of *Hi-Res Secrets* and author of the unique *Zero Gravity Pinball*) has come up with a game that knocks at the frontier of arcade design and doesn't quite get in. Combining the twisting-turning style of Monte Carlo Grand Prix racing with the grim reaper shadow of *Deathrace 2000*, this game is intended to keep you on the edge of your seat.

You're the driver of a turbo car trying to escape at high speed through ten convoluted mazes, hotly pursued by robot cars firing lasers. Your only weapons are bazooka rockets, portable oil slicks, and, if you have superhuman driving skills, the ability to cause the robot cars to crash. It is the driving skill that enables you to survive the later mazes. The rockets, though, are especially useful for blasting shortcuts through the walls of the mazes. Each time you crash you lose points, and, if the enemy gets you, you must start all over.

If you pursue the game long enough to master its "unique paddle control," you can increase the speed of the game for the next try. The ten possible speed settings raise the difficulty quotient. RRA

*Death Race '82*, by Don Fudge, Avant-Garde Creations (Box 30160, Eugene, OR 97403; 503-345-3043). $29.95.

**Early Games for Young Children.** By John Paulson. There is a tremendous shortage of good preschool computer programs, so any new one is welcome. *Early Games for Young Children* is not a flashy program. it does impart basic training in numbers and letters in a competent manner. Even the menu is a picture menu, so no reading skills are needed. The program is geared for kids two years old to seven years old.

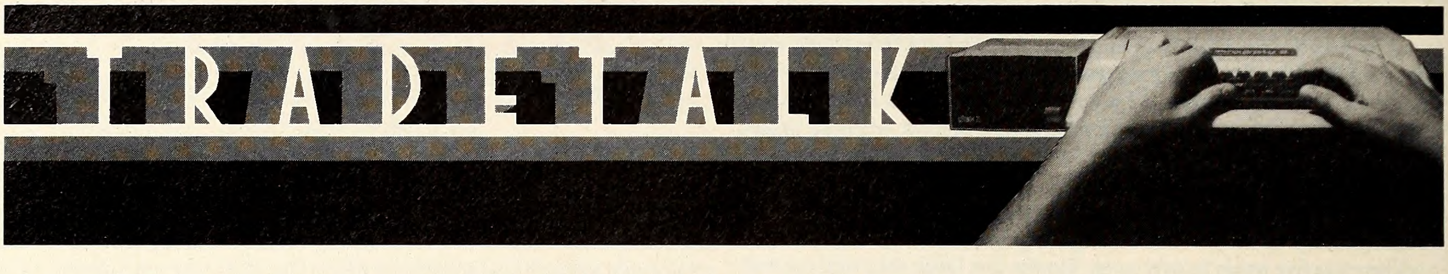
There are nine programs, each providing training in a different area. One of the nicest features of this program is that it starts out teaching children the Apple keyboard. The numbers and letters are shown full screen in color.

Then children are asked to count multicolored blocks. Mastering that, they move on to easy addition and subtraction, using the same blocks. The drills are slow paced, and children like to do them over and over again. For the older group, alphabet drills ask children to enter the letter after the one shown on the screen. A shape comparison program trains perception of similarities and differences.

Children love to print their names. The *Name* program requires an adult to enter a child's name. Then the child is asked to spell their name in small letters on the screen. When the youngster gets it right, the computer prints the name in multicolored screen-high letters. The squeals of delight when children see this are worth the price of the program.

Included also is a neat little drawing program that enables children to draw on screen with very simple commands. Upper keys draw upward, lower keys draw downward, corner keys draw diagonally, and the space bar changes color. That is all they need to know. Any picture they make can be saved to disk for later retrieval, perhaps to show parents when they get home. RRA

*Early Games for Young Children*, by John Paulson, Learning Tools (Suite 140, Helard Plaza North, Minneapolis, MN 55426; 800-328-1223). $29.95. ■




☐ **Syntauri** (Palo Alto, CA) has announced completion of a private placement financing led by **Steve Wozniak,** cofounder of Apple Computer and independent investor. Elected to Syntauri's board of directors: **Nicholas Fortis,** previously vice president of finance for Nestar Systems; **Ellen V. B. Lapham,** founder and president of Syntauri Corporation; **Philip Roybal,** manager of communications programs for Apple Computer; and **Thomas A. Skornia,** recently vice president and general counsel for Advanced Micro Devices and cofounder of Third Wave Investors, a venture capital investment fund.

☐ **Anthony P. Morris,** president of **Morris Decision Systems** (New York, NY), the national account support dealer for Apple Computer, has named three new vice presidents to the company: **Roger A. Williams,** vice president of sales and marketing; **Kevin S. Clougherty,** vice president of research and technology; and **Robert M. Bowen II,** vice president of finance and administration.

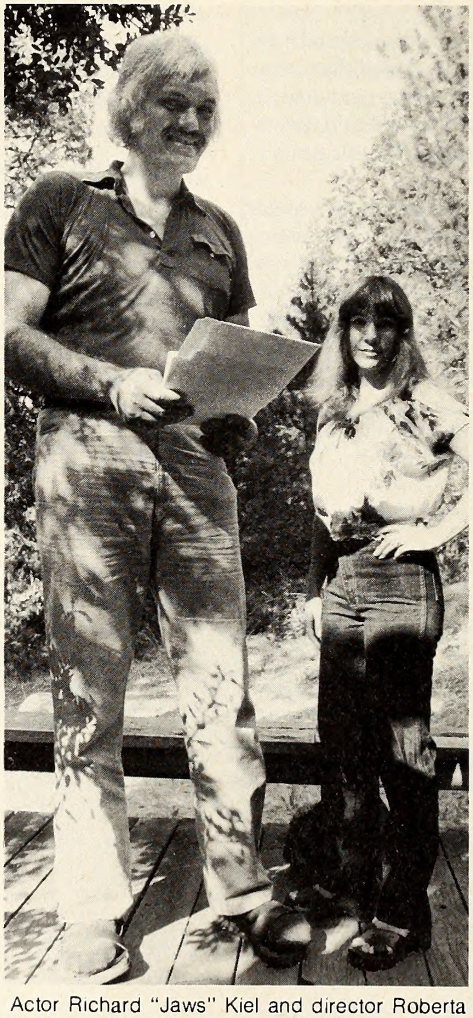
☐ **Advanced Logic Systems** (Sunnyvale, CA) has appointed **Tech Plus** (Boston, MA) its exclusive sales representative in the United States. Tech Plus will handle the ALS Synergizer product line for adding CP/M to the Apple II. "This will complete the support of our existing dealer or distributor relationships," comments **Dick Ribas,** vice president of sales for ALS.

☐ **Artsci/Softape** has moved. The software company is now located at 5547 Satsuma Avenue, North Hollywood, CA 91601.

☐ To accommodate rapid growth experienced during the past year, **SofTech Microsystems,** marketer of the UCSD p-System, has relocated to a 33,000-square-foot facility in the community of Rancho Bernardo, California, in the northern part of San Diego. The new quarters are located at 16885 West Bernardo Drive, San Diego, CA 92127. The new phone number is (619) 451-1230.

☐ **Sierra On-Line** (Coarsegold, CA), formerly On-Line Systems, is budgeting nearly one million dollars in the coming year for various dealer support projects. The company has leased CompuVision, a computer/video "robot sales device," and is featuring actor Richard Kiel in a series of "infomercials" in one hundred stores. A WATS line and dealer support newsletter are available, and the marketing department is creating several other dealer support programs involving posters, point of purchase displays, and an ongoing publicity program.

☐ **Dan Illowsky,** best known as the author of the popular *Snack Attack* from **DataMost** (Northridge, CA), has formed his own company, **Funtastic,** in Drexel Hill, Pennsylvania. At first, Funtastic will publish only Illowsky's new games; late next year, *Snack Attack* and *County Fair* will also become available through the new company. Meanwhile, DataMost has withdrawn the apparently prematurely released Illowsky game, *Space Kadet*, from the market. The future of Tom Corbett is as yet unknown.



Actor Richard "Jaws" Kiel and director Roberta "Hot Tub" Williams take a break during the shooting of a video "infomercial" for Sierra On-Line.

☐ **Dr. Robert Harp,** formerly chairman of the board of Vector Graphic, has announced the executive team and corporate strategy of **Corona Data Systems** (Westlake Village, CA), the firm he established in the summer of 1981 and has managed as board chairman since his resignation from Vector last April. **Dan Carter,** formerly executive vice president of Televideo Inc.'s computer systems division and general manager of the microcomputer systems division at Commodore Business Machines, has been named president of Corona. **George McMurtry,** Corona's vice president of sales, was previously national sales manager for microproducts at Pertec Computer Corporation. **Robert Kramarz** will continue as marketing vice president for the firm. He was Vector's manager of market development and director of corporate communications. A vice president of finance and operations has also been selected, according to Dr. Harp, but his name has not been released.

In its first overseas distribution contract, Corona has signed with **Management/Personal Computers** (London, England) to market Corona's line of hard disk subsystems in Great Britain. The agreement reportedly does not limit Corona's right to sell products in British retail stores such as Computerland. Corona's announced two-step distribution policy for worldwide sale and support of its subsystems will make end-user sales the responsibility of Independent Sales Organizations—retail dealers, OEMs, and system houses—and stocking master distributors will be responsible for supplying the ISOs. Corona will then supply and support the distributors. Three new North American offices will be responsible for training distributors in selling to dealers and for providing back-up support to the distributors.

Corona expects to ship a thousand units per month by year end. Current plans call for shifting production facilities from Chatsworth, California, to corporate headquarters in Westlake Village, California.

☐ **Taurus Software** (San Francisco, CA) has been formed to provide user-friendly, nontechnical applications software and control programs for CP/M-based systems. Their initial product, *CP+*, was introduced at the West Coast Computer Faire in March. Taurus incorporated in February, obtaining venture capital from Merrill, Pickard, Anderson & Eyre, Bank of America's high technology specialists for managing venture capital investments.

☐ **Entech,** the computer skills center (Commack, NY), has opened to provide hands-on training in microcomputers and word processing equipment to businesses, self-employed professionals, and individuals. Seminars range from the essentials of operation—assuming no prior knowledge—to the advanced operation of the most sophisticated hardware and software available. Custom training programs can be arranged off-site. **Philip Ackerman,** president, cofounded Databit and served as vice president of engineering. Brochures, seminar schedules, and costs are available by calling (516) 543-3352.

☐ **Computer Learning and Software Services** (Marquette, MI) is a combination educational and retail service offering a variety of computer education classes for business professionals, educators, home users, and children. In addition, C.L.A.S.S. provides consultation and support services, consumer education classes, and software/hardware sales. C.L.A.S.S. welcomes inquiries from dealers who wish to sponsor seminars in their areas. Call (906) 225-0433.

☐ The College of Education at the **University of South Alabama** (Mobile, AL), in cooperation with the School of Continuing Education, will sponsor a two-day workshop on "Microcomputers in Education" in Biloxi, Mississippi, on November 11 and 12, 1982. For details and registration forms, contact **Judy Campbell,** University of South Alabama, Mobile, AL 36688; telephone (205) 690-6528.

☐ **Information Systems and Supplies** is the latest division of **Leading Edge Products** (Canton, MA). The division was formed, according to chairman **Michael Shane**, "as a result of our recognition of the special needs of the National Office Products Association and the concept that these markets have outstanding growth potential for microcomputer products and services. We intend to work very closely with the members of these groups to provide them with the kind of products and information that will help them to greater sales and better service for their customers." **Louis Sebok**, former head of Wang's office supplies division, has been appointed general manager of the NOPA/NOMDA divison.

☐ One hundred software packages, spanning virtually every popular operating system, are being introduced by **Single Source Solutions,** a new software company in Concord, California. President **Michael L. Dean** says the company plans to market its programs—currently including utilities, languages, business and word processing applications, educational tools, and games—through computer dealers and OEMs, developing a strong dealer/OEM network to allow lower prices for the end user.

☐ **Business and Professional Software** (Cambridge, MA), developer of *Apple Business Graphics,* has contracted with ten independent sales organizations representing Apple Computer and peripheral manufacturers to market their *Screen Director* and *PIK* software packages. The move marks BPS's first step toward developing a national sales network. "These ten rep contracts will significantly bolster our dealer network, which already comprises more than three hundred companies nationally," says sales manager **Jeffrey Turner.** "Our plans call for us to have the entire country covered with rep territories and to double the number of our dealers this fall. In addition, we have contracted distributors in Europe, Asia, Australia, and South America."

☐ **Shugart Associates** (Sunnyvale, CA) has decentralized the majority of its operations and formed five major divisions. The new divisions and their respective vice president/general managers are rigid disk drive division, **Tom Gardner;** floppy disk drive division, **Carl Neun;** Sunnyvale division, **Paul Penney**; marketing and sales division, **George Sollman**; and Optimem division, **Peter Lloyd.** Shugart's central corporate staff now consists of four groups, headed by their respective vice presidents: quality, **Tom Gilmer;** finance and planning, **Bob Bledsoe;** personnel and administration, **Chris Carlton;** and technology, **Mike Feldstein.** The company has reduced its thirty-five hundred member work force by approximately one hundred twenty-five indirect (salaried) employees. According to president **James Campbell,** the reduction in indirect labor eliminates job duplication and streamlines the division and central corporate staffs.

☐ **Vern Raburn,** former vice president of consumer products with **Microsoft** (Bellevue, WA), is now executive vice president and general manager of **Lotus Development Corporation** (Cambridge, MA). He will oversee marketing and operations, and the launching of Lotus's latest product, *1-2-3.* Raburn states that he welcomes "the opportunity to work on a key product to fill an important need for integrated productivity tools."

☐ **Software Distributors** (Culver City, CA) has appointed **Jack Hennessy** sales manager responsible for the company's domestic and international sales. His additional responsibilities will include marketing liaison, customer service, and development of new sales markets.

☐ **Queue** (Fairfield, CT), the publisher of the *Microcomputers in Education* newsletter, has opened its first Computer Learning Center at 161 Kings Highway, Fairfield, CT 06430. The center offers computer-assisted tutorials on a wide variety of academic subjects, using the Apple II and Apple III, among others. They also offer specialized programs on SAT and GRE preparation, computer literacy and programming, and computer applications in business. Queue is planning to open additional learning centers and is seeking people, particularly teachers, interested in owning or operating a Computer Learning Center in other areas.

☐ Following the opening of new production and administration facilities at 7 Hangar Way in Watsonville, California, **Ultra Magnetics,** a flexible disk manufacturing company, has appointed **Robert Glass** director of manufacturing and **William Swain** vice president of finance and administration. Glass, former head of the flexible disk media division at Memorex, will be responsible for all manufacturing operations including production, manufacturing engineering, materials scheduling, and facilities planning and maintenance. Swain has been promoted from the position of vice president, controller, for Frank Scott Enterprises, the Aptos, California, venture capital firm backing the company.

☐ **George Sass,** president of **Apple Advertising** of Annapolis, Maryland, and **Jane Sass,** president of **Sassafras Studios,** have announced the merger of their two companies. The corporate structure of Apple Advertising remains the same, except that it will now trade as Sassafras. States George Sass, "Our companies have been working closely during the past year, and there will be several advantages to working under one corporate structure." Also, a survey indicated a lot of positive awareness for the new name. And there was one other thing, regarding the ad agency's original moniker: "We chose the name five years ago in connection with the first ad on earth before Apple Computer became a household word," says George Sass. "It's time to change our name and end the confusion."

☐ **The Denver Software Company** (Aurora, CO) is spending one million dollars in the initiation of a national distribution network of sixty manufacturers' representatives for its products, part of a mass merchandising effort including on-site product demonstrations, display racks for software and literature, national advertising, product comparisons, and videotape presentations. DSC will conduct quarterly training programs for reps and has created a product support department with a hot line number to provide additional support to dealers and end users. Purchasers of DSC's new "password registration system" application programs must call the hot line number and ask for the individual package code password to unlock the product for use. The product support department uses the system to conduct market research and control inventory.


**Completely Redesigned.**
**Now, the Grappler+.**

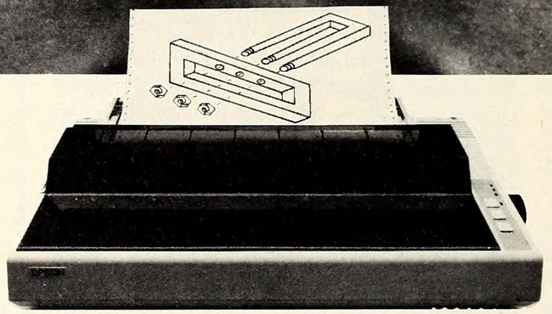
The original Grappler was the first graphics interface to give you hi-res screen dumps from your keyboard. The new Grappler+ with *Dual Hi-Res Graphics* adds flexibility with a side-by-side printout of page 1 and page 2 graphics.

Interfacing the Grappler+ to a wide range of printers is easy as changing a dip switch. 4K of exclusive firmware makes the Grappler+ the most intelligent, full-featured Apple® Printer Interface made. And, the Grappler+ is Apple III compatible.*

The imitations are many, so insist on the #1 Apple Graphics Interface on the market. Insist on the Grappler+. Available now at most Apple dealers.

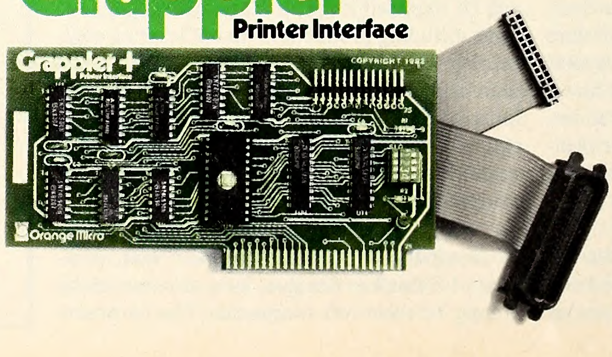
*Requires additional software driver.
**Requires graphics upgrade.

©Orange Micro, Inc. 1982



ACTUAL APPLE II PRINTOUT USING GRAPPLER AND EPSON MX100

## With The Grappler+™ Printer Interface




**The Grappler+ Features:**
• Dual Hi-Res Graphics • Printer Selector Dip Switch • Apple III Compatible* • Graphics Screen Dump • Inverse Graphics • Emphasized Graphics • Double Size Picture • 90° Rotation • Center Graphics • Chart Recorder Mode • Block Graphics • Bell Control • Skip-over-perf • Left and Right Margins • Variable Line Length • Text Screen Dumps.

The Grappler+ also works with Pascal and CPM.

**The Grappler+ interfaces with the following printers:**
• Anadex • Centronics • Datasouth • Epson** • NEC • C-Itoh • Okidata**

The original Grappler is available for IDS 460, 560, Prism, Microprism.

Orange Micro Inc.
3150 E. La Palma, Suite G
Anaheim, California 92806
(714) 630-3620

Foreign Dealer Inquiries Welcome TX183511 CSMA

# THE PASCAL PATH

By Jim Merritt

## Tools of the Craft, Part 17

**Plymouth Rock and Dinner Rolls.** You should be reading these words sometime in early November, before any of us has had a chance to sit down to a sumptuous Thanksgiving feast. It's unlikely that we'll be worth much of anything afterward, and, besides, you'll probably prefer to busy yourself with holiday frolicking, rather than sitting, hermitlike, at a computer console. However, to keep the well-known effects of the impending holiday season from impeding our progress along the Pascal Path, we'd better bite off a big chunk of the cable TV data-manipulation program now, while we still enjoy the clarity of thought that comes from having comfortably empty stomachs.

Here is the program *Cable*, which was left unfinished (but operational) at the conclusion of last month's column:

```
PROGRAM
    Cable;
  (* DESCRIPTION: Permit the interactive
     establishment and maintenance of records
     concerning a Cable TV franchise's
     subscribers. *)
    CONST
      Header=   'CABLE DATA BASE (V1.1
                 10-Sep-82)';
      Blank=     ' ';
      (* Maximum house number—
         unrealistic *)
       MaxHNum= 999;
      (* Customer account numbers range
         from 1 to MaxAcctNum; 0 as an
         account number signifies that the
         home in question contains no
         subscribers. *)
       NoSubscriber=
                 0;
       MaxAcctNum=
                 MaxInt;
    TYPE
       AcctNumType=
        NoSubscriber .. MaxAcctNum;
       StreetName=
        (Redwood, Tanglewood, Sandalwood,
         Driftwood);
       HouseNumber=
        1 .. MaxHNum;
       (* How our model is structured:
             A Town is composed of named
               Streets.
             A  Street is composed of
                numbered Homes.
             A   Home is modeled by the
                 information we wish to
                 record about it and its
                 residents. *)
         Home=
          AcctNumType;
         Street=
          ARRAY [HouseNumber] OF Home;
         Town=
          ARRAY [StreetName] OF Street;
       CComType=        (* Cable program
                            commands *)
         (Change, Display, Quit);
    VAR
     Smallville
      :Town;
     UserQuits
      :Boolean;
    FUNCTION
     Capital (Ch
               :Char)
       :Char;
    (* Return Ch, converted to upper case
       (capital), if Ch is lower case. *)
    BEGIN (* Capital *)
     Capital := Ch; (* No change unless lower
                       case *)
     IF ((Ch >= 'a') AND (Ch <= 'z'))
       THEN (* It's a lower case letter—
              transform it! *)
        Capital := Chr(Ord(Ch) - Ord('a') +
          Ord('A'));
       (* Otherwise, it's not a lower case letter,
          so leave it alone. *)
    END (* Capital *);
    PROCEDURE
     NewTown(VAR T: Town);
    (* DESCRIPTION: Ready the model of a
       new town, T, by "emptying" all its
       Homes. *)
    BEGIN (* NewTown *)
     (* stub *) WriteLn(Output, ' NEWTOWN:
          NOT YET IMPLEMENTED');
    END (* NewTown *);
    PROCEDURE
     ChangeTown(VAR T: Town);
    (* DESCRIPTION: Permit the interactive
       selection and modification of one (or
       more) Home(s) in a Town, T. *)
    BEGIN (* ChangeTown *)
     (* stub *) WriteLn(Output,
          '  CHANGETOWN: NOT YET
          IMPLEMENTED');
    END (* ChangeTown *);
    PROCEDURE
     DisplayTown(VAR T: Town);
    (* DESCRIPTION: Permit the interactively
       controlled display of information
       recorded for one or more Home(s) in a
       Town, T. *)
    BEGIN (* DisplayTown *)
     (* stub *) WriteLn(Output,
          '  DISPLAYTOWN: NOT YET
          IMPLEMENTED');
    END (* DisplayTown *);
    FUNCTION
     CableCommand
      :CComType;
    (* DESCRIPTION: Prompts for and
       accepts user input characters until one
       corresponds to a CComType command,
       then returns the matching value. C, D,
       and Q map onto Change, Display, and
       Quit. Treats capitals and lower case as
       identical. Echoes blank for blank,
       command name for command
       characters, and the input character itself
       along with the message "—NOT A
       COMMAND" for all others. Pressing the
       return key is equivalent to pressing
       space bar. *)
       CONST
        Prompt= 'Command: Change,
            Display, Quit >> ';
       VAR
        ComCh
         :Char;
        Valid
         :Boolean;
     BEGIN (* CableCommand *)
       Write(Output, Prompt);
       Valid := False;
     REPEAT
       Read(Keyboard, ComCh);
       CASE Capital (ComCh) OF
        'C':
         BEGIN
          Valid := True;
          Write(Output, 'Change');
          CableCommand := Change;
         END;
        'D':
         BEGIN
          Valid := True;
          Write(Output, 'Display');
          CableCommand := Display;
         END;
        'Q':
         BEGIN
          Valid := True;
          Write(Output, 'Quit');
          CableCommand := Quit;
         END;
        Blank:
         Write(Output, ComCh);
       END (* CASE Capital (ComCh) *);
       IF ((NOT Valid) AND (ComCh <>
             Blank))
        THEN
         BEGIN
          WriteLn(Output, ComCh, '—NOT A
               COMMAND');
          Write(Output, Prompt);
         END;
     UNTIL Valid;
     WriteLn(Output);
     END   (* CableCommand *);
BEGIN (* Cable *)
   WriteLn(Output, Header);
   WriteLn(Output);
   NewTown(Smallville);
   UserQuits := False;
   REPEAT
    CASE CableCommand OF
     Change:
      ChangeTown(Smallville);
     Display:
      DisplayTown(Smallville);
     Quit:
      UserQuits := True;
     END (* CASE CCom *);
   UNTIL UserQuits;
END (* Cable *).
```

This month, we'll develop the stub procedure NewTown and prepare ourselves for the "big jobs" of writing DisplayTown and ChangeTown next month. With any luck, the groundwork we do now will help us to dispose of the last two procedures without too much difficulty. The less effort we need to expend to finish the program over the year-end holiday, the better!

**NewTown.** If we had to use "plain English" in expressing the "emptying" of a town relative to the requirements of our model, we might describe it this way: "For each street in the Town T, mark every house on it as containing no subscribers." Use of the word *for* in the description suggests that we might profitably employ a FOR loop in turning the computer's attention from one street to the next. We need only have the FOR loop control variable—which we'll call StreetNow—step through all the values of type StreetName. Within the loop body, we may then use StreetNow as the first index to T. You can see that if we do so each successive iteration of the loop selects a new street automatically.

The process of selecting successive Houses on a Street is not different in concept from that of selecting successive Streets within a Town. Thus, we may nest a second FOR loop within the body of the StreetNow loop and use it to empty the houses on the street indicated by StreetNow. We'll give this "inner" loop a control variable, HouseNow, which, because it can assume integer values in the HouseNumber subrange, is therefore suitable as the second index to T. We may now replace the stub for NewTown with a very concise and elegant Pascal translation of the rough description presented earlier:

```
  VAR
    StreetNow
      :StreetName;
    HouseNow
      :HouseNumber;
BEGIN (* NewTown *)
  FOR StreetNow := Redwood TO Driftwood
      DO
    For HouseNow := 1 TO MaxHNum DO
      T[StreetNow][HouseNow] :=
            NoSubscriber;
END (* NewTown*);
```

This method of emptying a town is perfectly satisfactory from a procedural standpoint. It does exactly what we want, and its similarity to our rough description makes it very readable. However, try compiling and executing a version of *Cable* that uses this code for NewTown. From the time you see *Cable*'s "header" to the time that the program's prompt line appears on your video screen, several seconds—perhaps several uncomfortable seconds—elapse, in which you have time to wonder if the program is working at all. NewTown is the culprit, requiring nearly eight seconds to empty an entire Town. Eight seconds may not sound like a great deal of time, but studies have shown that even this brief interval can seem like an eternity to someone who is sitting (helplessly) at a computer terminal, waiting for a response. If you compiled and executed the *Cable* skeleton, using NewTown as presented earlier, and then proceeded to fidget during the "dead air" that characterizes NewTown's operation, you may be wondering if this pause can possibly be shortened.

The execution of any Pascal statement requires some small amount of time, although any single statement generally executes in much less time than you can perceive. Complicated programs often seem to take a while to do their jobs, because hundreds and thousands of statements must be executed before you see any results. For instance, any call to NewTown results in the execution of at least 3,996 statements—one assignment statement for every element in the Town T. This is a direct consequence of NewTown's design. We can change the design slightly and achieve substantial increases in NewTown's speed by exploiting two facts:

1. Pascal permits entire arrays to be assigned with only one statement.

2. "Mass assignments" are faster than equivalent assignments to individual elements.

The first fact reminds us that it is possible to empty an entire Street of 999 Houses at once, while the second exhorts us to do so whenever possible.

Let's take a moment to discuss the rather remarkable second claim. Remember that the Pascal compiler produces *P-code*, which is in turn interpreted by a 6502 machine-language program, the Apple Pascal *P-machine*. Usually, the compiler translates a single statement into several different P-codes. About the same number of P-codes are produced for any assignment statement, whether it deals with individual variables, array elements, or entire structures, so let's simplify the discussion by supposing that one assignment translates into one P-code. In other words, in our hypothetical example, one P-code may symbolize the transfer of any amount of information from one location in memory to another.

For every P-code, the P-machine has to take four separate steps: fetch the code, recognize it, identify (and often fetch) the data on which it is to operate, and perform the operation. To simplify the discussion, let's say that each of the first three steps requires one unit of time to complete, and that the fourth can last for one or more units, depending on the complexity of the operation being performed. The fourth step in transferring a single Boolean value from one variable to another might take just one time unit, while transfer of a hundred-element array might require 100 units. Thus, the P-code corresponding to the assignment of a single value executes in a total of four time units, while that corresponding to the assignment of an entire array requires 103 units.

Given these assumptions (which model the P-machine's behavior well enough for this discussion), let's compare the speed of a FOR loop containing a "single-value" assignment statement as its body against one "mass assignment" statement in copying the contents of one 100-element array into another. Since the FOR loop's body consumes four time units every time it is executed and must be executed one hundred times, it's easy to see that the FOR loop cannot finish in less than four hundred time units. In reality, the loop probably requires at least twelve hundred units, since some extra P-codes must be executed in order to increment and test the value of the loop control variable prior to each execution of the body. As we've seen, the mass assignment P-code needs only 103 time units—between one-half and one-fourth of the FOR loop's time—to do the same job. This substantial improvement in performance stems mainly from the fact that the mass assignment P-code spends more time doing useful work and less time dealing with "bookkeeping overhead" than one hundred (to three hundred) separate P-codes.

We oversimplified in the explanation given earlier so that we could concentrate on acquiring an intuition for the reasons why whole-array assignment is inherently faster than element-by-element assignment. In doing so, we overlooked many aspects of the P-machine architecture and the Pascal system environment that affect execution speed. So, even though we can be reasonably sure that we can cut NewTown's execution time by clearing entire Streets at once (thus reducing the total number of P-codes executed) whenever possible, we should actually try out the new technique to get a feel for the degree of performance improvement it promotes in the real world.

The following body for NewTown uses mass assignment as often as possible. First, it empties Redwood Street, House by House. Then, it uses Redwood as a template to clear all subsequent Streets through mass assignment. This version of NewTown's body should replace the one presented earlier; note that the declarations for StreetNow and HouseNow remain necessary:

```
BEGIN (* NewTown—fast version *)
  FOR HouseNow := 1 TO MaxHNum DO
    T[Redwood][HouseNow] := NoSubscriber;
  FOR StreetNow := Succ(Redwood) TO
      Driftwood DO
    T[StreetNow] := T[Redwood];
END (* NewTown *);
```

Granted, this latest NewTown doesn't correspond as well as its predecessor did to our initial description of the initialization process. On the other hand, while remaining perfectly readable and understandable, it reduces the number of distinct assignment statements executed, from 3,996 to 1,002, and cuts execution time to just over two seconds—only about 30 percent of the time needed by the first routine to do the same job.

Of course, you shouldn't worry overmuch about optimizing your code for speed or size as we've done here. The more important thing is to concentrate on getting the method right, so that your code works and is readable. Good programmers usually defer optimization until the entire program operates correctly. However, initialization code rarely involves critical methodology. More often, it consists of nothing more than dozens—sometimes hundreds—of assignment statements. Intelligent restructuring of assignments can often result in big savings of both execution time and RAM memory space, with little or no effect on program clarity. Therefore, such economies are often included by concerned programmers right from the very start, especially when they are writing initialization code. In particular, a good initialization strategy is to use mass assignment whenever the initialization code involved will be called regularly or at times when human beings are waiting to see results.

**Prelude to DisplayTown and ChangeTown.** The vague specification for DisplayTown states that it permits us to display an entire town or just part of one. Offhand, our intuition tells us that display, like initialization, can be driven by simple loops. The real challenge here is to find a mechanism that lets the user specify the entire Town or a subset of it with equal ease.

It seems reasonable to identify a Home by its address, consisting of a HouseNumber and StreetName. Given a mechanism for collecting an address from the user, we can acquire two addresses, then display all Homes from the first address to the second. ChangeTown also requires the collection of at least one address from the user, so this same mechanism, designed correctly, can be employed by both routines. Similarly, some means for displaying the address and contents of a Home must be created, and, while it will certainly be of use when we write DisplayTown, it may also prove handy for ChangeTown.

Clearly, our "address collection and display mechanism" is simply another variation of data conversion, this time between the character representation of an address and its internal form (as a HouseNumber and StreetName). Let's follow common sense and agree that the character representation of a Home address is an integer in the range of 1 to 999, followed by one or more blanks and then the name of a Street (for example, Redwood, Driftwood, and so on). How do we deal with the character representations of an address?

Prior to *Cable*, we have written programs that interact with the human user without requiring the entry of words, phrases, or names. Single-character yes or no answers and the occasional number or two are all that our programs have ever needed to acquire from the outside world. Of course, we have been able to send the user wordy prompts, and even long messages, by encoding them as String constants within our programs.

We already know how to acquire a number from the user; either we can take advantage of the facilities that are built into Read and ReadLn, or we can use our own IntegerInput routine. But how may we receive a sequence of characters, and where may this kind of input data be stored while our program examines it? It should be obvious that we can use arrays to solve our problem. For instance, we could declare a variable, Answer, to be an "ARRAY [1..40] OF Char." It would be a relatively simple matter to write two procedures, one that would accept input characters into this structure from the keyboard and another to display the characters held by Answer on the screen. However, Apple Pascal already includes a predeclared data type, named String, for which these and many more operations are provided.

**Strings.** Since we have already been using String constants and literals for many months now, we won't spend any time reviewing them. Instead, we'll dive right into the thick of things, covering String variables and the various predeclared routines that Apple Pascal provides for String manipulation.

**String Variables.** A *String variable* (from now on, just called "String") is a container, like any other variable. The job of any String is to hold a (possibly empty) sequence of characters. Strings are declared, along with all other variables, in a VAR section.

There are two parts to the type descriptor of a String variable. One is the identifier "String," and the other is a bracketed Integer constant that determines the maximum length—the *physical length*—of the String in question. For example, "String[1]" indicates a String that can contain only one character, while "String[40]" defines a String that can hold up to forty characters in a row. The physical length specification is optional; the lone identifier "String" is equivalent to "String[80]." The smallest permissible physical length is 1; the largest is 255.

For any Integer constant PLen, valued from 1 to 255, "String[PLen]" is equivalent to writing "PACKED ARRAY[0 .. PLen] OF Char." (Although included for reasons of accuracy, the keyword *PACKED* is irrelevant to this discussion and should be ignored here; its meaning will be explained in a future column.) A String, however, is no ordinary Char array. Consider the String variable S, declared as "S: String." Under normal circumstances, S[0] is not subject to direct access by your program. If you assign a String constant to S, Pascal will put the characters into the variable starting at S[1]. Any attempt by your program to use S[0] causes a "Value Range Error" to occur at run time. In fact, in writing your program, you should treat S as if it were declared with the type descriptor "Array [1 ..80] OF Char." In other words, you should forget S[0] entirely.

The Pascal system doesn't forget S[0], though. It uses that slot to keep track of the String's *dynamic length.* For example, after the assignment "S := 'Hello' " the String S logically contains a five-character value. Appropriately enough, Ord(S[0]) becomes 5, indicating that only the first five slots in S contain meaningful characters and that all other character positions should be considered unoccupied. In fact, if your program tries to access a String array element using an index value larger than the current dynamic length of S, a "Value Range Error" will occur. Rather than try to use S[0], you should employ (and, unless you know arcane tricks, you *must* employ) the predeclared function Length(S) in order to determine the dynamic (logical) length of S, where S is a variable or constant of any String type. Note that

"Length(S)" is conceptually equivalent to "Ord(S[0])." However, by making the length of a string available only through a function, the designers of Apple Pascal tried to help make your programs more readable and also to remove the possibility that a program might accidentally change the value of a String's length count, perhaps rendering the String value invalid.

To summarize, a String variable's dynamic length (as reported by the Length function) can change, depending on the String values assigned to it. The variable's physical length, however, is fixed, limiting the maximum number of characters that a String can hold.

**Apple Pascal String Facilities.** Besides Length, there are several other predeclared functions and procedures that manipulate String values. The function Concat takes one or more arguments, all of which must be valid String values. It produces a String value that is the concatenation of all the arguments. To illustrate, the following Boolean expression is True:

```
(Concat('Abra', 'cadabra') = 'Abracadabra')
```

The Pos function takes exactly two String arguments, a Pattern and a Source, in that order, and produces an Integer result, which is zero if the Pattern does not appear in the Source and greater than zero otherwise. If the function value is nonzero, it indicates the character position in Source at which the leftmost occurrence of Pattern begins. Note that Pos does *not* treat capital letters as equivalent to their lower-case counterparts. "Pos('in', 'winching')" returns the value 2, while "Pos('IN', 'winch')" and "Pos('123', '456')" both return 0.

The useful function Copy accepts three arguments: a String value, Source; and two Integer expressions, Position and Count. Copy returns a String value that is the *substring,* which begins at the specified Position in Source and includes the number of characters specified by Count. For instance, Copy('12345',3,1) yields the result 3, and Copy('12345',3,2) gives 34. Copy returns the empty String if Source is empty, if Position is less than 1 or greater than Length(Source), if Count is less than 1, or if (Position+Count−1) is greater than Length(Source). To illustrate, Copy('ABC',0,3), Copy('ABC',1,4), and Copy('ABC',2,0) all return the empty (null) String. You can prove this for yourself by experimenting with the following program, which asks for Source, Position, and Count and displays the value and Length of Copy(Source, Position, Count):

```
PROGRAM
  TestCopy;
(* Exercise Apple Pascal's Copy function *)
  CONST
    Empty=      '';
  VAR
    Source,
    CopyVal
       :String;
    Position,
    Count
       :Integer;
BEGIN (* TestCopy *)
  REPEAT
    Write(Output, 'Source: ');
      ReadLn(Input, Source);
    Write(Output, 'Position: ');
      ReadLn(Input, Position);
    Write(Output, 'Count: ');
      ReadLn(Input, Count);
    CopyVal := Copy(Source, Position, Count);
    Write(Output, 'Copy(Source, Position,
      Count)= ''');
    Write(Output, CopyVal, ''' Length= ');
    Write(Output, Length(CopyVal));
    WriteLn(Output);
    WriteLn(Output);
  UNTIL (Source = Empty);
END (* TestCopy *).
```

In using this program, make sure you enter a null Source as your last experiment, since that will signal the end of the REPEAT loop and also the end of the program. Try verifying the examples given above and feel free to invent your own. Don't be afraid to use negative numbers for either Position or Count; Copy deals sensibly with them.

Three procedures, Str, Insert, and Delete, round out Apple Pascal's repertoire of String-manipulation facilities. Str takes two arguments—an Integer expression and the name of a String variable—and sets the value of the variable to the character representation of the expression value. Assuming S is a String variable, Str(3, S) would leave the String 3 in S, while Str(2+3*4+7, S) would put 21 in S.

Insert requires three arguments—a String expression, the name of a String variable, and an Integer expression. The String expression is inserted into the String variable just prior to the character position specified by the Integer expression. For instance, let's suppose that String variable S contains *Path.* After either of the procedure calls Insert ('scal Pa', S, 3) or Insert('Pascal ', S, 1), S would contain *Pascal Path.*

Delete also takes three arguments—the name of a String variable and two Integer expressions. The first expression is taken as a character position in the String variable, and the second expression is the number of characters to remove from the String. Starting at the specified character position, Delete removes the given number of characters from the String. If S contains the result of the Insert example just given (*Pascal Path*), then Delete(S,3,7) would restore the String value *Path* to S. So, by the way, would Delete(S,1,7).

**Strings, Functions, and Parameters.** Although Apple Pascal predeclares functions that return String values and that can be used in String expressions, a String is still considered an array type by the compiler, so you are not permitted to define functions of your own that return Strings. Of course, you can write procedures that manipulate arbitrary String variables through the VAR parameter mechanism.

Note that the physical lengths of actual and formal VAR String parameters should match exactly. At the very least, the physical length declared for the formal parameter must not be greater than that of the actual parameter. Otherwise, a procedure or function would be "fooled" into thinking that the actual parame-

# V new for Apple II™

## I. APPLEsurance II. . .

**DUAL AUTOMATIC DIAGNOSTIC AND ASSURANCE SYSTEM**

The only double-safe system now available for insuring the reliability, integrity, and dependability of your Apple II and peripherals. Identify problems before you lose time and data. APPLEsurance II includes:

(1) A standard disk controller card which also automatically runs sophisticated diagnostic routines on basic system hardware each time you power-up. Descriptive error messages indicate any problem areas and, in some cases, the specific chip or component that's responsible.

(2) Disk based diagnostic and assurance tests for further and more extensive checks of the basic hardware plus elaborate standard and optional analysis of peripheral cards and sub-systems.

APPLEsurance II suggested list prices:

Complete System with Disk Controller Card and Disk Based Software . . . . . . . . . . . . . **$150.00**

Disk Controller Card with Power-Up Diagnostics only . . . . . . . . . . . . . . . . . . . . **$125.00**

Disk Based Software only . . . . . . . . . . . . . . **$50.00**

## II. GRAPHITTI. . .

**GRAPHICS/PARALLEL PRINTER CARD**

Sophisticated text and graphic routines available via simple keyboard commands. Avoid "squashed" printouts. GRAPHITTI on-board firmware lets you utilize the Apple high resolution graphic screens – circles are printed as circles. Use with most of today's popular dot-matrix printers. Graphic support for Epson, Okidata, NEC 8023, C-Itoh Prowriter, and IDS Prism. Compatible with most software packages including PFS:Graph,™ CP/M,™ and Apple Pascal.™

Firmware 'text' features include: Set Line width via standard ^InN format, Enable or Disable Line Feed after Carriage Return, Set Lines per Page, Set Left Margin, Set or Clear 8th Bit. Print High Resolution Page 1 and/or 2, Double Size, Emphasized Mode, Inverse Mode, Rotate 90 Degrees, Specify Left Margin. . . and more.

GRAPHITTI Graphics/Parallel Printer Interface Suggested list . . . . . . . . . . . . . . . . . . . . . . **$125.00**

## III. VERSAbox. . .

**THE ONLY INTELLIGENT SPOOLER/BUFFER WITH AN OPTIONAL REAL TIME CLOCK INTERFACE DISPLAY**

For use with all Centronics parallel and RS-232C INPUT/OUTPUT interfaces. VERSAbox's high-speed input combined with up to 60K buffer (more with character compression) frees up your computer from the slow job of printing. Modems and other compatible devices can also be buffered. Multiple software selectable outputs permit you to choose between two or more VERSAbox outputs or use them simultaneously. You can buffer multiple inputs simultaneously too.

Add the optional REAL TIME CLOCK to VERSAbox and you get a six digit LED numeric time display. Your computer can read the time via the RS-232C port. Another useful option is the SWITCHED 120V SOCKET that can turn external devices on and off by clock control.

Cabinet is 11"W, 8"D, 8"H. VERSAbox suggested list price with 16K buffer **$249.00** Options extra.

## IV. VERSAcard. . .

**FOUR CARDS ON ONE + TWO NEW OPTIONS**

NOW! The four separate card functions – with true simultaneous operation – that made VERSAcard the preferred multifunction card for Apple II – (1) Serial Input/Output Interface (2) Parallel Output Interface, (3) Precision Clock Calendar, and (4) BSR Control – PLUS NEW OPTIONS: Screen Dump Graphics ROM and BSR Time and Event Program.

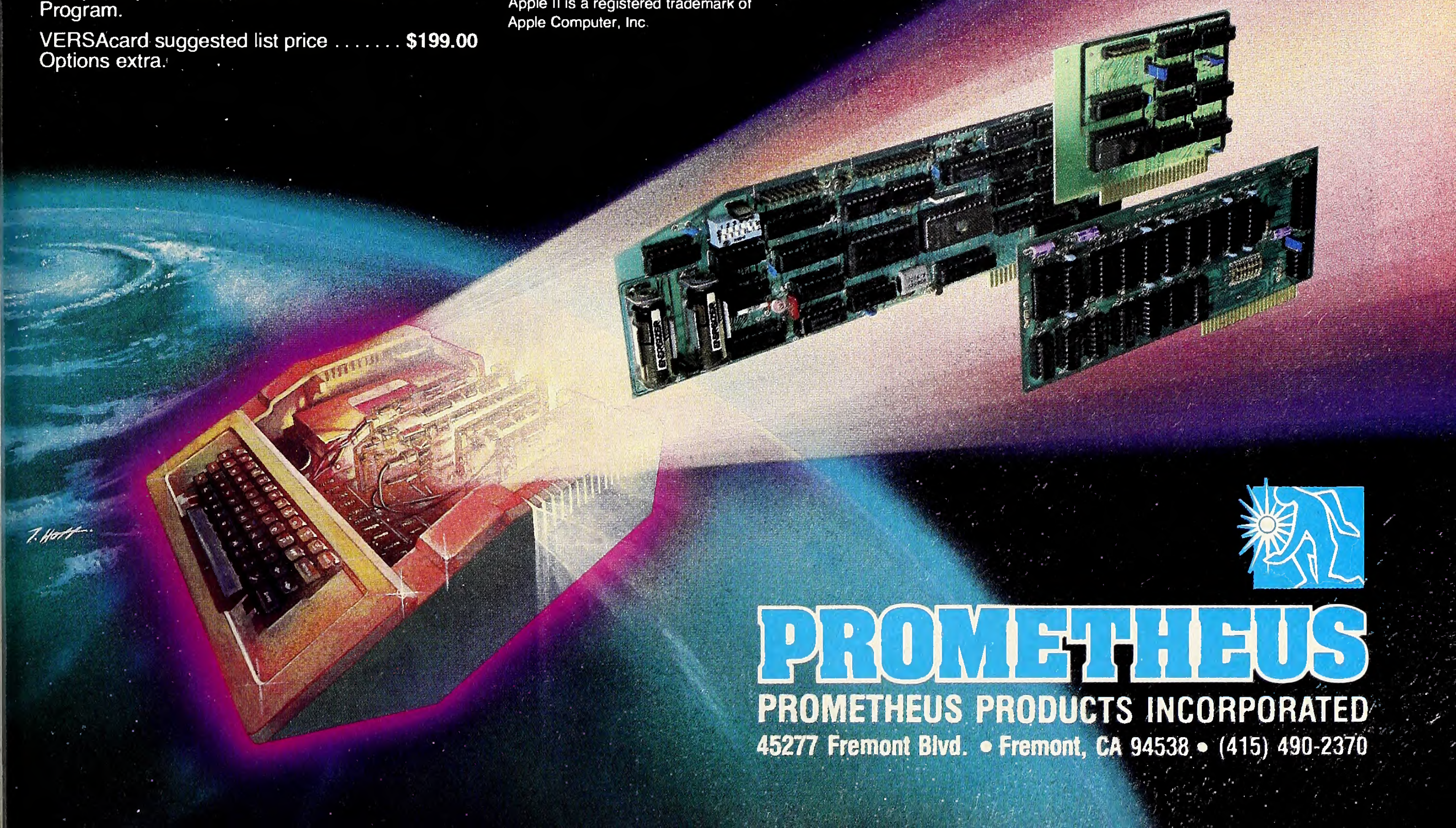
VERSAcard suggested list price . . . . . . . **$199.00** Options extra.

## V. EXPAND–A–RAM. . .

**16K to 128K EXPANDABLE RAM CARD**

A low cost solution to future larger memory needs. Start with 16K (**$225**), 32K (**$249**), or 64K (**$375**), and expand to 128K later. Or, simply order the full 128K size now (**$495**). No Apple modification required. All cards include memory management software, ram test, and utility functions. Disk Emulators for Apple DOS,™ Apple Pascal,™ and CP/M™ available. Optional: Expand Visicalc™ to 128K and 80 column format.





PROMETHEUS

PROMETHEUS PRODUCTS INCORPORATED
45277 Fremont Blvd. • Fremont, CA 94538 • (415) 490-2370

ter was larger than it really was and might try to assign a very long String value to it, overwriting any important information that had the misfortune of lying next to the actual String parameter. In the case of a mismatch, the compiler will issue error message #175("Actual parameter max string length < var formal max length"). For value parameters, the dynamic length of the actual parameter cannot be greater than the physical length of the formal parameter or a String Overflow error will occur during execution.

**String or Char?** Single character constants and literals may be treated by the compiler as data of type String or Char, depending on the type of the variable to which they are assigned. For instance, A would be viewed as a Char datum in the assignment "ChVar := 'A' " provided that ChVar is of type Char. Elsewhere in the same program A would be treated by the compiler as a String of length 1 in the statement "StrVar := 'A' " assuming StrVar is of type String. In "Ord('A')" the constant would be seen as a Char, but in Concat(A, B) both single-character constants would be taken as Strings.

For another example of the dichotomy of Chars versus single-character strings, note that regular array indexing may be used to access individual characters of a String as Char values. Thus, if S contains *This is a String*, then S[4] is the Char value s. However, Copy(S,4,1), although also the value s, is considered a String.

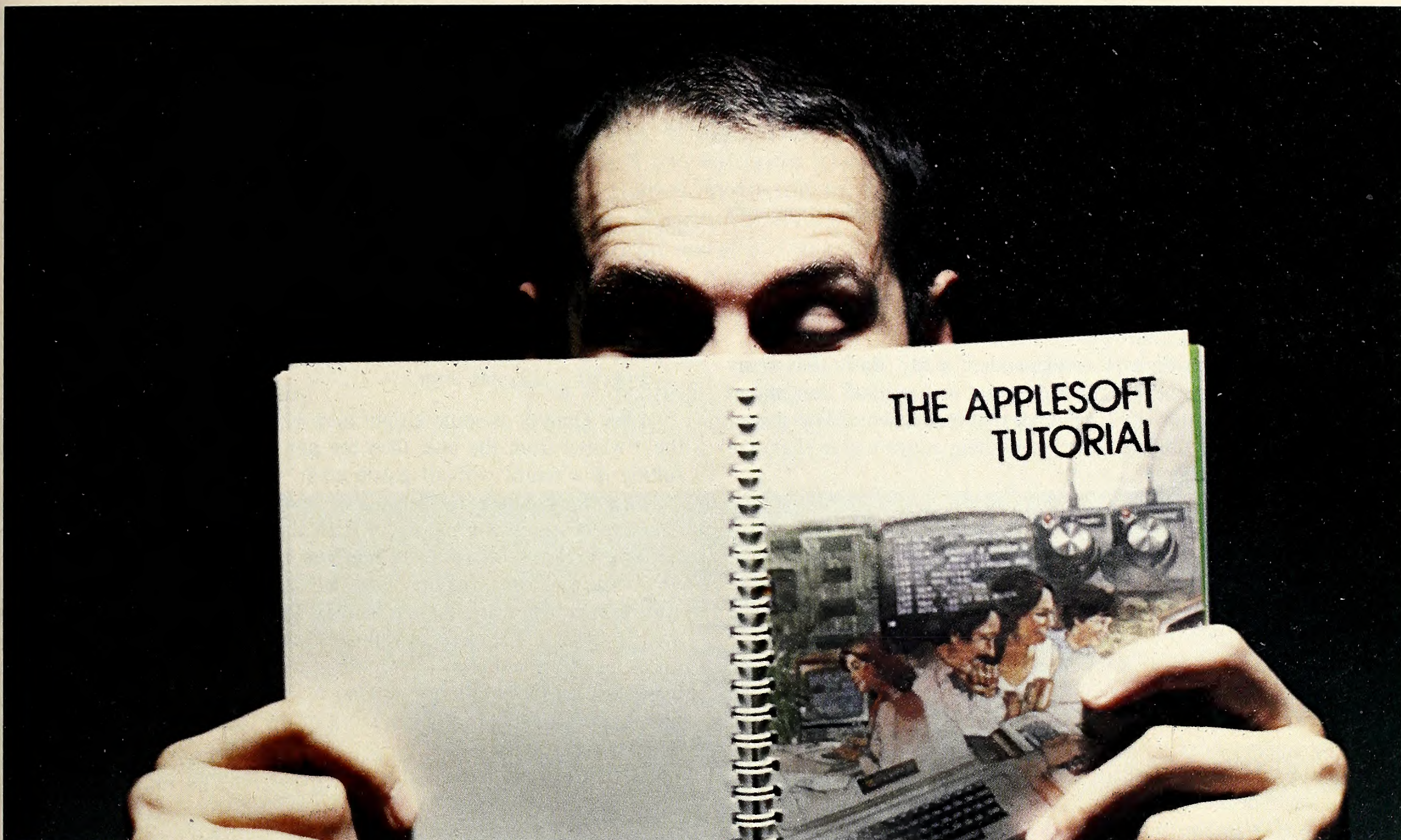
The ambiguities between Char and String values confuse many beginning programmers. It is tempting to consider Char values and single-character Strings as equivalent, but they are not, and one should never be used in a context that requires the other. In *Cable*, and subsequent programs that use Strings, you'll be shown various spots where you can sidestep (and sometimes even exploit) the strange relationship between these two data types. As a matter of fact, we'll do so several times in developing DisplayTown and ChangeTown.

**Looking under the Tree.** Now that you've filled your head with Pascal knowledge for another month, you deserve to fill up on that holiday dinner, so go to it! After your meal, nestle cozily by a warm fire (or a warm computer if you have no hearth) and prepare for next month by thinking about how you'd implement the following data conversion routines:

```
PROCEDURE
  IntToString(Source: Integer; VAR Dest: String;
              MinFW: Integer; LPad: String);
(* Build the character-string representation of
   decimal Source, such that it contains at least
   MinFW characters. Pad on the LEFT using Pad
   string, if necessary to achieve the Minimum
   Field Width, MinFW. Concentrate the final,
   padded representation of Source onto the right-
   hand end of Dest. *)
FUNCTION
  GoodInteger(VAR S: String;        VAR SP:
              Integer; VAR Dest: Integer)
    :Boolean;
(* Return True if character sequence in S, starting
   at position SP, represents a valid decimal
   Integer (ignoring leading blanks). If so, SP
   becomes SP + (length of sequence), and
   Dest acquires the value of corresponding
   Integer. On False return, SP and Dest remain
   untouched. *)
```

Right away, you can see that IntToString and GoodString bear striking resemblances to our old friends IntegerOutput and IntegerInput; indeed, the clever reader should be able to rewrite those two earlier routines to the new specifications without much difficulty. A couple of new wrinkles are evident in IntToString, though: the concepts of *minimum field width* and *left-padding*. These were included in the specification in order to permit IntToString to mimic Write and WriteLn's output formatting capabilities. However, where Write and WriteLn "pad" on the left with blanks only, IntToString can use any pattern specified by the LPad parameter.

If you wish, you might want to try writing IntToString and GoodString on your own, before "official" versions are presented next time. (No fair using the built-in Str procedure to make quick work of IntToString!) These two routines will form the heart of our address conversion facilities and will be made global to the *Cable* program, for the same reason that the function Capital was: to facilitate their use in several different routines. An updated version of *Cable*, including the global IntToString and GoodString as well as all routines we've developed so far, will be presented at the start of next month's column.




# So You Think You Know Applesoft!

## BY DAVID WISEMAN

So, you think you know how to use an Apple. You've been reading All About Applesoft for ten months, you've keyed in all the programs, maybe you've even taken a few courses in Basic programming. Maybe you even teach courses in Basic programming. Whatever your level of expertise, you may be surprised to find out what you don't know. We're all still learning.

Even short, simple Applesoft programs can contain syntax puzzles. So we present a brief exercise for the Apple student. Some of Applesoft's oddities are collected in this quiz. These are not Applesoft bugs; each is fully explained in the Applesoft manual.

Applesoft is the ultimate judge of this quiz. If the answers do not satisfy you, try the program on your Apple and see for yourself. The quiz programs are shown exactly as they should be entered. Applesoft will parse the lines according to its own rules. If you list the program, you will see how Applesoft has grouped the letters into reserved words. The results are often surprising. Take the quiz and see how you rate. Are you really the Applesoft heavyweight that you think you are?

1. This program gets a character from the keyboard and displays its ASCII value. Will it work for all keyboard input?

```
10 GET A$
20 PRINT ASC (A$)
30 GOTO 10
```

2. The following program accepts numeric input from the keyboard and prints it out. If an "A" is entered instead of a number, a "?reenter" message will result. Are there any nonnumeric characters that will be accepted?

```
10 INPUT A
20 GOTO 10
```

3. One of the four lines in this program produces a syntax error when the program is run. Which line is incorrect?

```
10 PRINT""""
20 PRINT"";
30 PRINT E""+
40 PRINT A0E;
```

4. Which line in this program produces a syntax error message?

```
10 PRINT ROTTEN
20 TABS = SCRN (MID,DLE)
30 LETTUCE = CABBAGE
40 SCALES = FISH
```

5. Which lines of this program contain syntax errors?

```
10 IF TUBA THEN 20
20 GRIND = COFFEE
30 REM SLEEP: FOR EVER
40 VAL ("E - E")
50 ? A
```

6. Each of these lines produces an error message. What are the messages that will be displayed?

```
10 A$=2
20 GOTO 60
30 IF "B" THEN POP = WEASEL
40 HIMEM: 0
50 CONTENTMENT
```

7. Here are three one-line programs. Describe what happens when each is run.

```
10 FOR X% = 1 TO 5 : PRINT X% : NEXT
10 GOSUB 10
10 POKE 33,0
```

8. Match up the items on the two lists below. Match the memory locations in the lettered list with the functions in the numbered list.

| | |
|---|---|
| 1. Speaker | A. −16287 |
| 2. Enter Monitor | B. 37 |
| 3. Button zero | C. −16336 |
| 4. Cursor vertical | D. −936 |
| 5. Clear screen | E. −151 |

9. Describe the actions of the following immediate mode commands.

```
DATA "THIS","AND","THAT"
RESUME
STOP
DIM PR (LEN ("BONE")),FR$ (PR(0))
```

10. Here is an opportunity for independent study. Basic uses many English words as commands. Write a program in Applesoft that makes sense in English. Can you write a one-liner that makes sense? Will it run? You will probably be able to write something more clever than the simple examples that follow.

```
10  STOP : WAIT A,MINUTE,MR POSTMAN
20  STOP : WAIT SO,PATIENT,LY

10  HOME : HOME : ON THE RANGE GOTO 10

10  IF BAD ENGINE THEN RECALL MY CAR
```

**The Answer Key: Don't Peek!** Here are the quiz answers. We work on the honor system, so you can grade yourself. Score 10 points for correctly answering each question. Give yourself partial credit where applicable.

1. The get command halts the program until a single key or a key combination is pressed. The variable A$ is assigned the value of the key that is pressed, with one exception. Control-shift-P creates a null string, a string with no characters. Then ASC ("") gives a value error halting the program. If line 20 is changed to:

```
20  PRINT ASC (A$+CHR$(0))
```

the program will work correctly. A control-shift-P will print zero. Note that typing control-C will not stop this program when it's waiting for a keyboard input.

2. The input command peculiarities make foolproofing program data input difficult. Let us limit our discussion to single character input. The input command graciously accepts the numbers 0 through 9. In addition, several other characters will be accepted as valid numbers:

The colon (:) and comma (,) both produce an "extra ignored" message. A zero is put into the variable A.

E is accepted. Again, a zero is put into the variable A. Applesoft assumes that this is an abbreviated exponential entry.

The plus (+), minus (−), period (.) and right arrow or control-U are all acceptable. Applesoft assumes that these are numeric zeros.

3. Line 30 is the only line with a syntax error. The ending plus sign is the offender. Line 10 will space one line regardless of the number of quote marks that follow. Note that in line 20, the period prints as a zero.

4. Line 20 has the only syntax error in this program. It is not the tab combination that causes the problem. Tab must be followed by a left parenthesis to be parsed as an Applesoft command. Here Applesoft is attempting to use the ABS function (absolute value). Line 10 contains the reserved word rot, but this is only reserved if it is followed by an equal sign. Line 30 will assign the value of Cabbage to Tuce. Let is a noise word that Applesoft recognizes as part of a value assignment statement. Finally, line 40 contains the word scale. Like rot, this is a hi-res graphics command only when followed by an equal sign.

5. Let us take the lines one at a time. Line 10 contains a common error. The A in tuba and the T in then combine and parse as the word "at." This creates a syntax error. Variable names ending in A are bad news.

Line 20 will produce a syntax error. The GR in Grind will put you into lo-res graphics, and Applesoft will not make sense out of what remains.

Line 30 is fine. Once a rem is encountered, the remainder of the line is treated as a remark. "For ever" is part of the remark.

Line 40 also produces a syntax error. Applesoft can readily evaluate the line, but it has no instructions on what to do with the value. There would be no problem if we had written:

```
10  A = VAL ("E − E")
```

or

```
10  PRINT VAL ("E − E")
```

Line 50 has no problems. The question mark is another way of writing print. If you list this line, the question mark will display as print.

6. Here are the errors for each line:

| | |
|---|---|
| Line 10 | Type mismatch error |
| Line 20 | Undefined statement error |
| Line 30 | Syntax error |
| Line 40 | Out of memory error |
| Line 50 | Syntax error |

Line 30 gives us some insight into what Applesoft does. If we drop the =weasel from the line, then the pop is executed by Applesoft, resulting in a return without gosub error.

Line 50, of course, contains an embedded cont command. When cont is executed within a program, it halts the program, and Applesoft appears to hang up. A control-C returns control to the operator.

7. Each of the three one-liners has a problem. The first produces a syntax error. The variable in a for-next loop must be a real variable; the integer variable X% is unacceptable. The second quickie program produces an out of memory error. After twenty-five gosubs without a return, Applesoft's internal return stack is full. The third program runs fine. Unfortunately, it sets the width of the text window to zero, which hangs Applesoft.

8. You should have matched the two columns as shown here:

| | |
|---|---|
| 1. Speaker | C. −16336 |
| 2. Enter Monitor | E. −151 |
| 3. Button zero | A. −16287 |
| 4. Cursor vertical | B. 37 |
| 5. Clear screen | D. −936 |

9. Some commands will not work in immediate mode. Data will not generate an error message, but the values on the data line will not be available to any read command. The data line does nothing. Resume is used to return from an error procedure. In immediate mode it has no meaning and may hang the system or generate a meaningless error message. Stop is a valid immediate mode statement, but it accomplishes little. The word "break" is printed. The final line dimensions two arrays without any problem.

10. Grade this exercise as an essay question. Give yourself one point for each guffaw.*

**Midterm Grades Are In.** Now that you have completed the quiz and totaled your score, you must be eager to find out how well you have done. Here are the ratings:

| | |
|---|---|
| 100 − 85 | Microcomputer Guru |
| 84 − 70 | Experienced Bit-Fiddler |
| 69 − 55 | Normal Norman |
| 54 − 40 | Neophyte |
| 39 − 20 | Ph.D. *Space Invaders* |
| 19 − 0 | Danger to Humanity |

*Bob Wiseman has owned an Apple for three years. He teaches an introduction to personal computers class and has occasionally thought of himself as an Applesoft expert. He credits the oddities presented in this quiz with keeping him humble in that respect.*

***Bonus Contest: Softalk Literati, a Parse-Fail Course in Programming.** A prize of twenty-five dollars toward the purchase of any of our advertisers' products will be awarded to the best program of any length that meets the requirements of question 10. Each entry must be a legal Applesoft program that can be read as grammatical English before Applesoft parses it. Please observe the rules concerning variable names and line lengths. These programs must be able to be typed and run without syntax errors. Judgments will be based on the originality, length, and inherent humor of the entries. Extra credit will be awarded if the program actually does something interesting.

Mail entries to Softalk Lit, Box 60, North Hollywood, CA 91603.

# Word Processing:

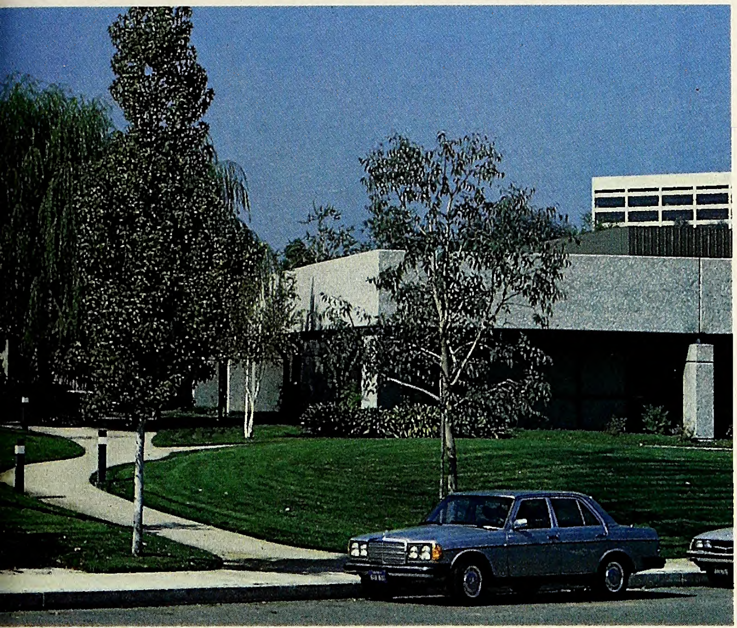
## BY JONATHAN MILLER

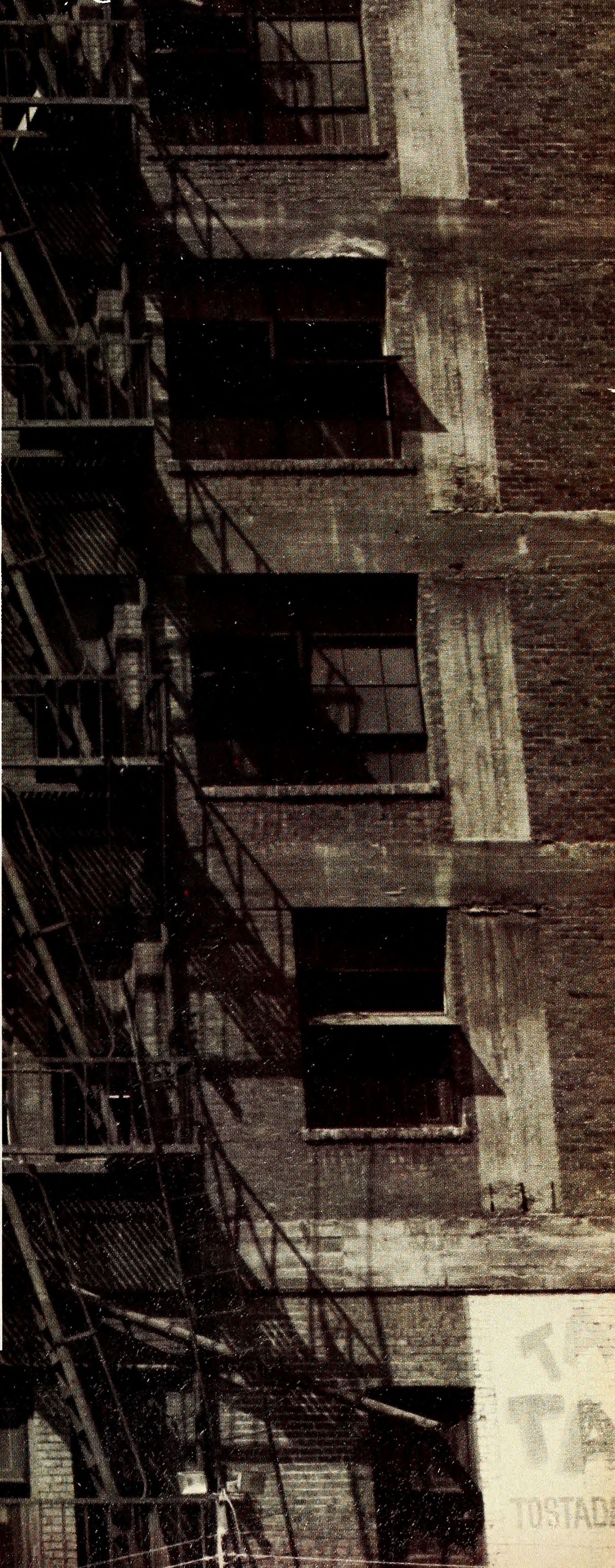
Whenever energy manager Robert Bonoff wants to explain what he does for a living, he resorts to a parable. He calls it the parable of the pails.

A person occupied a large room, Bonoff relates. In the room were 2,880 empty pails, representing the number of quarter-hours in a thirty-day month. It was raining outside—a deluge that was to make a run at the Biblical record of forty days and forty nights.

To amuse himself, the person in the room pretended that each raindrop represented a kilowatt of electricity. Every fifteen minutes, he opened a window and placed one of the pails outside. When a quarter-hour had elapsed, he retrieved the outside pail and replaced it with another. At the end of a month—he had planned for this vigil, of course, by stocking in some good books, his Apple, and assorted junk foods—he was sitting in a room of buckets that were no longer empty.

From the side the buckets looked the same—standard high-tech galvanized gray. From the top, however, it was a different story. Because the rain had fallen at different rates during each fifteen-minute interval, the water levels in the pails were different. Some of the pails





# Super-Text Energy

were nearly empty, others half-full. One pail demonstrated the difference a drop makes. It was much closer to being full than the others.

**Data Bucket Management.** For all the water he collected, our friend paid a price per drop. Then he paid another fee based on the amount of water in that one most nearly filled pail. Had he been able to control the rainfall, to make the water level in all the pails even, he would have paid much less. For one thing, the utility charging the fees could then have provided smaller pails.

The man couldn't control the rain. But he could study the rain and change buckets more often when the fall was heavy, less often when it was light.

Which just goes to show that he who manages his buckets wisely shall reap a pail of profit. That one full pail represents the typical commercial user's peak demand, the greatest usage of power during a fifteen-minute period in any given month, for which the user pays a premium. Some utilities may use a thirty-minute or hour-long demand period in their rates, says Bonoff, but all of them charge their commercial customers for two things—units of usage and highest usage. In other words, there's a special rate for the most filled bucket and another rate for all the buckets.

The rationale behind the demand charge is simple enough, Bonoff

# HI-LIGHT YOUR GRAPHICS!

Bring your Apple® graphics programs out of the dark ages with the help of Avant-Garde Creations' enlightening **HI-RES SECRETS** programmer's utility series! These two information-packed systems reveal everything you need to know about state-of-the-art Apple® hi-res graphics programming and more. Both are stand-alone packages, but together they make an unbeatable combination!

## NOW IN THE SPOTLIGHT:

**HI-RES SECRETS GRAPHICS APPLICATIONS SYSTEM (G.A.S.)** is a complete utility that takes you step by step through the answers to all of your specific Apple® graphics application questions including: how to turn fair BASIC programs into good BASIC programs, translate BASIC programs into machine language programs, how to make business graphics, architectural and electronic designs, 3-D designs, marketable quality arcade and adventure games and more. **G.A.S.** also contains an exciting new super-fast color-filling Palette program that has 140 gorgeous colors, 160 different patterns, 4 separate fill algorithms and the program even allows you to fill on both black and white backgrounds! **HI-RES SECRETS GRAPHICS APPLICATIONS SYSTEM** includes 3 unprotected disks plus an extensive manual at a special introductory price of only $75.00. Available separately, **HI-RES ARCHITECTURAL DESIGN** and **HI-RES ELECTRONIC DESIGN** retail for $29.95 each.

## HAVE YOU SEEN THE LIGHT?

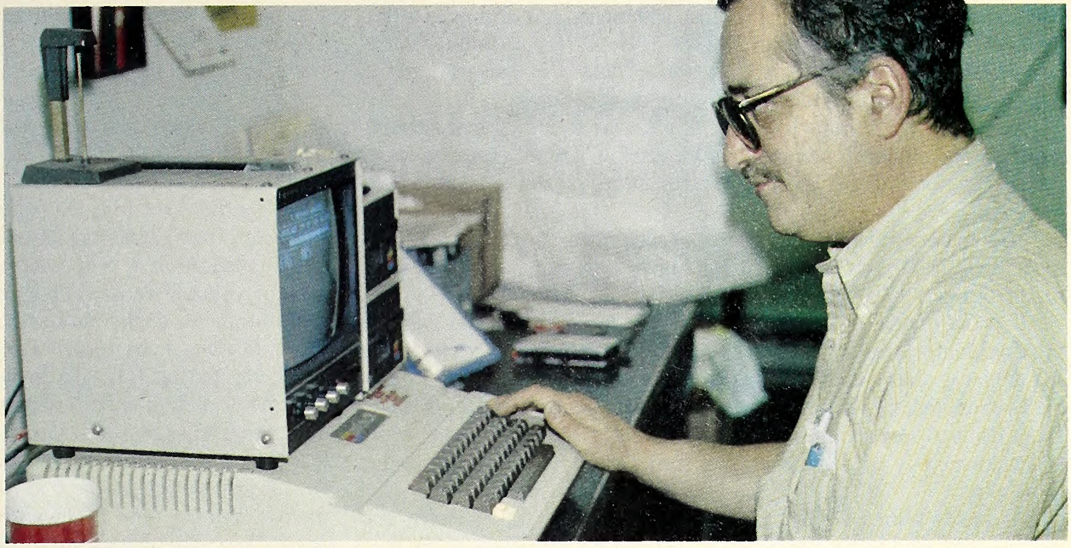
**HI-RES SECRETS** is Avant-Garde Creations' original best-selling programmer's graphics utility. From block, vector and hplot shapes to animation, type font, music and color-fill programs, **HI-RES SECRETS** reveals all! From novice to expert, this indispensible graphics utility system is a must for every programmer's library. **HI-RES SECRETS** contains 4 disks with a 263-page manual and retails for $125.00.


No permission or royalties are required for the use of **HI-RES SECRETS** or **G.A.S.** routines in your own programs. All 3 systems run on Apple II 48K DOS 3.3 and are available at your local dealer or you may order direct from Avant-Garde Creations.



AVANT-GARDE CREATIONS

**AVANT-GARDE CREATIONS**
P.O. BOX 30160 • EUGENE, OR 97403 • (503) 345-3043
*BUSINESS, EDUCATION, UTILITY AND ENTERTAINMENT SOFTWARE*



With the help of his Apple II Plus and *Super-Text*, energy manager Robert Bonoff prepares proposals in hours that used to take days to prepare by hand.

explains. The utility has to have the capacity to meet extraordinary as well as ordinary demands, which means they have to invest in sufficient generators to supply the peaks. It seems only fair, they argue, that those who impose this additional capital burden on the system pick up the tab—and that tab's a charge that can represent up to a third of a company's monthly power bill.

**Baby, It's Cold Inside.** Which brings us back to our protagonist, your relationship to him, and the moral of this tale. To see dramatic decreases in your monthly power bill, you have two concerns—overall usage and peak usage. To keep costs down, you must find a way to even out your usage demands. Enter the energy manager. Bonoff is the forty-eight-year-old founder and president of the Energy Savers Group, of Stamford, Connecticut. The name—picked from among combinations randomly spewed out by his Apple II Plus—conjures up images of placard-waving consumerists, but it actually refers to Bonoff's $300,000-a-year energy management company. Unlike the typical news-at-eleven meteorologist, Bonoff actually does something about the weather—at least inside buildings. He installs sophisticated microprocessors and programmable controllers that can manipulate up to seventy different electric load factors in commercial buildings and factories. The trick is to do more than smooth out the peak loads while lowering overall usage—it's to deliver promised reductions in both areas so that no one is really aware you did it.

"A good energy system should be totally transparent to the people in the building," Bonoff explains. "If people become uncomfortable, they'll start to override the system and defeat the whole thing."

On a simplistic level, the way to manipulate loads for fans, lights, water heaters, air conditioners, pumps and compressors, to name the most obvious juice guzzlers, is to shut off one or more of them. On that plane, energy management is your basic binary on-off proposition—unless you leave such piddling details to a human, in which case the air conditioners that were supposed to go off at five o'clock are still humming away at eight the next morning.

The binary proposition gets considerably more complicated when you're also trying to flatten demand peaks. If that is also the objective, you have to rely on a sophisticated program that produces a minimum number of simultaneous equipment starts.

"When we cost up a job," says Bonoff, "what we're really talking about is what it costs us per load or group of loads to control."

**Micros Don't Take Holidays.** Generally speaking, the most profitable deal for him is an all-electric factory-office complex that isn't going twenty-four hours a day. The environment presents more problems in one sense, but it also yields many more solutions because you have more load-factor combinations to play with. Such a challenge is made for the microprocessors and programmable controllers Bonoff uses to combine and manipulate load factors.

"What these things are essentially are computers that interface with the real world," he explains. "You can make them do virtually anything you want."

In a typical building, says Bonoff, he's dealing with perhaps sixty to seventy load factors. His equipment not only turns lights on and off at prescribed times, it remembers to shut the system down on the Fourth of July when no one shows up for work. A school district near Bonoff's adopted home of Stamford gloats about its energy-management foresight every time a snowstorm forces the superintendent to cancel classes. Because the schools are under microprocessor controls and linked via modem, the superintendent can turn the system off by dialing a single number.

Short of holidays and retooling furloughs, Bonoff isn't dealing with total system shutdowns. He's optimizing. He's calculating—factoring in historical energy use patterns and data from strategically placed sensors inside and outside the building to determine the latest possible moment to turn off the air conditioner, the lights, the heat. He is creating a subtle new environment. The building is alive, waking up and turning in at different times every day. And during those days, he's manipulating the on-and-off cycles of water pumps and air conditioner compressors, an energy-management technique known as duty cycling.

According to Bonoff, it's been found that a compressor actually works more efficiently (draws less electricity) if it is made to work harder—by, for example, being deliberately turned off for, say, ten minutes every half hour.

"Instead of cycling on and off five or six times as it would under thermostat control," says Bonoff, "it will maybe only run and cycle off two or three times."

If at the same time, you engineer different recycling schedules for multiple compressor units, you can further increase savings by reducing the demand charge that is linked to simultaneous turn-ons.

In some cases, says Bonoff, an energy manager will arbitrarily establish a demand target of, say, 15 to 20 percent of historical usage, which will automatically turn off noncritical load factors if that energy goal is threatened.

"You can almost always turn off an electric hot water heater, and nobody will miss it if it goes off for fifteen minutes," says Bonoff.

There are other times when a target-minded energy manager will cause a little discomfort, such as when he shuts off the air conditioning in a building a half-hour before it empties at rush hour. But a well-designed system always allows for overrides of the target goal when it threatens critical load factors. The trick is to make sure a minor inconvenience doesn't become a major discomfort.

**Too Cheap To Be True.** If Bonoff takes on a job, he says he usually can cut a company's monthly fuel bill upward of 15 to 20 percent. For Burndi, a leading manufacturer of electrical connectors, the savings ran in the neighborhood of $8,600 on a system cost of $58,000. Generally speaking, Bonoff is able to ensure a full payback on the installation within eighteen months to two years. That would certainly seem grist to the mill for bottom-liners, but Bonoff first has to get his foot in the door—literally. "Most people just don't believe you can save them that kind of money," he says.

To overcome such resistance in post-OPEC-crisis America and get by that door, he offers his prospective clients that time-honored no-cost-or-obligation free survey, along with a copy of his client list. All they must do is have someone show him around the facility and furnish him with fuel bills for the past fifteen months.

Once he has determined what load factors are controllable through his on-site inspection, he pores over the fuel bills, extrapolates the energy-use patterns, and—*voila!*—he has a proposal to make. It's your basic, no-nonsense, fourteen-page contract, with all these boiler-plate paragraphs and options for this and that combination of load factor refinements. An absolute time-busting horror that, in the company's not-so-distant founding four years ago, took Bonoff the better part of three days to prepare by hand.

Until, in the dawning of a better idea, he discovered word processing and *Super-Text*, ranked among the leaders of *Softalk*'s word processing top ten.

**They Yawned When He Sat Down at the Keyboard.** In the beginning, it was just Bonoff, his typewriter, a programmable Texas Instruments calculator, and a secretary.

"I'd write the letter by hand, then she'd type it up rough. I'd make ten million corrections and then she'd type up a new rough draft and I'd make more corrections."

Well, you get the idea. The man was making a lot of unnecessary work for himself. Here he was in the business of conserving energy and he was wasting his. It wasn't long before he realized there had to be a better way.

Bonoff's first move toward word processing was to contract with a local computer service bureau. Bonoff still had to do his figuring on his T.I. calculator, but now he could manipulate paragraphs like load factors on the bureau's mainframe word processor. What had taken him three days was now only taking an hour. It was a vast improvement over his Stone Age ways, but it, too, had its shortcomings. You had to pay for the computer time and on-line storage, so Bonoff started investigating the possibilities of having his own terminal.

"At least I wouldn't have to leave to go over to the bureau."

Owning his own computer was the furthest thing from his mind when, in late 1978, he punched a magazine bingo card for this strange-sounding company "with a name like a fruit. What the hell was an Apple?" Company literature that answered that question was dutifully filed away, but early in 1980 it was read in earnest as Bonoff took the plunge and bought an Apple II Plus.

When he dived into the software end of the pool, he first went for *VisiCalc*, a half gainer that saved him another, shall we say, fruitless day in the preparation of proposals. His second foray was into word processors, but Bonoff frankly admits he probably didn't make as many comparisons as he might have. He was familiar with Data General and Wang office systems, tested *WordStar*, *Apple Writer*, and *Super-Text* in the store, and read all that he could on the others. He basically knew what he wanted in a word processor—global search and replace and an ability to patch and paste—but in the final analysis, there was no one feature that tipped the scales in *Super-Text*'s favor.

"It just seemed to me that *Super-Text* could do more for the same money than the others," says Bonoff. "All I can say now is that I can make *Super-Text* stand up and go through hoops."

**Some Frills Are More Basic than Others.** Bonoff freely admits he is no software freak. When he decided on *Super-Text*, he did so with the attitude that he could always buy another moderately priced program if it didn't pan out. What he was looking for, and what he feels he found, was something as easy to use as his typewriter. Instead of hitting a tab key to indent a line, now he simply had to strike a control key.

He became so comfortable with the program that when Muse Software sent him an enhancement allowing for eighty-column as well as forty-column display he passed on buying the Videx card enabling him to use it. "I guess it would be nice," he reflects, "but to me that's just a frill."

It's a frill because he can look at his forty-column display, he says, and know pretty much how it will appear on the printed page. Besides, if he doesn't want to rely on his educated eye, he can always fall back on another program update—a preview screen, in the print mode, that permits him to see the left and right margins of a page and reformat accordingly.

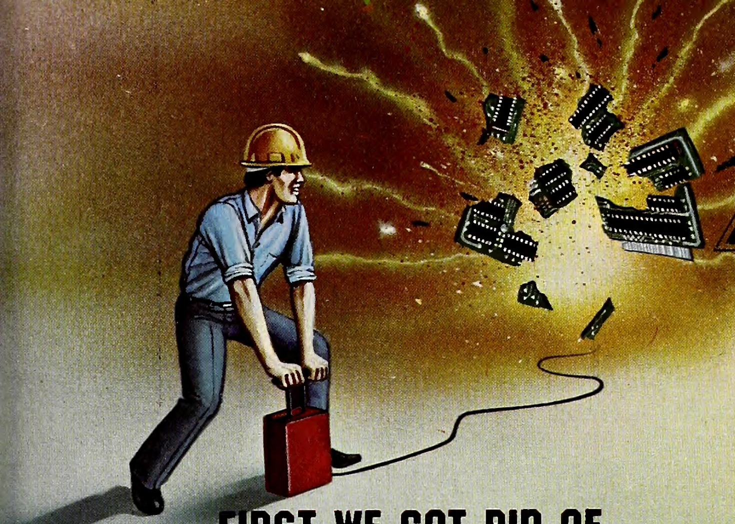
Speaking of screens, another useful feature that bewitches Bonoff is the ability to split his monitor. He finds this particularly useful because he can put one part of a file page on the upper part of the screen and display a related section occurring four pages later in the file on the lower half.

But the basic beauty of the program lies in the ease with which it permits him to whip off "a terrific letter" or knock out fourteen-page proposals at a clip of three a week. And the key to that accomplishment is another program adjunct, a Form Letter module, whose name Bonoff regards as a very poor description of such a powerful word processing tool.

"You can create any kind of document you want—a report, or a letter—with as many variables in the body of the file as you want," he says admiringly.

What the program permits a user to do is personalize form letters and documents of all types. Which means he can punch in the name and address of the form letter recipient; refer to that individual by name in the body text; adjust the content for particular occupations so Dr. Jones is schmoozed as a member of the medical fraternity; use prompt statements to insert data directed at one individual in a group mailing; include extra text for special subsets of a mailing list (like complimenting Sunbelt residents on their wonderful weather), along with the sine qua non of form letter packages—formatting and printing the envelope.

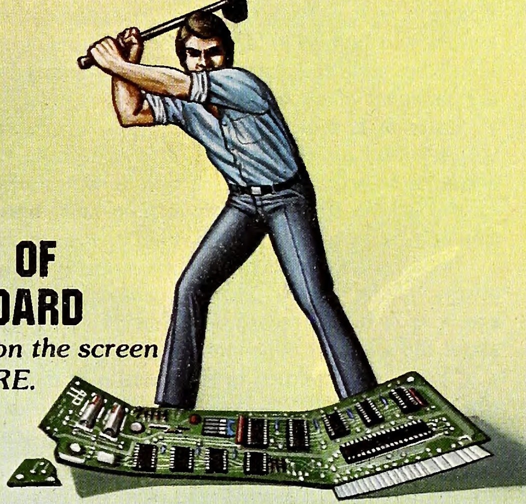
The personalizing touch comes into play for Bonoff when he's fashioning a proposal that can have as many as a half-dozen options. One of these would be interfacing a microprocessor with an Apple via a modem to generate a print-out of energy-use data at specific times or to, say, dial a maintenance man at home. So when the monitor asks Bonoff what options are to go into the proposal to T. Jackson Witherspoon of North American Widget Company, the program knows enough to execute page breaks regardless of which ones he selects. Concludes Bonoff: "It's ideal for any businessman who wants to print out proposals that aren't exactly the same."



# ScreenWriter II

**FIRST WE GOT RID OF THE LOWER CASE ADAPTER**

*and put lower case on the screen WITHOUT HARDWARE.*



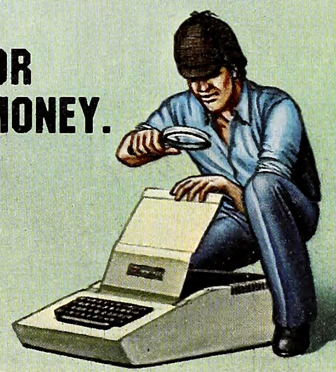
**THEN WE GOT RID OF THE 80 COLUMN BOARD**

*and put seventy columns of text on the screen WITHOUT HARDWARE.*

**THEN WE GOT RID OF WASTED EDITING TIME**

*by developing printer spooling AGAIN WITHOUT HARDWARE!*

**AND WE'RE ALWAYS LOOKING FOR NEW WAYS TO SAVE YOU TIME AND MONEY.**



When SIERRA ON-LINE INC. began work on SCREENWRITER II, we wanted to make it the most powerful word processor on the market. We began by putting in boldfacing, underlining, form letter capabilities, global search and replace, support of multiple disk drives, full macro capabilities, generation of up to four indices, complete formatting capabilities, a software based keyboard buffer, full editing capability of any DOS 3.3 text or binary file (including APPLESOFT programs) and all other features you need for professional word processing. That was just the beginning. . . .

Our next step was to make SCREENWRITER II hardware independent, so we eliminated the lower case adapter and the 80 column board. SCREENWRITER II is the only word processor on the market to offer true software based lower case and 70 column screen available today.

The move that put us over the top was the introduction of printer spooling* which allows you to do your editing while printing, making your word processing time more efficient.

With SCREENWRITER II, we've added all the features you need, did away with all the problems you can do without, and we did it all at a price that you can afford-$129.95.

*Printer spooling feature only available for printers with APPLE parallel or SSM AIO interface, and EPSON printers/APPLE II and APPLE II Plus are registered trade marks of APPLE COMPUTER INC.

*We Don't Think That Word Processing Should Cost An Arm And A Leg, And At Sierra On-Line Inc., We Won't Stand For It.*

SCREENWRITER II runs on any 48K APPLE II / II Plus with DOS 3.3 and is available now at your local computer store or order directly from . . . . . .

ADD ONE DOLLAR FOR SHIPPING     VISA, MASTERCHARGE, CHECK, C.O.D.

Sierra ON-LINE INC 36575 MUDGE RANCH ROAD · COARSEGOLD, CA 93614 · 209-683-6858

**The Formatting Years.** As Bonoff tells it, he fell into word processing much as he did into low-voltage electronics and energy management—sort of by accident. As a kid growing up in Rockville Center, Long Island, Bonoff was drawn to technical things: erector sets, chemistry sets, the challenge of turning a pile of junk into an elegant piece of hardware. But this hobby didn't suggest a possible career until he was home sick one day with a broken television set. The repairer who came to fix the TV had very narrow skills, yet he was able to make a living—doing what Bonoff had a gift for. If the repairer had been an engineering type expert, Bonoff wonders where his life might have gone; he might never have recognized the potential for himself in this kind of career. Bonoff signed up for a mail-order electronics course from RCA Institute and learned how to fix all manner of household appliances. He was a Hardy boy right there in the Big Apple, running a repair business out of the basement of his parents' home.

Later on, he took a stab at running his own stereo store, but when the industry leapt from hobbyist purity to cutthroat merchandising, he cast about for something new. And kind of by accident, he found himself in burglar alarms. His job performance was such that it attracted the attention of a rival company, which in turn led to the Saga of New Mexico.

Bonoff, who managed an alarm company for his new employer, was always on the lookout for new product lines. Sometime in 1977, an executive in that company sent him an ad for some electronic gadget about the size of a table-model radio.

"You were supposed to be able to plug this gadget into a wall and control more than a hundred electronic operations going on all over a building. It was being advertised as an energy-management tool and I didn't even know what energy management was."

His education was beginning fast.

When he wrote the company for information, he received a dealership application along with the promotional material.

"They were trying to sell pie-in-the-sky franchises," he recalls. But the concept still intrigued him, so much so that when the sudden bankruptcy of his employer deposited him on the street, he decided to investigate the funny radio more thoroughly.

Somehow or other he managed to scrounge up a business partner, hopped a plane to New Mexico where the company was located, and returned with a draft dealership contract and two boxes in hand. Then things went sour again. His attorney declared the contract to be too one-sided. Then the company went bankrupt. Then the equipment blew up.

Hardly a propitious beginning, but Bonoff pressed forward, looking for other companies in the electronics field. He finally latched on to Pacific Technology, in Kirkland, Washington, the largest manufacturer of microprocessor-based energy management systems in the country, according to Bonoff. His experience in New Mexico had scared him off franchising, but a nonrestrictive dealership was something else again. No protected territory, on the one hand, but no restriction on handling other products either.

Today, four years later, he is installing or subcontracting installations on Pacific Technology and Texas Instruments equipment. His client list is growing: the Crest Line leather goods division of Swank, Burndi, Walden Books, and New Medical Associates, operators of a chain of nursing homes. The business has increased tenfold, from $30,000 to $300,000 annually, and his work crew now includes four installers, two salesmen, and a secretary.

**It's Always Fair Weather.** The force, in short, is definitely with energy management. The conservation ethic appears firmly entrenched in the public consciousness at the same time market-oriented utilities, faced with growing overcapacity, are actively promoting all-electric buildings. Load management has become the utility industry buzz word and electric heat pumps (that heat and cool buildings more efficiently than standard equipment) its sexiest product. The greater the conversions and rate breaks for off-peak users, the greater the prospects for Bonoff's business.

So when you pose the standard question and ask where he thinks the industry's going, he comes straight back with an answer. He has a popular product and the word processing power to market it rapidly. Says Bonoff, "The growth potential is tremendous."

# Olivieri's Outline of Word Processors

## by Peter Olivieri

It's time now to begin examining some specific word processing packages. We'll look at two packages this month and at two or three more in each of the next several articles. Then, in a special installment of the series, we'll evaluate each package and include the reactions and experiences users have reported.

It was tempting to print an enormous table with program characteristics listed down one side and the package names across the top—a fairly common approach to presenting this kind of material. But such an arrangement does not make it particularly easy to become familiar with the attributes of a particular package, nor does it allow for the inclusion of special features or unique aspects of a product. So we'll cover each package separately, including such information as the package name, the vendor's address, the package price, the equipment needed, and comments about special features or interesting characteristics.

The comments sections are likely to be especially useful. If you've listed your word processing needs, as we suggested last month, you'll be able to determine how close a given package comes to meeting them. One more bit of advice: when you're ready to buy, look carefully at the commands summaries provided by most packages' documentation. By referring to these, you'll be able to determine what options a package allows and how much effort is required to accomplish a particular task. If, for example, you have to press eight different keys just to erase a single character, it's easy to deduce that the system is not a winner when it comes to ease of use.

**Zardax,** Computer Solutions, Box 397, Mount Gravatt, Queensland 4122, Australia. Action-Research Northwest, 11442 Marine View Drive S.W., Seattle, WA 98146; (206) 241-1645. $295.

Equipment required: 48K, a shift-key modification (hardware supplied), one or two disk drives, monitor, printer.

*Zardax* was designed with the human in mind. Most people can learn to use it in a relatively short period of time. Like most word processors, *Zardax* can handle form letters, contracts, mailing lists, and report preparation.

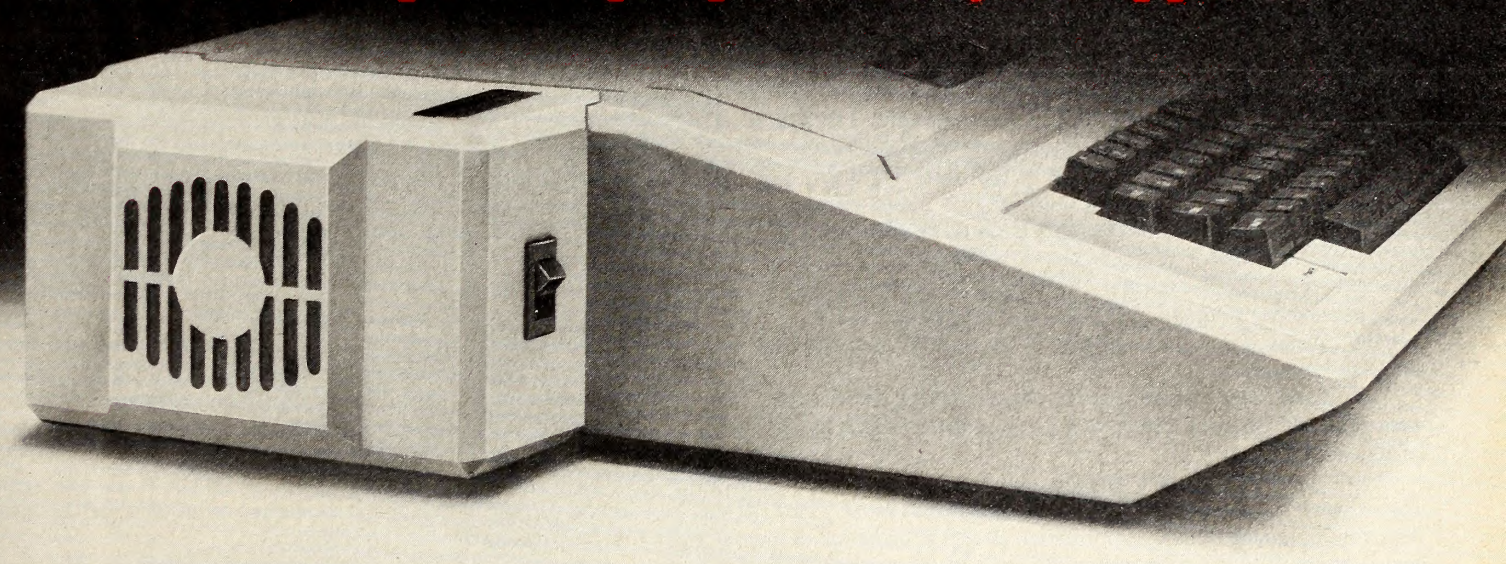
The program gives you a forty-column format that has upper and lower case and on-screen underlining. Since this is done on the hi-res screen, no lower-case chip is required. Optionally, the very same program can be used with a Vision-80, Doublevision, Smarterm, Sup'R Terminal, or Videx eighty-column board to get eighty columns and upper-case/lower-case display. In addition, the program will take advantage of a 16K RAM card.

*Zardax* is compatible with a variety of printers, including NEC 5510, NEC 5515, Diablo, Vista V300, Qume, Sprint 5, Centronics 737, Silentype, and the Epsons, among others. You can control page width, margin size, and justification. You can do single-sheet, continuous-sheet, and multiple-copy printing. Escape or control characters can be sent to the printer to produce emphasized characters and double-width characters or to take advantage of any other special features of your particular printer. If you have a daisy wheel printer, you can use subscripts, superscripts, overstriking, various line-spacing options, and different character pitches.

The program enables list processing. You can create a form letter with certain fields left blank and merge a list of names and addresses with the letter, allowing for personalized communication.

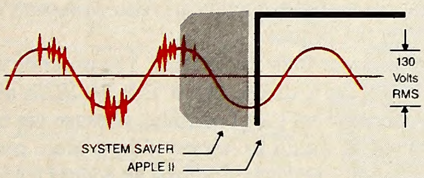
# System Saver™

## The most important peripheral for your Apple® II.



### For Line Surge Suppression

The SYSTEM SAVER provides essential protection to hardware and data from dangerous power surges and spikes.

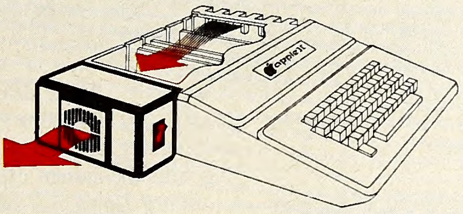



By connecting the Apple II power input through the SYSTEM SAVER, power is controlled in two ways: 1) Dangerous voltage spikes are clipped off at a safe 130 Volts RMS/175 Volts dc level. 2) High frequency noise is smoothed out before reaching the Apple II. A PI type filter attenuates common mode noise signals by a minimum of 30 dB from 600 khz to 20 mhz, with a maximum attenuation of 50 dB.

### For Cooling

As soon as you move to 64K RAM or 80 columns on your Apple II you need SYSTEM SAVER.

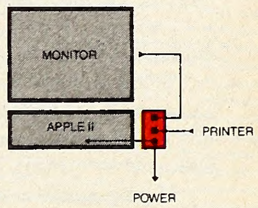
Today's advanced peripheral cards generate more heat. In addition, the cards block any natural air flow through the Apple II creating high temperature conditions that substantially reduce the life of the cards and the computer itself.

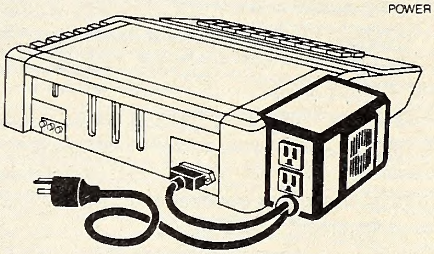


SYSTEM SAVER provides correct cooling. An efficient, quiet fan draws fresh air across the mother board, over the power supply and out the side ventilation slots.

### For Operating Efficiency

SYSTEM SAVER contains two switched power outlets. As shown in the diagram, the SYSTEM SAVER efficiently organizes your system so that one convenient, front mounted power switch controls SYSTEM SAVER, Apple II, monitor and printer.

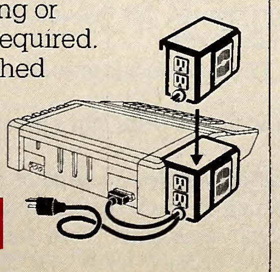


The heavy duty switch has a pilot light to alert when system is on. You'll never use the Apple power switch again!

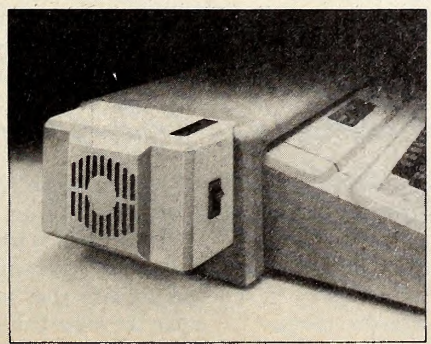
### Easy Installation

Just clips on.
No mounting or hardware required.
Color matched to Apple II.



UL LISTED

### Compatible with Apple Stand



**$89.95 at your local dealer or order direct by phone or mail.**

For phone or mail orders include $2.50 for handling. New York State residents add 6¼% sales tax.
VISA and MASTERCARD accepted.
Dealer inquiries invited.

**Kensington Microware Ltd.**
**300 East 54 Street, Suite 3L**
**New York, NY 10022**
**(212) 486-2802**

KENSINGTON MICROWARE

Documents created with *Zardax* can be up to 13,500 characters long. Longer documents must be treated as separate files (although this doesn't really present a problem).

When you get ready to use the system, you're instructed to "install" the software and hardware. Installing the software means informing it about how your system is set up. Installing the hardware requires a keyboard modification. This must be done inside the machine itself and requires that you remove a chip and attach some clips. This process is easy and painless—and makes your word processor work better. While the procedure for the modification is clearly explained in the manual, some photographs depicting what must be done would also have been nice. In fact, there are no illustrations at all in the user manual.

Despite the absence of illustrations, the user manual is among the best. It's very clearly written and thorough. Well-written documentation is, of course, one of the most important characteristics to look for in a system, because if you can't understand the manual your problems are likely to be many.

The *Zardax* user manual begins with an overview of the system and then guides you through it by means of a tutorial that makes use of many of its features.

The next section contains reference material for using the system; the menus you'll see on screen are given and explained in detail. There's also a section containing technical information about the operation of the system. Listings of many of the printer drivers and other routines are provided in full. If you know a bit about programming, you're likely to find this information useful. At the back of the manual is a tear-out reference card.

Should any questions or problems arise, program users can take advantage of a user hot line. If a disk is damaged and is returned within the first year after purchase, it will be replaced for $15.

The *Zardax* package offers even more nice features. Brief descriptions of some of these follow.

*Transfer.* Documents can be transferred easily from one disk to another (leaving the original file intact on the original disk).

*Screen display.* On the forty-column on-screen display, words are separated by a dot located about mid-character vertically. You might think this would be distracting, but it didn't turn out to be. Carriage returns are indicated by curving arrows on both size displays.

*Moving within a document.* Moving around within a document is easy. To go to the beginning, for example, type control-B; to go to the end, type control-E. Up, down, left, and right are also designated by their logical control letters.

*Videoprint.* This option lets you see what the document will look like when printed before you actually commit it to paper.

*Centering.* Text can be centered automatically.

*Stop printing.* This command allows you to stop printing in the middle of a document.

*Conditional paging.* This option allows you to specify that a new page should be begun under certain conditions. It allows you to prevent pages having only one line on them or lists being broken up between pages.

*Find and replace.* With this feature you can locate and alter all occurrences of a particular word or phrase in your document.

*Cut and paste.* This allows you to move sections around in a document. You can "cut" something out of one area and "paste" it in another. Single paragraphs—or entries in a list—can be moved simply by pressing control-M once and then hitting U or D to move up or down paragraph by paragraph.

*Glossary.* You can maintain a glossary of frequently used phrases and have them printed in your document by simply pressing the previously designated key.

*Labels.* Mailing labels can be printed.

*Multiprinting.* Two or more very long documents can be linked together for final printing.

*Zardax* is an excellent and very thorough word processing system. It's easy to use, contains most of the features you're likely to want as part of your system, and has a good user manual.

**Word Handler II**, Silicon Valley Systems, 1625 El Camino Real, Suite 4, Belmont, CA 94002; (415) 593-4344. $199.

Equipment required: 48K, one or two disk drives, monitor, printer.

The *Word Handler II* word processing system is designed to run without requiring any hardware modifications whatsoever. It also offers some other capabilities that aren't found in the typical word processing package. One such feature is direct-to-disk editing, which allows you to work with documents that are larger than the Apple's main memory.

Some of the strong points of this package include its ease of operation and the clarity of its display format (which includes lower case, underlining, superscripts, subscripts, boldface type, justification, and unlimited tabs). You can move forward (or backward) by character, word, line, or page, and you can move to any particular phrase you specify. Blocks of text can be copied and/or moved, and your printed document can include page headings and page footers.

*Word Handler II* displays a full sixty-six columns on screen. In effect, what you see on screen is what you get on the page. The displayed characters are, as you would expect, somewhat small and you may experience some problems with resolution on some color television sets. Nevertheless, it's very nice to be able to see the final form a document will take without having to add special hardware to your system.

The program is compatible with a variety of printers including the Epson MX 80/100, Qume, Diablo, NEC, and the IDS 460/560. The standard word-wrap feature is included in the system. If you want uppercase letters, you can either use the escape key or install a shift-key modification (not included).

If users need questions answered or encounter problems, they are invited to call a special hot line. The reference card provided with the program is also quite a good source of information.

In addition to displaying the text of the document on the screen, the program includes some lines at the bottom of the screen that tell you such things as the name of the document, what page you are on, and so on.

The current version of the program has no provision for moving quickly to the beginning or the end of a document. Also, you can't send special control characters to the printer.

The user manual is brief (fewer than fifty pages). The program's designers believe that users can learn to use *Word Handler II* in about twenty minutes, and it is simple. That's a good thing, because the manual is not especially appealing. It would be improved by a more professional appearance. For example, a key is symbolized by a letter enclosed in a rectangle that looks hand-drawn. Trivial? Yes. But it does reflect on the product.

It appears that the manual was created using *Word Handler II.* This is certainly a good idea. But if figures and drawings are going to be included they should be well done. The content of the manual is not difficult reading, and it does indeed achieve the goal of getting the user working with the system after only a short time.

*Word Handler II* offers some more attractive features:

*Fabulous fan fold.* Documents can be printed to appear on fan-folded paper. In other words, you can print a document in such a way that it can easily be turned into a brochure.

*Twenty questions.* Included in the package is a pamphlet answering the twenty "most asked questions" about *Word Handler II.* This is one example of the manufacturer's concern for and responsiveness to users of the program.

*Moving around and through.* Cursor movement around and through the text is relatively easy (for example, control-W moves the cursor through the text one word at a time). The arrow keys handle the direction of movement.

*Space identification.* Unbreakable spaces can be identified. This prevents certain combinations of words from being split between two lines. Many professional authors demand that this or some other hyphenation capability be a part of their system.

*Replace and conquer.* Global replace is included as a standard part of the program.

*Field insertion.* The insertion of fields into a predefined form letter is available.

The system works rather well despite some of the aforementioned limitations. As always, it's up to you to decide which features are most important to your particular application. Then you're in a position to make the most cost-effective decision given your definition of need. ■

# Format II.™ Word processing so advanced anyone can use it.


Format II was tested for six months in the demanding arena of Wall Street. Distributed by local dealers to law firms, financial institutions, and the like, Format II was used by secretaries and non-technical personnel. Overwhelmingly, Format II was recognized as one of the easiest and most effective word processing systems available *in any form.*

Here's why Format II is unique among word processing programs:

**What you see is what you get.** Format II performs virtually any editing and formatting function you can imagine and displays on the screen the text exactly as it will print out—paragraphs, underlining, justification, page breaks.

**Simple to Use.** You edit and format text with single key strokes. "D" for delete, "E" for edit, "I" for insert, "J" for justify, etc. Easy-to-remember commands because they make sense.

**It supports all printers** compatible with the Apple. Shadow printing, boldfacing, proportional spacing, sub- and superscripting are all available.

**Includes a sophisticated Mailing List.** Stores and retrieves names and addresses which may be printed on labels or incorporated into documents. Powerful "logic" commands allow you to select only those records which match specified criteria.

And more. Format II is lightning fast and menu driven. Add them all together and it's clear why Format II has generated such enthusiasm. Now, at a cost of $250, you *too* can buy the best.

Format II is available from most local dealers. If not, ask them to contact us, or order directly and receive a 30 day money-back guarantee.

Kensington Microware Ltd., 300 East 54 St., Suite 3L, New York, NY 10022, (212) 486-2802.

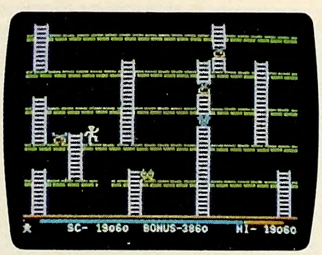
KENSINGTON MICROWARE

*System Requirements: 48K Apple® II Plus with 80 column card*

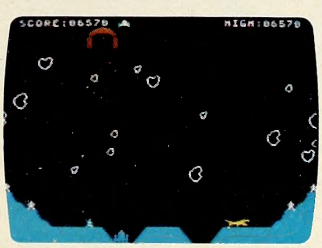
# the finest computer games. You keep telling us.


**SEAFOX**
*Sub against the convoy*
*Apple & Atari disk*

**DEADLY SECRETS**
*Hi-res adventure*
*Apple disk*

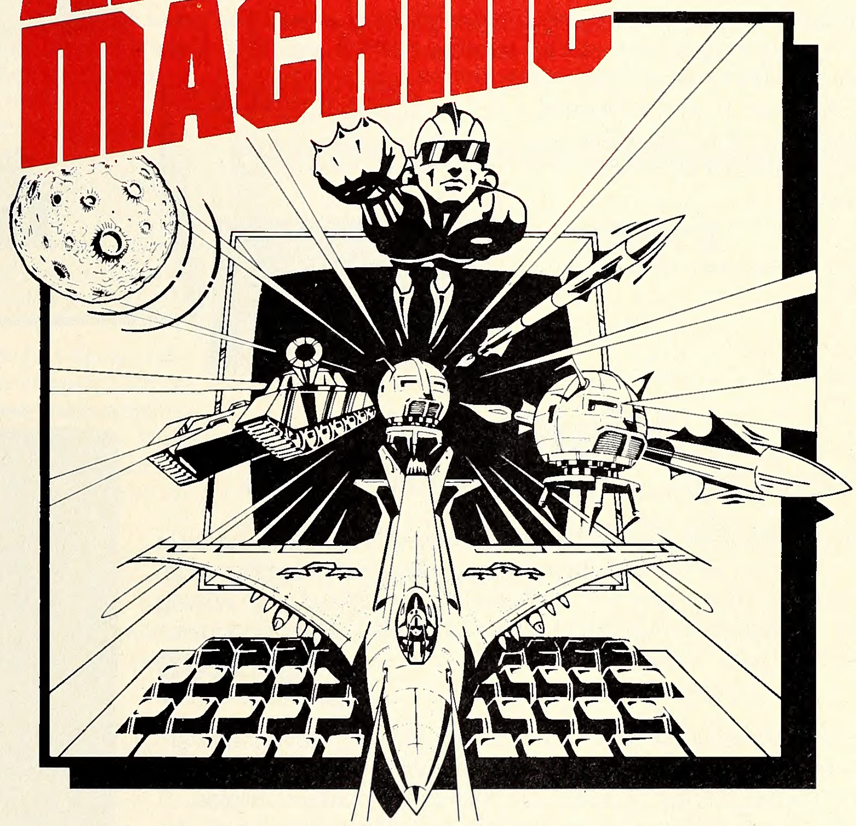



**APPLE PANIC**
*The arcade classic*
*Apple disk, Atari cassette & disk*



**STELLAR SHUTTLE**
*Monsters and meteors*
*Atari cassette & disk*

## THE ARCADE MACHINE



*PUT US OUT OF BUSINESS! THE ARCADE MACHINE lets you design and produce your own computer games, without any programming knowledge! Send us your best game and enter the BRØDERBUND ARCADE MACHINE CONTEST. We'll be giving away thousands of dollars worth of hardware and software in prizes. If you have a creative touch and an artistic eye, you too can be an arcade designer. Write to Brøderbund for contest details or visit your participating retail store! (Available on Apple disk.)*

**Software**

*All Brøderbund games are fully guaranteed. If they ever fail to boot, return the original disk to Brøderbund for a free replacement. If you have physically damaged the disk please include five dollars for replacement.*

# VENTURES WITH VISICALC

BY JOE SHELTON

"What if" just might be *VisiCalc*'s forte. Even users who are just beginning to use *VisiCalc* can quickly see ways of using this program in what-if analysis. There are some other less well known ways of using *VisiCalc* as an aid to decision making. This month we'll look at two such methods: decision matrix and probability/value matrix, both of which can be used by managers in arriving at qualitative or quantitative decisions.

Ever wonder if you could have fun with *VisiCalc*? Well, if you're a person who likes to gamble, *VisiCalc* could be an eye opener. In the course of this installment we'll also take a look at some ways of using *VisiCalc* to determine probabilities.

**Probabilities in Business.** Managers often make decisions based upon unconscious calculations of probabilities. Decision matrix and probability/value matrix are two methods of quantifying the decision-making process.

The decision matrix facilitates the decision process by helping to define the important factors and their significance to the outcome. The decision matrix method is a cousin to probability theory. The probability/value matrix uses probabilities to provide quantitative answers. These two decision-making methods are similar. The first provides a numeric method of choosing a nonnumeric alternative, while the second provides a numeric solution to a numeric problem.

**Decision Matrix.** How do managers make decisions? Often, their process runs something like this: look at as much of the available information as possible; apply judgmental factors (probabilities); make a decision. But often managers who have enough information available to make a good decision don't take enough time to weigh the facts. Here's where a decision matrix comes in.

A simple decision matrix can be used for making decisions about anything from buying a company to what color rug an office should have. If you want to apply this method to your own life, try using it to decide which car to buy or where to take the family on vacation.

As always, the first thing to do is boot *VisiCalc*.

Our next step is to define the format of the matrix. Let's put decision criteria down the left side of the matrix and decision options across the top.

For purposes of example, why don't we decide which of three people to hire for the position of company sales manager. In this instance we'll rate Smith, Jones, and Brown only on the three criteria shown in columns A and B, although you could add as many decision criteria as you wanted.

Enter the decision criteria shown in figure 1 into columns A and B of the matrix. Then enter the candidates shown in row 2 into the appropriate cells.

| | A B | C | D | E |
|---|---|---|---|---|
| 1 | | | | |
| 2 | DECISION CRITERIA | SMITH | JONES | BROWN |
| 3 | TECHNICAL EXPERIENCE | | | |
| 4 | MANAGERIAL EXPERIENCE | | | |
| 5 | EDUCATIONAL BACKGROUND | | | |

Figure 1.

Next comes the evaluation. In each cell in the matrix, enter a value that subjectively or objectively evaluates each decision point against each decision criteria. Any numeric reference can be used, but for our purposes, let's use 1 to 10 (with 10 being the highest rating). For example, we would rate Smith as a 7 in technical experience by entering a 7 in cell C3. Complete the remainder of the matrix by entering the values shown in figure 2.

| | A B | C | D | E |
|---|---|---|---|---|
| 1 | | | | |
| 2 | DECISION CRITERIA | SMITH | JONES | BROWN |
| 3 | TECHNICAL EXPERIENCE | 7 | 7 | 8 |
| 4 | MANAGERIAL EXPERIENCE | 8 | 6 | 6 |
| 5 | EDUCATIONAL BACKGROUND | 7 | 7 | 9 |
| 6 | RESULT | 22 | 20 | 23 |

Figure 2.

Summing each of the columns (for example, @SUM(C3...C5)) would give you a gross estimate of each manager's potential. See figure 2 for the result.

Based on the assessment illustrated in figure 2, Brown seems to edge out the other two candidates. But is this really a fair assessment? A case could be made that managerial experience is worth more than either technical experience or educational background. One easy method of weighting the values would be to fudge the managerial section by adding additional points to values in the managerial experience row.

A more professional (and more useful) approach would be to apply weighting variables to each selection criterion. Move to cell A9 and B9 and enter MGR EXP WEIGHTING. You could weight each of the values, but for our example we'll weight only managerial experience. Assume that managerial experience is worth 1.5 times the other criteria and enter 1.5 in cell C9.

In C6 enter +C3+(C4*C9)+C5. Replicate the formula into D6 and E6 using Relative, Relative, No Change, and Relative references.

Brown and Smith are now tied. We might give the nod to either one of them based on personality or on our own instincts. Or, we might add another decision criterion and base our decision on that. Remember, we could have many more decision criteria, each with a different weighting.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | DECISION CRITERIA | | SMITH | JONES | BROWN |
| 3 | TECHNICAL EXPERIENCE | | 7 | 7 | 8 |
| 4 | MANAGERIAL EXPERIENCE | | 8 | 6 | 6 |
| 5 | EDUCATIONAL BACKGROUND | | 7 | 7 | 9 |
| 6 | RESULT | | 26 | 23 | 26 |
| 7 | | | | | |
| 8 | | | | | |
| 9 | MGR EXP WEIGHTING | | 1.5 | | |

Figure 3.



| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | | | | | | HISTORICAL | |
| 2 | | SUPPLY | PRICE | SEASONAL | FACTOR | SALES | FORECAST |
| 3 | JAN | | | 8 | | | |
| 4 | FEB | | | | | | |
| 5 | MAR | | | | | | |
| 6 | APR | | | | | | |
| 7 | MAY | | | | | | |
| 8 | JUNE | | | | | | |
| 9 | JUL | | | | | | |
| 10 | AUG | | | | | | |
| 11 | SEP | | | | | | |
| 12 | OCT | | | | | | |
| 13 | NOV | | | | | | |
| 14 | DEC | | | | | | |

Figure 4.

You can see that using *VisiCalc* can make the process of determining a solution much easier.

Now try "what if?" See what weighting value allows Smith to have the highest score. Enter 1.6 in C9 and you can see a .2 difference. Entering 2 in C9 gives a 1 point difference.

**Probability/Value Matrix.** Forecasting is a perfect example of a numbers game that business people often play ineffectively. Assume that you're going to forecast the sales of anchor cable. A number of factors can affect sales. For example, bad weather could mean that fewer people are boating, the price of the cable could be so high that people aren't replacing it regularly, or the manufacturer might not be producing enough cable to meet the demand. In addition, each of these factors might have a different effect on each month you're trying to forecast. With all this in mind, how would you go about forecasting sales?

The probability/value matrix involves using subjective estimates to determine a factor that can be applied against a base forecast to determine an actual forecast.

The first step is to set up the matrix shown in figure 4. You'll notice that this matrix is similar to the one shown in figure 1.

Next, some assumptions about the model. We will once again enter a value between 1 and 10 in each cell. This value will indicate how we assess each factor against the sales of the previous year. For example, the 8 that has been entered in D3 indicates that we expect to do 80 percent of last year's January sales based on the assumption that we'll have poor weather in January. Imagine that you are forecasting the sales of cable and fill in the cells in columns B, C, and D with values from 1 to 10 based upon whatever assumptions you choose.

Next, determine the factor to apply against historical sales. Move to cell E3. Consider what we are trying to accomplish. We have assumptions for three of the factors that might affect January sales. You'll notice that the prior columns are supply and demand oriented. If we look at the value in D3, we assume it to mean that we expect to do 80 percent of last year's sales. If last year's sales totaled one thousand feet, we can expect to sell eight hundred feet in January of this year, based on the seasonal adjustment. The same reasoning can be applied to the remaining two values. Then it's a simple matter to average the factors in column E and apply the factor that results to historical sales. This will provide a forecast in column G. The formula in cell E3 would be:

```
@AVERAGE (B3...D3)/10
```

Enter some historical sales values (in feet) in column F. Then, in column G, multiply cell E3 times F3 (+E3*F3) to get the forecast for January. You can complete the matrix by replicating the formulas in columns E and G and entering the remaining historical sales.

What if you feel that one sales factor should carry more weight than the others? Let's say that the seasonal factor is twice as important as the supply or the price. Then the formula in cell E3 would be:

```
@AVERAGE(B3,C3,D3,D3)/10
```

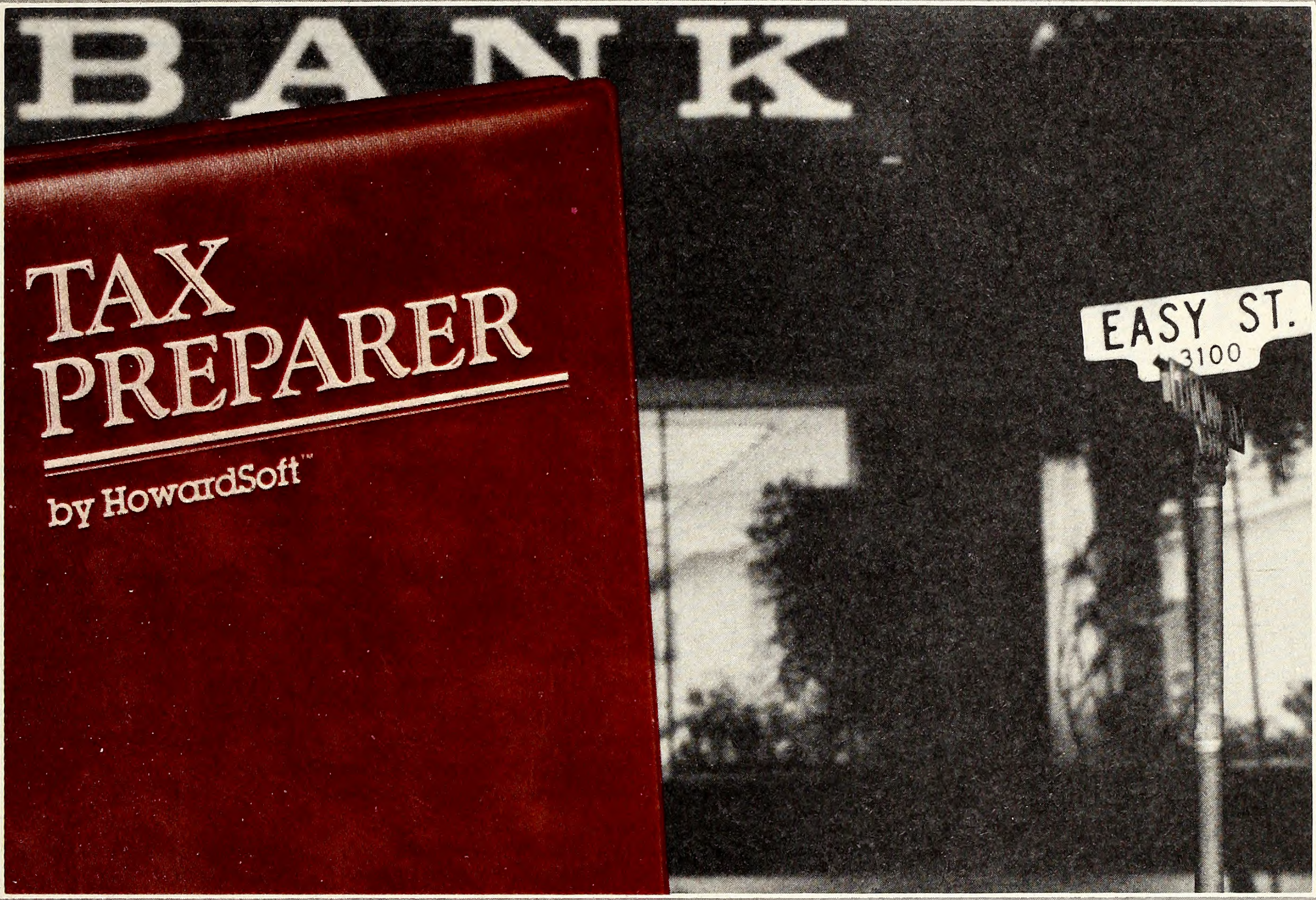
or

```
@AVERAGE(B3...D3,D3)/10
```

If you use the second formula, it's quite easy to enter additional sales factors into the model. Add some columns between columns B and D

# WHERE WILL YOU BE ON APRIL 15TH?

Professionals prosper with the TAX PREPARER By HowardSoft™




Whether you're a professional trying to expand your business or an individual trying to minimize your tax bite, this software is the answer . . . and NOW is the time to buy. You can start entering data now. Then when the new tax laws and forms are finalized each year, you can alter your software with our low-cost update and effortlessly print out tax returns in IRS format.

The most complete tax package for personal computers, the TAX PREPARER by HowardSoft handles more than 20 forms and schedules and provides full itemizing and recordkeeping. No wonder that thousands of professionals have made it the best selling tax software for the past 2 years.

☐ PREPARE TAX RETURNS FASTER
☐ END THE MANUAL ERRORS
☐ ENTER DATA NOW TO GET AHEAD

UPDATE POLICY: Tax laws and tax forms may change, but your software never need be out-of-date. When you buy HowardSoft you are assured of timely low-cost updates and are protected from future price increases. Buy now for early updating.

## . . . AND WHERE WILL YOU PUT THE EXTRA MONEY YOU EARN?


HowardSoft financial software gives you the edge through objective comparisons, realistic projections, and the full consequences of the latest tax laws. This software is a must for making intelligent buy/sell decisions in the difficult investment climate of today.

Apple and IBM Editions now at your local dealer.

HSS Howard Software Services
for the SERIOUS Personal Computer User
8008 Girard Ave. Suite 310 | La Jolla, CA 92037 | (714) 454-0121

| | First Throw | Second Throw | Third Throw |
|---|---|---|---|
| 1. | Heads | Heads | Heads |
| 2. | Heads | Heads | Tails |
| 3. | Heads | Tails | Heads |
| 4. | Tails | Heads | Heads |
| 5. | Heads | Tails | Tails |
| 6. | Tails | Heads | Tails |
| 7. | Tails | Tails | Heads |
| 8. | Tails | Tails | Tails |

Figure 5.

and the @AVERAGE expression will include them in computing the average. If these additional factors should be weighted, you'll have to include them a second time after the range (B3...D3) in the expression.

**It's Fun Time—Place Your Bets.** Let's try a favorite form of probability: tossing a coin. It is well known that the probability of throwing heads or tails on one toss of a coin is 50 percent. In other words, the odds of throwing either one are one chance out of two possibilities. If you throw a coin three times, the probability that it will come up heads is 50 percent for each throw. The question is, what is the probability of throwing a coin and getting heads three times in a row?

It's obvious that the odds of heads coming up on the first throw are 50 percent. But what about the odds on the second and third throws? Well, if you throw one coin three times, you have a chance of getting the combinations shown in figure 5.

For three throws there are eight possible outcomes and, as you can see, only one possibility of getting heads three times in a row! The odds are one out of eight, or 12.5 percent.

There's a numeric method of determining probability that is easier than writing down all the possible outcomes and figuring the results.

Let's begin with the total possible outcomes. There are two possible outcomes for each toss and three tosses. The total number of outcomes is determined by multiplying the number of possible outcomes for each toss times itself for the number of throws being made. In our example you would multiply 2 to the third power, or 2 x 2 x 2 = 8. If you tossed the coin five times, there would be thirty-two possible combinations.

If you're interested in throwing something other than straight heads or tails, it's important to consider what the odds are of throwing two tails and one heads. Look at figure 5 and you'll see that it's possible to throw two tails and one heads in three different ways. But if the question is couched, "What are the odds of throwing heads followed by two tails in three throws?" the odds would be 12.5 percent, or one in eight. Heads-tails-tails occurs only once.

The next question is, can *VisiCalc* compute the odds of throwing heads five, six, or seven times in a row? Let's give it a try.

Clear *VisiCalc* (/CY).

In A3 enter POSS/EVENT. This is the total number of combinations that are possible with each throw of the coin. Enter the answer 2 in B3. If we were throwing dice, the answer for throwing an individual die would be 6.

You'll probably want to set the column width to 12 (/GC12).

In A6 enter "# OF ROLLS.

In B5 enter TTL POSS for total possible and in B6 enter COMBIN for combinations.

In A8 through A17 enter 1 through 10.

In B8 enter +B3^A8. This expression will evaluate 2 (the possible occurrences per roll) to the nth exponent, where n is the total number of rolls.

Replicate B8 into B9 through B17 using No Change and Relative references.

Column B should display some interesting information. The odds of throwing heads ten times in a row are 1 in 1,024. That translates into a probability of .0009 percent. Want to bet you could throw heads ten times in a row?

*VisiCalc* can be used to determine many other types of probabilities. In future columns, we'll examine some other ways to have fun with probability.

# FAST TALK.

## Now, 212 for your Apple* from Novation.

### 1200 bps, full duplex

You can turn your Apple into a fast talker—and do it right now. Our 1200 bps full duplex option is at your dealers.

It's an upgrade to our full personal communication system, the Apple-Cat II modem. The add-on is simple and no additional slot in your Apple is needed. You can start with the basic system, then move up to the 212 full duplex whenever you're ready.

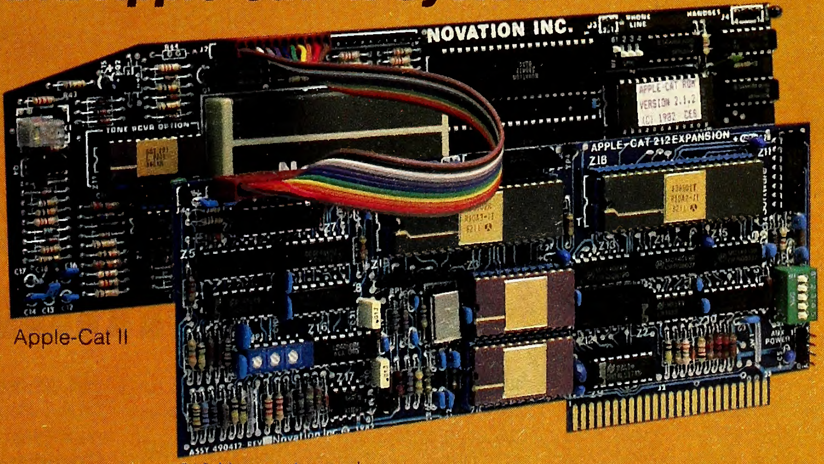
It's the quickest, most accurate way to work with another computer, to swap programs, to access data banks. Whatever.

It's also the quickest, surest way to enhance the investment you've made in your Apple.

Get all of the details. Call (213) 996-5060 or 800-423-5419.

Or see your dealer.

### 212 Apple-Cat™ II System




Apple-Cat II

212 Up-grade card

Novation™

Novation, Inc., 18664 Oxnard Street, Tarzana, CA 91356

## VISICALC*

INTERACTIVE ELECTRONIC WORKSHEET — **YES**

ON-LINE REFERENCE GUIDE — **NO**

"NAMING" OF CELLS OR AREAS — **NO**

PLAIN ENGLISH PROMPTS — **NO**

INDIVIDUAL COLUMN WIDTHS — **NO**

EXTENSIVE FORMATTING CAPABILITIES — **NO**

PROTECTED CELLS — **NO**

MULTIPLE, LINKED WORKSHEETS — **NO**

SORTING CAPABILITY — **NO**

# VisiCalc® was a swell idea for then.

**The next generation.** First generation electronic worksheets were a good idea. They were early software management tools that could eliminate a lot of hours with a spreadsheet, calculator, pencil and eraser. Enter Multiplan, the next generation electronic worksheet that's as easy to use as it is useful.

**Make comparisons.** Compare Multiplan to any of the earlier electronic worksheets. We've given you some "prompts" above.

**Compare learning time.** Multiplan's tutorial book brings you up to speed. Fast. But Multiplan doesn't stop there. Multiplan's On-line Reference Guide gives you instant help if you have questions. It knows where you are in Multiplan and offers information related to your problem, right on the screen.

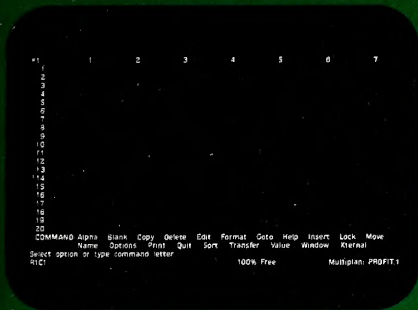
**Compare ease of use.** All Multiplan prompts are full length words or phrases. And Multiplan provides "naming," the ability to assign a plain English name to any cell or area. "Gross Profit = Sales—Cost" rather than "AA44=AZ23—BK154." Which means you can work more intuitively. And faster.

**Compare utility.** Multiplan lets you link related worksheets so that information is transferred between them automatically. For instance, you can keep regional sales forecasts on separate sheets but link them with your overall company forecast. Then, just change the forecast for any region, and the company forecast sheet is updated automatically. Something you can't do with first generation worksheets.

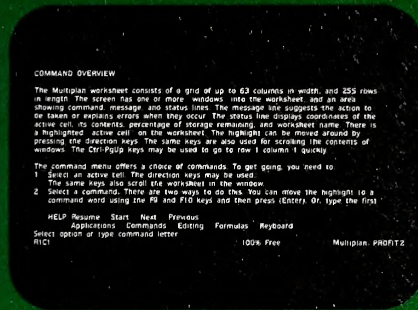
**Compare reports.** Not just the work you can do, but the way you can present it. Multiplan's flexible formatting options allow you to produce presentation-quality reports. And its sorting capability lets you sort by either alphabetic or numeric order. So a sales manager who normally lists sales regions alphabetically could sort by amount sold and conveniently rank by sales performance. The

*Based on features in releases VC-202B0-AP2 and VC-156Y0-IBM of VisiCalc on the Apple II and IBM-PC respectively.

# MULTIPLAN

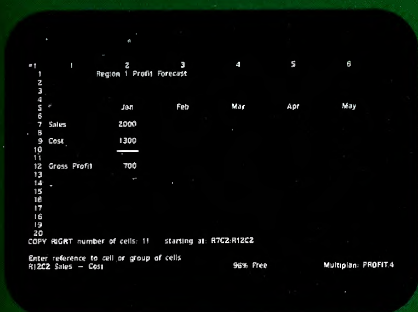


INTERACTIVE ELECTRONIC WORKSHEET **YES**

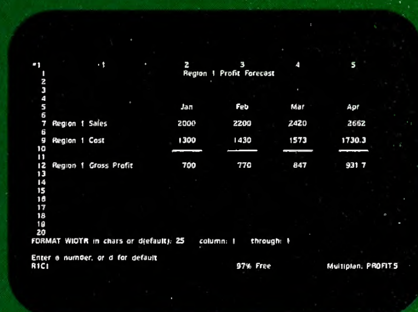


ON-LINE REFERENCE GUIDE **YES**

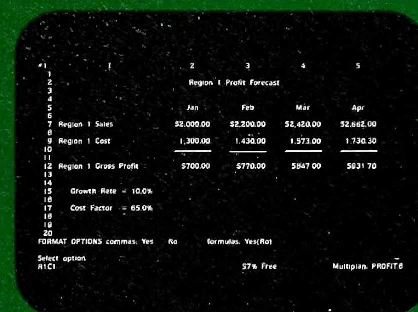
"NAMING" OF CELLS OR AREAS **YES**



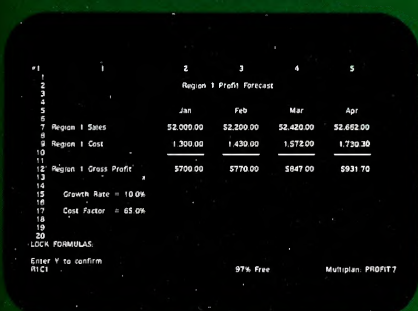
PLAIN ENGLISH PROMPTS **YES**



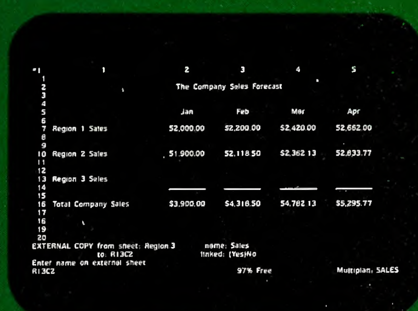
INDIVIDUAL COLUMN WIDTHS **YES**



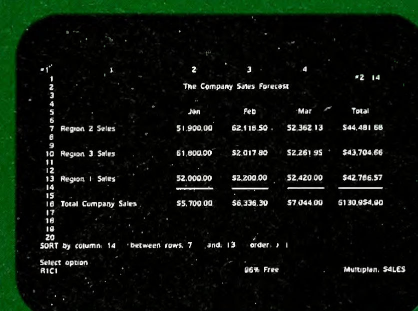
EXTENSIVE FORMATTING CAPABILITIES **YES**



PROTECTED CELLS **YES**



MULTIPLE, LINKED WORKSHEETS **YES**



SORTING CAPABILITY **YES**

# Multiplan™ is a great idea for now.

result is a more meaningful and useful presentation of data.
**Compare availability.** With Multiplan, you're not limited to a single range of computers. Multiplan is available for Apple® and for microcomputers that run MS™-DOS, XENIX,™ or CP/M-80® operating systems. Multiplan supports both 40- and 80- columns on the Apple II.
**Compare the source.** Microsoft was the world's first microcomputer software company. Today, Microsoft software is running in well over a million installations, worldwide. Languages. Utilities. Business programs. All, maintained at the state-of-the-art.
**Compare for yourself.** Drop into your computer store. Compare Multiplan's powerful, user-oriented features to any electronic worksheet on the market. If you've been using VisiCalc, Multiplan's ability to directly utilize your VisiCalc files lets you easily upgrade to Multiplan. And that's just another of the many features designed to make Multiplan the electronic worksheet for now. And years from now.

**BETTER TOOLS FOR MICROCOMPUTERS**

**MICROSOFT®**

# Presto! DOS PIE in CP/M

# BY DON WORTH

There is a small but very vocal group in the Apple user community who will tell you that the *PIE* full screen editor, published by Hayden, is the last word in editors for a microcomputer. Despite the risk of exaggeration, many will claim that *PIE* is more flexible and responsive than most high-powered editing systems even on large time-sharing systems. Perhaps it is *PIE*'s well tuned command structure and careful "user engineering" or perhaps it is just that its aficionados have used it so long that they have become overly biased (perhaps a bit of both). Whatever the reasons, the thought of having to work under another operating system, such as CP/M, without one's favorite editor can be likened to fixing a watch while wearing boxing gloves! Even if another full screen editor was available, having to juggle two sets of control key layouts in one's mind while simultaneously trying to keep straight the differences between DOS and CP/M is a task that is next to impossible. Why can't *PIE* run under Apple CP/M?

Why not? The 6502 processor is still there, albeit hidden behind the SoftCard's Z-80. A small *front end* could be written in Z-80 assembly language that would read a text file into *PIE*'s memory buffer, switch CPUs, and pass control to *PIE*. When *PIE* exits, the front end, running under the Z-80 again, could save the edited file back to the CP/M disk. To give credit where credit is due, this is exactly how Randall Hyde did an early, somewhat experimental, forty column version of *PIE* that ran under CP/M. What was needed was a reliable front end that would support both forty and eighty column versions of the *PIE* editor as well as providing support for both text editing and program development. Enter *CPMPIE.*

**How It Works.** *CPMPIE* consists of an assembly language front end, written in Z-80, wrapped around the 6502 machine language *PIE* editor. The *PIE shell*, or command mode, is removed and functionally replaced with the Z-80 code. The front end reads a CP/M text file into memory (exactly where the *PIE* editor expects to find it) and then calls *PIE* in 6502 mode. Should the editor exit back to the front end (using shift-control-P) the Z-80 code writes the modified memory image back out to the same (or any other) CP/M text file. Obviously, since *PIE* must run in 6502 mode, *CPMPIE* will only run on an Apple II with a Z-80—so owners of other CP/M machines are out of luck! In theory, any version of the *PIE* editor will work with *CPMPIE*, although it has only been tested with the Smarterm eighty column version and the standard forty column version. The new *PIE Writer* should also work if certain changes are made to the front end (the editor takes up more space in memory and zero-page assignments are slightly different). This will be left to the enterprising reader.

When writing the front end, several special features were added for editing files with a suffix of .asm. When an .asm file is read, all tab characters are expanded to *PIE*'s default tab positions (every eight columns). *PIE* is also placed in upper case lock upon entry. When the file is later written back to disk, tabs are inserted whenever possible and blanks are removed. This was done to save disk space (about 33 percent over files without tabs) and to provide a more natural appearance to assembler listings. Tabs are not inserted within quoted strings, however, as this could play havoc on define byte ASCII strings!

As a CP/M file is read, line feeds are discarded and the most significant bit is forced on. As a file is written back to disk, the most significant bit is turned off and a line feed is inserted after each carriage return. A control-Z character is added to mark the end of the file. These are all standard CP/M conventions.

**Using CPMPIE.** The procedure outlined later will create a com (command) file called CPMPIE.com. (Of course the file may be renamed to PIE.com, P.com, or anything else if typing six letter commands is difficult.) To invoke CPMPIE, one may type any of the following:

```
CPMPIE (newfile.typ)
```

or

```
CPMPIE (oldfile.typ)
```

or

```
CPMPIE (oldfile.typ outfile.typ)
```

In the first case, no file is loaded, and upon exit, *CPMPIE* will save the text that has been entered as a new file using the name given. In the second case, the file substituted for oldfile.typ is first read into memory, and, when *PIE* is exited, the old file is overwritten by the changed text. In the third case, an existing file is read, but the modified text is written to the file designated as the output file. For example:

```
CPMPIE PROG.ASM PROG.BAK
```

will read from Prog.asm but will write to Prog.bak. Note that the above syntax is not entirely standard for CP/M but is probably less confusing. Remember also that if the input file has the .asm suffix then tab processing will be done and *PIE* will initially be placed in upper case lock. To exit *PIE* without saving to the output file (in any of the three cases above) simply press reset. CP/M will warm start and the input file, if there is one, will not be affected in any way.

**Procedure for Creating CPMPIE.** To create *CPMPIE*, first boot a *PIE* distribution disk and configure the editor to your hardware and your own tastes (upper/lower case display modes, bell column, shift key modification, and so on). Also, tabs should be set at every eight columns (this is the default). Now, boot DOS and isolate the *PIE* editor in a binary file as follows:

```
BLOAD APPLE PIE
BSAVE EDITOR,A$1280,L$2080
```

To the author's knowledge, the address and length here are identical for all versions of *PIE*. *PIE Writer*'s editor begins at the same location but is greater in length. After you have created a file containing the editor (the *PIE* command shell has been removed), boot CP/M and create a new CP/M disk containing only the APDOS.com and DDT.com files found on the CP/M master disk. Your first job will be to move the editor over to CP/M:

```
APDOS
A:EDITOR=A:EDITOR
```

You will be instructed to insert the DOS disk containing the binary file you previously created and then the CP/M disk you have prepared (this cycle will occur twice). Now you are ready to type in the *CPMPIE* front end. Type:

```
DDT
```

to get into DDT. The computer will respond:

```
DDT VER 2.2
?
```

Type:

```
S100
```

and the display will return:

```
0100  00
```

DDT is showing you the contents of location 0100 in hexadecimal. It may not be 00 on your machine but that's okay. Type the new value shown in Hex Dump, Part 1 (location 0100 is to be a C3). DDT will then show you location 101. Type its value (CD) and continue until all of Part 1 has been entered (through address 259). When you are finished, type a period to exit the Set mode. You can do this at any time during entry if you make a mistake and use the S command with another address to start from where you left off or to back up. Now type *S2416* and repeat the above procedure to type the data in Hex Dump, Part 2. When

```
0100  C3 AF 24 C3 AF 24 00 4C 00 BF 00 3A 00 00 21 86  ..$..$.L...:..!.
0110  12 22 D0 F3 2A DE F3 77 11 49 02 0E 09 CD 05 00  .".."..w.I......
0120  11 80 00 0E 09 CD 05 00 11 5C 00 0E 13 CD 05 00  ......... ......
0130  11 5C 00 0E 16 CD 05 00 FE FF CA 0A 02 2A 00 F0  . ...........*..
0140  22 06 F0 3A 07 F0 D6 10 32 07 F0 3A 03 F0 D6 10  "..:....2..:....
0150  32 03 F0 21 80 00 22 00 24 3E 00 32 14 24 32 15  2..!..".$½.2.$2.
0160  24 CD 15 02 32 03 24 3A 14 24 3C 32 14 24 3A 0A  $...2.$:.$¼2.$:.
0170  01 B7 CA D1 01 3A 14 24 E6 07 FE 01 C2 90 01 3A  .....:.$.......:
0180  15 24 B7 CA 90 01 3E 09 CD 20 02 3E 00 32 15 24  .$....½.. .½.2.$
0190  3A 0A 01 E6 07 C2 AA 01 3A 03 24 FE 20 C2 AA 01  :.......:.$. ...
01A0  3A 15 24 3C 32 15 24 C3 61 01 3A 15 24 B7 CA C1  :.$¼2.$.a.:.$...
01B0  01 4F 3E 20 C5 CD 20 02 C1 0D C2 B2 01 79 32 15  .0½ .. ......y2.
01C0  24 3A 03 24 FE 27 C2 D1 01 3A 0A 01 EE 01 32 0A  $:.$.'...:....2.
01D0  01 3A 03 24 B7 CA F5 01 CD 20 02 FE 0D C2 61 01  .:.$..... ....a.
01E0  3E 00 32 14 24 3A 0A 01 E6 80 32 0A 01 3E 0A CD  ½.2.$:....2..½..
01F0  20 02 C3 61 01 3E 1A CD 20 02 CD 32 02 0E 10 11   ..a.½.. ..2....
0200  5C 00 CD 05 00 FE FF C2 00 00 11 53 02 0E 09 CD  ..........S....
0210  05 00 C3 00 00 2A 06 F0 7E E6 7F 23 22 06 F0 C9  .....*.. ..#"...
0220  32 02 24 2A 00 24 77 23 22 00 24 3E 01 BC 3A 02  2.$*.$w#".$½..:.
0230  24 C0 21 80 00 22 00 24 11 5C 00 0E 15 CD 05 00  $.!..".$. ......
0240  B7 3A 02 24 C8 E1 C3 0A 02 1B 2A 53 41 56 49 4E  .:.$......*SAVIN
0250  47 20 24 07 0D 0A 45 52 52 24  G  $...ERR$
```

Hex Dump, Part 1.

```
2416  0D 0A 43 50 4D 50 49 45 20 2D  ..CPMPIE –
2420  2D 20 41 50 50 4C 45 20 50 49 45 20 56 45 52 53  – APPLE PIE VERS
2430  49 4F 4E 20 32 2E 30 0D 0A 0D 0A 28 43 29 20 43  ION 2.0....(C) C
2440  4F 50 59 52 49 47 48 54 20 31 39 38 31 0D 0A 20  OPYRIGHT 1981..
2450  20 20 20 50 52 4F 47 52 41 4D 4D 41 20 49 4E 54     PROGRAMMA INT
2460  45 52 4E 41 54 49 4F 4E 41 4C 20 49 4E 43 2E 0D  ERNATIONAL INC..
2470  0A 0D 0A 46 52 4F 4E 54 20 45 4E 44 20 42 59 20  ...FRONT END BY
2480  44 4F 4E 20 57 4F 52 54 48 20 28 55 43 4C 41 2F  DON WORTH (UCLA/
2490  4F 41 43 29 0D 0A 0D 0A 0D 0A 24 20 4E 4F 54 20  OAC)......$ NOT
24A0  46 4F 55 4E 44 24 4C 4F 41 44 49 4E 47 20 24 21  FOUND$LOADING $!
24B0  6C 00 11 04 24 31 5F CB 0E 10 CD 61 26 3E 00 32  1...$1_....a&½.2
24C0  7C 00 32 68 00 3E 00 32 0A 01 3A 65 00 FE 41 C2  .2h.½.2..:e..A.
24D0  EC 24 3A 66 00 FE 53 C2 EC 24 3A 67 00 FE 4D C2  .$:f..S..$:g..M.
24E0  EC 24 3E 80 32 0A 01 3E 00 32 94 02 11 16 24 0E  .$½.2..½.2....$.
24F0  09 CD 05 00 21 80 12 22 D0 F3 2A DE F3 77 2A 00  ....!.."..*..w*.
2500  F0 22 06 F0 11 5C 00 0E 11 CD 05 00 FE FF C2 3A  ."... .........:
2510  25 11 5C 00 CD 35 26 11 9B 24 0E 09 CD 05 00 3A  %. ..5&..$.....:
2520  05 24 FE 20 C2 00 00 3A 5D 00 FE 20 CA 00 00 2A  .$. ...: .. ...*
2530  06 F0 22 02 F0 36 00 C3 E5 25 11 A6 24 0E 09 CD  .."..6...%..$...
2540  05 00 11 5C 00 CD 35 26 11 5C 00 0E 0F CD 05 00  ... ..5&. ......
2550  FE FF CA 11 25 21 00 01 22 00 24 3A 07 F0 D6 10  ....%!..".$:....
2560  32 07 F0 3E 00 32 14 24 CD 0C 26 FE 1A CA C4 25  2..½.2.$..&....%
2570  FE 0A CA 68 25 32 02 24 3A 14 24 3C 32 14 24 3A  ...h%2.$:.$¼2.$:
2580  02 24 FE 0D C2 8C 25 3E 00 32 14 24 3A 0A 01 B7  .$....%½.2.$:...
2590  CA B9 25 3A 02 24 FE 09 C2 B9 25 3A 14 24 E6 07  ..%:.$....%:.$..
25A0  2F 3C E6 07 3C 4F 3E A0 CD 2C 26 0D CA 68 25 3A  /¼..¼0½..,&..h%:
25B0  14 24 3C 32 14 24 C3 A6 25 3A 02 24 F6 80 CD 2C  .$¼2.$..%:.$...,
25C0  26 C3 68 25 3E 00 CD 2C 26 11 5C 00 0E 10 CD 05  &.h%½..,&. .....
25D0  00 3A 07 F0 C6 10 32 03 F0 3A 06 F0 32 02 F0 2A  .:....2..:..2..*
25E0  06 01 22 06 F0 3A 05 24 FE 20 CA F8 25 21 04 24  .."..:.$. ..%!.$
25F0  11 5C 00 0E 10 CD 61 26 CD 40 26 3E 00 32 7C 00  . ....a&.@&½.2 .
2600  32 68 00 2A 06 F0 22 00 F0 C3 0B 01 2A 00 24 3E  2h.*.."......*.$½
2610  01 BC CA 1B 26 7E 23 22 00 24 C9 21 80 00 22 00  ....& #".$.!..".
2620  24 11 5C 00 0E 14 CD 05 00 C3 0C 26 2A 06 F0 77  $. ........&*..w
2630  23 22 06 F0 C9 CD 40 26 11 80 00 0E 09 C3 05 00  #"....@& ........
2640  21 5D 00 11 80 00 0E 08 CD 61 26 3E 2E 32 88 00  ! .......a&½.2..
2650  21 65 00 11 89 00 0E 03 CD 61 26 3E 24 32 8C 00  !e.......a&½$2..
2660  C9 7E 12 23 13 0D C2 61 26 C9  . .#...a&.
```

Hex Dump, Part 2.

you have finished you can type *D100,259* and *D2416,2669* to check your work. If you want to play it safe, you can save what you have typed to disk before proceeding. Press control-C and, when the system prompt appears, type:

```
SAVE 38 PIEFE
DDT PIEFE
```

This will save the memory image you have entered and return you to DDT for further work. Now type the following DDT commands (omitting the comments in parentheses):

```
IEDITOR        (ready to read Editor file)
R17C           (read Editor into location 17C+100 such that
NEXT PC         it will reside at 1280 in 6502 mode)
2700 0100
S36E           (now change locations 36E, 373, 378, and 37D
036E 08 11      from 08 to 11)
036F 85 .
S373
0373 08 11     (this is necessary since these addresses refer
0374 85 .       to locations in PIE's shell, which was removed)
S378
0378 08 11
0379 85 .
S37D
037D 08 11
037E 85 .
```

If you plan to use the forty column version of *PIE* you will need to change the location of *PIE*'s text buffers from $4C00 (eighty column versions) to $4000. To do this type:

```
S107
0107 4C 40
0108 00 .
```

If for some obscure reason you want to use the forty column *PIE* with a

# Inviting!

When the invitation arrives, you know immediately that things will be done in style.

That's the way people are thinking of **Strictly Soft Ware.**

We've issued an invitation for you to enjoy super prices, prompt delivery, and superior service.

When you're on the **Strictly Soft Ware** mailing list, you'll automatically receive our industry-leading catalog (Apple or IBM). You'll also receive news of sales, unadvertised specials, and new program releases — valuable information you can use.

So respond to the best invitation you'll find in mail-order software. Call 1-800-848-5253. There'll be no regrets.

*Your presence is requested among those who have discovered great prices, technical support, and customer service at Strictly Soft Ware.*

*R.S.V.P. 1-800-848-5253*

## SPECIALS SPECIALS

- Dictionary
- Real Estate Analyzer
- Creative Financing
- Visi Schedule
- Taxman II
- Star Maze
- Screenwriter Pro
- The Accountant
- Pursuit of Graf Spee
- Gertrude's Puzzles

**Above Specials In Effect Until Dec. 1, 1982**

## Holiday Gift Idea #1

*The **Strictly Soft Ware** Subscription Plan. Prepayment allows the account to be drawn upon for whatever you want, whenever you want. When we send the package, we'll provide a balance statement as well. Here's a gift that can last all year long. Also ideal for large institutions.*

## Holiday Gift Idea #2

*The **Strictly Soft Ware** Gift Certificate. Here's the perfect way to take the guesswork out of holiday giving. A gift certificate and the **Strictly Soft Ware** holiday catalog guarantee the perfect gift for everyone on your holiday list.*


We take Master Card or VISA (include card # and expiration date). Ohio residents add 5.5% tax. Include $2.50 for postage. 3% discount if check accompanies order.

Send for free catalog. Prices subject to change. Holiday Hours: 9-9:30 weekdays; 9-6 Sat.; 12-5 Sun.

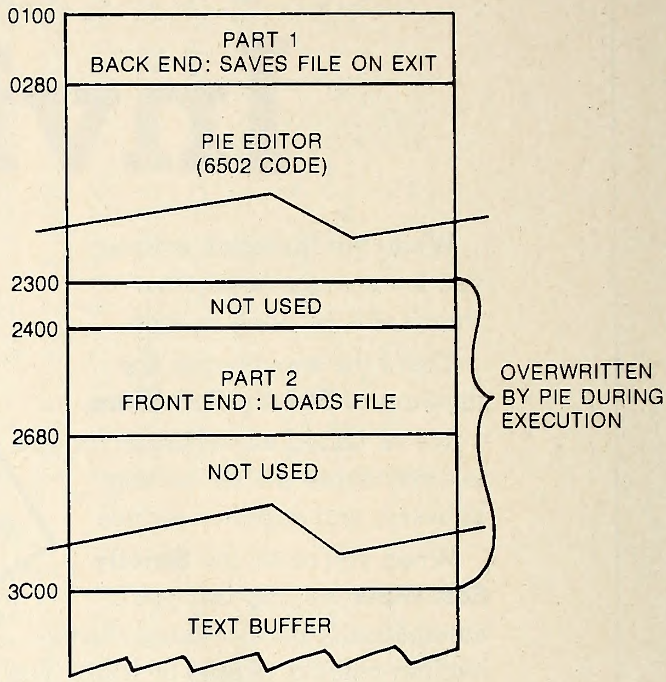
**Strictly Soft Ware**
Post Office Box 338
Granville, Ohio 43023
order by phone **800-848-5253**
in Ohio **614-587-2938**

Smarterm installed, you can cause *CPMPIE* to switch the Smarterm over to forty column mode by setting addresses 010C and 010D to BF and E0, respectively ($E0BF causes the Smarterm to select forty column mode). Don't do this if you are using an eighty column version of *PIE* or if you do not have an eighty column board! If you have some other form of eighty column board and want to do this, consult the manufacturer's documentation for the proper I/O address to select.

When all of the above has been done (phew!), type control-C to exit DDT and, when the system prompt reappears, type:

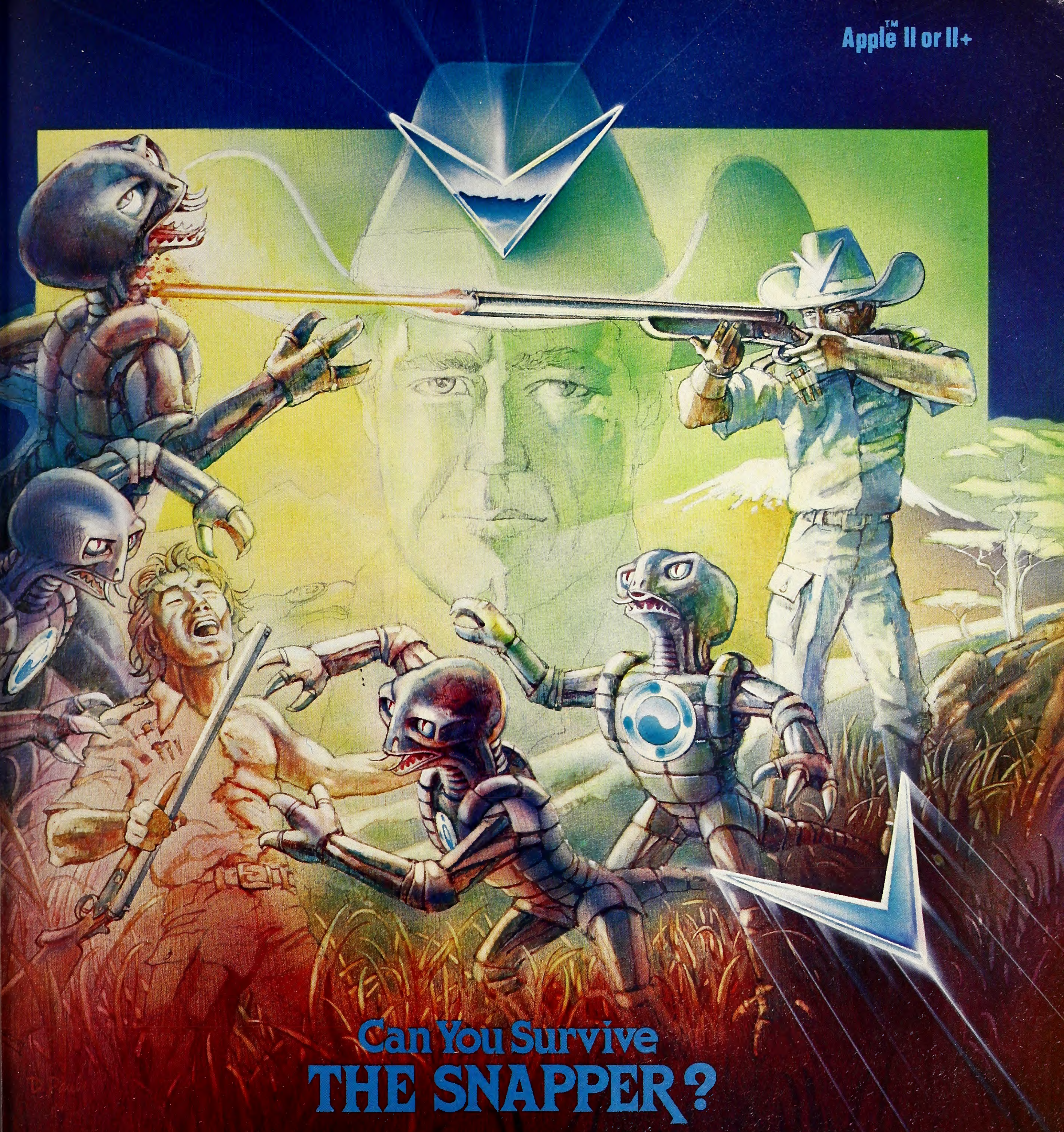
SAVE 38 CPMPIE.COM

*CPMPIE* is now ready to use! For those who are curious about inner workings, we have included a memory map showing the organization of the *CPMPIE* front and back ends, the *PIE* editor itself, and the working space for the text file. There is also a complete listing of the *CPMPIE* source file, fully commented, which was done on the ASM assembler.




CPMPIE Memory Map.

```
        CPMPIE: THIS ROUTINE ACTS AS A FRONT END TO THE
                PIE EDITOR (WRITTEN IN 6502 MACHINE
                LANGUAGE). IT LOADS A FILE, CALLS PIE AND
                SAVES THE FILE UPON RETURN USING CP/M
                ACCESS METHOD SERVICES.

        COMMAND: CPMPIE FILENAME

                 ..OR..

                 CPMPIE INFILENAME OUTFILENAME

        PROGRAMMER: DON D WORTH (UCLA/OAC) 6/17/82

                        ;       EQUATES

0000    =               BOOT    EQU     0000H
0005    =               BDOS    EQU     BOOT+05H

0001    =               CONIN   EQU     1
0002    =               CONOUT  EQU     2
0009    =               PRINT   EQU     9
000A    =               RDCON   EQU     10
000D    =               RESDISK EQU     13
000E    =               SELDISK EQU     14
000F    =               OPEN    EQU     15
0010    =               CLOSE   EQU     16
0011    =               SFIRST  EQU     17
0012    =               SNEXT   EQU     18
0013    =               DELETE  EQU     19
0014    =               READS   EQU     20
0015    =               WRITES  EQU     21
0016    =               MAKE    EQU     22
0017    =               RENAME  EQU     23
001A    =               SETDMA  EQU     26

0006    =               FBASE   EQU     BOOT+06H
005C    =               FCB1    EQU     BOOT+5CH
006C    =               FCB2    EQU     BOOT+6CH
0080    =               DMA     EQU     BOOT+80H
0100    =               TBASE   EQU     BOOT+0100H

000D    =               CR      EQU     0DH
000A    =               LF      EQU     0AH
000C    =               FF      EQU     0CH
0007    =               BELL    EQU     07H
0009    =               TAB     EQU     09H
001B    =               ESC     EQU     1BH

                        ;       JUMP TO MAIN ENTRY

0100                            ORG     TBASE

0100    C3AF24          CPMPIE  JMP     FRONT
0103    C3AF24                  JMP     FRONT           ;MAINTAIN PIE SHELL OFFSETS

                        ;       PIE'S TEXT BUFFER ADDRESSES

0106    004C            LOMEM   DW      4C00H           ;START OF TEXT (80 COL)
0108    00BF            HIMEM   DW      0BF00H          ;END OF TEXT BUFFER
```




**As the clock ticks down the game begins, and danger closes in . . .**

To survive you must leave base in search of the blots. The more blots you eat the more points you make.

But time is against you! You must return to a base before time runs out.

Can you get back?

The grid is filled with danger—get caught in a Gamma-Field and you are annihilated.
Roaming whirlers will blast you off the grid.

The grid will disintegrate before your eyes—but find the magic ring before it disappears and the grid will be restored.

What next?!

The longer you survive the more treacherous and slippery the paths become.

Beware. . . You are about to enter the SNAPPER zone.
For **The Snapper** see your Apple® dealer.

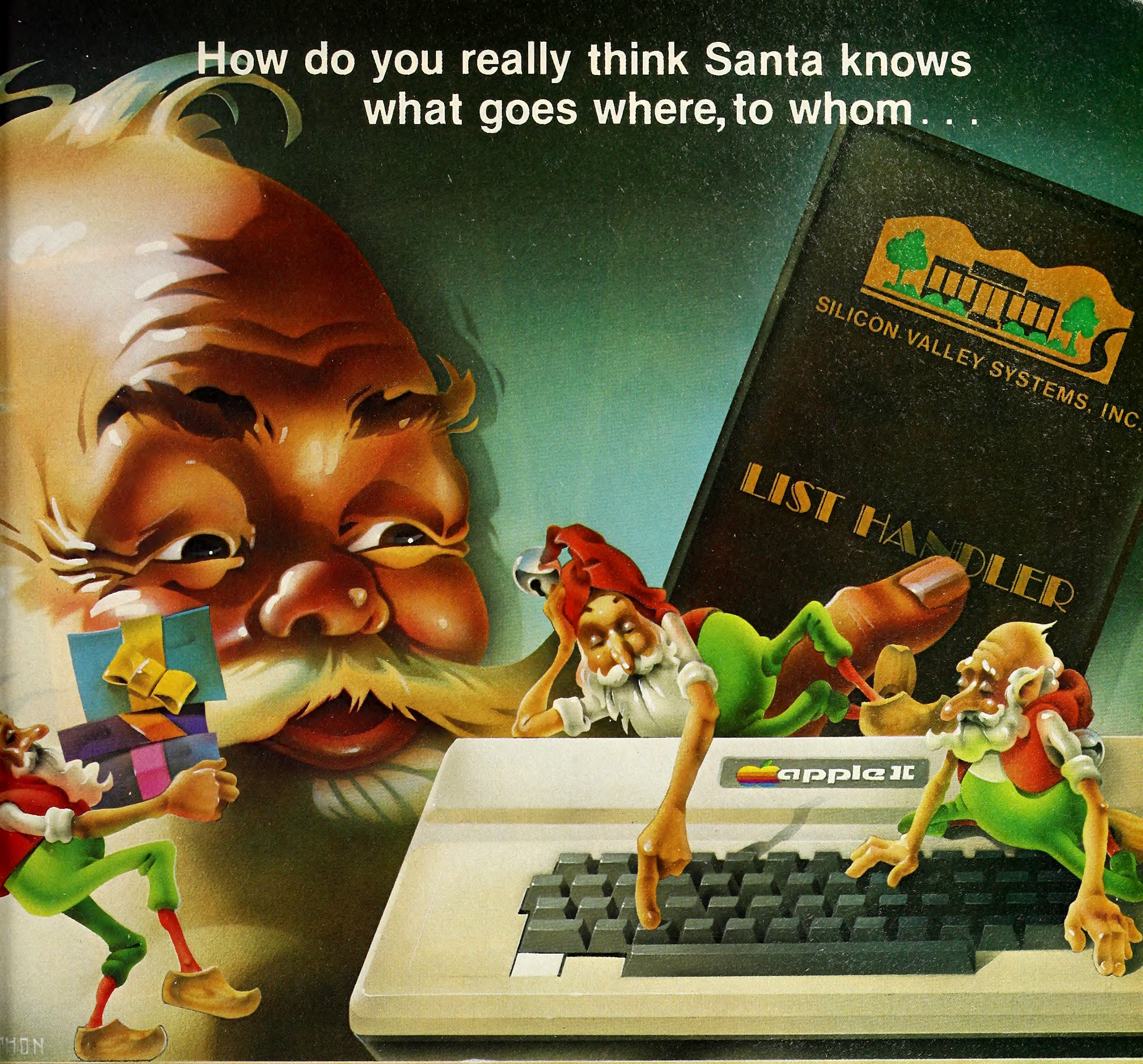
Simply the best. . .

Apple is a registered trademark of APPLE COMPUTER, INC.

**Silicon Valley Systems**

Silicon Valley Systems Inc. 1625 El Camino Real #4 Belmont, CA 94002 415-593-4344

```
1280    =           PIEINIT    EQU     1280H        ;PIE INITIALIZATION
1283    =           PIECOLD    EQU     1283H        ;PIE COLDSTART
1286    =           PIEWARM    EQU     1286H        ;PIE WARMSTART

F006    =           TXTS       EQU     0F006H       ;PIE LOW END OF TEXT
F002    =           TXTE       EQU     0F002H       ;PIE END OF TEXT
F004    =           TXTH       EQU     0F004H       ;PIE HIGH END OF TEXT
F000    =           REALTS     EQU     0F000H       ;REAL PIE LOW END OF TEXT

010A    00          FLAG       DB      00H          ;FLAGS
0080    =           ASM        EQU     80H          ;.ASM FILE, DO TABS

                    ;          RETURN HERE FROM FRONTEND, INVOKE PIE EDITOR

010B    3A0000      EDITOR     LDA     0000H        ;MODIFY FOR 80 COL DISABLE
010E    218612                 LXI     H,PIEWARM    ;SOFT ENTRY INTO PIE
0111    22D0F3                 SHLD    0F3D0H       ;WHERE TO GO
0114    2ADEF3                 LHLD    0F3DEH       ;GET 6502 SELECT ADDR
0117    77                     MOV     M,A          ;SELECT 6502

                    ;          BACKEND PROCESSOR

                    ;            FRONTEND PROCESSOR HAS LEFT "FCB1" WITH FILENAME
                    ;            OF OUTPUT FILE. "DMA" CONTAINS FILENAME IN
                    ;            PRINTABLE FORM (FILE.EXT$)

0118    114902                 LXI     D,SAVE       ;"SAVING..."
011B    0E09                   MVI     C,PRINT
011D    CD0500                 CALL    BDOS

0120    118000                 LXI     D,DMA        ;"FILE.EXT"
0123    0E09                   MVI     C,PRINT
0125    CD0500                 CALL    BDOS

0128    115C00                 LXI     D,FCB1       ;DELETE FILE JUST IN CASE
012B    0E13                   MVI     C,DELETE
012D    CD0500                 CALL    BDOS

0130    115C00                 LXI     D,FCB1       ;OPEN NEW FILE
0133    0E16                   MVI     C,MAKE
```




# LIST HANDLER

The List Handler is, **without doubt**, much more than a mailmerge program.

Presto! List Handler turns your Apple® into a quick response file cabinet with room to grow storage space. **With List Handler you can hold up to 3000 records per drive so that 24000 can be kept on-line at the same time with multiple disk drives.**

These records can be addresses if you choose, but here's where the versatility and flexibility come in.

Put what you like on those records!

- inventory
- form letters
- mailing lists
- invoices
- product lists
- short newsletters
- personnel directories
- labels

The list is endless.

**The List Handler will run on one or more drives, read and write DIF format and has unlimited sort fields.**

List Handler can stand alone or as an interface with your Word Handler (or most other word processors) for the most efficient word processing duo around.

The List Handler will surprise you, but the first surprise will be the price! At 89.95 the List Handler is the best and the only buy of its type around.

The enormity of Santa's job may have you skeptical of Santa's existence—but when you see the List Handler you might just believe in Santa again! See your Apple dealer for the List Handler.

**Simply the best . . .**

Apple is a registered trademark of APPLE COMPUTER, INC.

## Silicon Valley Systems

**Silicon Valley Systems Inc. 1625 El Camino Real #4 Belmont, CA 94002 415-593-4344**

# C.P.U. SOFTWARE



## The HUMAN FLY

Can you climb the World's Tallest Building? As you climb, the dangers increase.

Watch for angry police, menacing gorillas, earth shaking earthquakes, unpredictable birds, falling flower pots, and an occasional rising balloon (catch it and get a free ride up).

Elevation indicator shows where you are on the building.

Will you be a HUMAN FLY or a MOLE HILL CLIMBER? Find out in THE HUMAN FLY by Kevin Bagley.

HIRES action, SUPER SOUND, ASSEMBLY LANGUAGE, THE NEXT GENERATION OF COMPUTER GAMES.

48K APPLESOFT — 3.3 DOS

29.95 + shipping & handling or see your favorite dealer and ask for The HUMAN FLY by KEVIN BAGLEY.

## OIL RIG*

Are you the next 'J.R.' of the Oil industry??? Find out in OIL RIG!

An entirely new concept in gaming.

The Oil Market is constantly changing, even while you are typing in your transactions. Build your 'DYNASTY' to where you can prospect for oil, then drill for the moving pool of oil. Maybe even buy an offshore drilling platform.

Watch your cash flow — if your refinery explodes or your platform sinks, or??????????

9 skill levels and variable game lengths.

HIRES action, SUPER SOUND, ACTIVATED TEXT. ASSEMBLY LANGUAGE, THE NEXT GENERATION OF COMPUTER GAMES.

48K APPLESOFT — 3.3 DOS

29.95 + shipping & handling or see your favorite dealer and ask for OIL RIG By KEVIN BAGLEY

*Featured front page Wall Street Journal — April 22, 1982.

**COMPUTER PROGRAMS UNLIMITED**
9710 - 24th Ave. S.E. Everett, Washington 98204 (206) 337-5888

```
0135  CD0500            CALL   BDOS
0138  FEFF              CPI    0FFH         ;NO DIRECTORY SPACE?
013A  CA0A02            JZ     SAVERR       ;SAVE ERROR

                  ;     BEGIN WRITING TEXT TO DISK

013D  2A00F0            LHLD   REALTS       ;PROTECT LOCATION 0
0140  2206F0            SHLD   TXTS
0143  3A07F0            LDA    TXTS+1       ;CONVERT POINTERS TO Z80
0146  D610              SUI    10H
0148  3207F0            STA    TXTS+1
014B  3A03F0            LDA    TXTE+1
014E  D610              SUI    10H
0150  3203F0            STA    TXTE+1
0153  218000            LXI    H,DMA        ;START OF BUFFER
0156  220024            SHLD   BUFFP
0159  3E00              MVI    A,00H
015B  321424            STA    COLUMN       ;COL = 0
015E  321524            STA    BLC          ;BLANK COUNTER = 0

                  ;     LOOP TO COPY TEXT TO DMA BUFFER

0161  CD1502      SAVNG CALL   GETCHR       ;GET A CHARACTER
0164  320324            STA    CSAV
0167  3A1424            LDA    COLUMN
016A  3C                INR    A
016B  321424            STA    COLUMN       ;COL = COL + 1
016E  3A0A01            LDA    FLAG
0171  B7                ORA    A            ;ASM FILE?
0172  CAD101            JZ     SAVIT        ;NO, SKIP TAB PROCESSING

                  ;     INSERT TABS FOR .ASM FILES

0175  3A1424            LDA    COLUMN       ;AT TAB POSITION?
0178  E607              ANI    07H
```

# JIGSAW...A Mind Teaser

## The First Computerized Jigsaw Puzzle


Spend fascinating, intriguing, challenging hours
of endless, hi-res entertainment... with JIGSAW.
Great for anyone and everyone.

Take your favorite computer picture and convert it into a puzzle. The program comes with 7 supplied pictures (full-color, hi-res graphics). Each puzzle breaks apart into 180 separate pieces. Never repeats the same sequence of shapes–each game you randomly generate a brand-new puzzle.

The pictures supplied by JIGSAW are designed for players of all skill levels. Total program written in machine language, permits fast loading, quick picture generation, rapid response. A super teaching aid for young people (for size, color, and shape relationships).

**CAUTION: PLAYING JIGSAW MAY BE HAZARDOUS TO YOUR HEALTH.**
The fun and excitement are infectious and you'll spend hours in front of your computer!

This Program was Written by Joe Calabrese. JIGSAW requires Apple II, 48K. Now available at your local dealer.

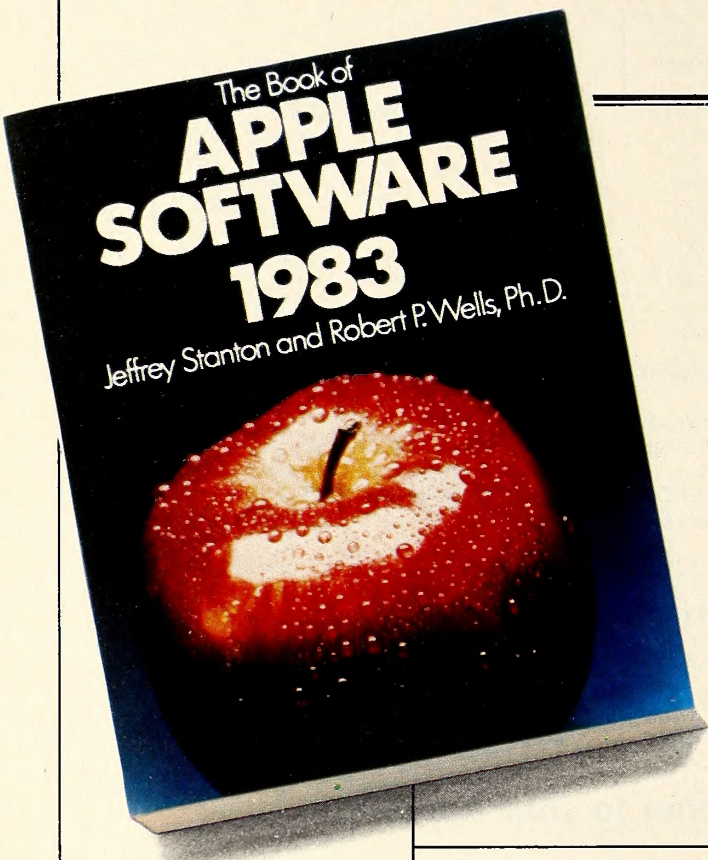
micro fun™
entertainment division
of micro lab

```
017A  FE01                CPI    01H
017C  C29001              JNZ    SAVNT     ;NO
017F  3A1524              LDA    BLC
0182  B7                  ORA    A         ;BLC = 0?
0183  CA9001              JZ     SAVNT     ;THEN NOTHING TO DO HERE
0186  3E09                MVI    A,TAB
0188  CD2002              CALL   PUTCHR    ;WRITE OUT A TAB FOR BLANKS
018B  3E00                MVI    A,00H
018D  321524              STA    BLC       ;BLC = 0 NOW

                   ;      EAT BLANKS UNTIL A TAB COLUMN IS HIT

0190  3A0A01       SAVNT  LDA    FLAG
0193  E607                ANI    07H       ;IN QUOTED STRING?
0195  C2AA01              JNZ    SUPRS     ;YES, SUPPRESS TAB INSERTION
0198  3A0324              LDA    CSAV
019B  FE20                CPI    20H       ;CHAR IS A BLANK?
019D  C2AA01              JNZ    SUPRS
01A0  3A1524              LDA    BLC
01A3  3C                  INR    A
01A4  321524              STA    BLC       ;IF SO, JUST COUNT IT
01A7  C36101              JMP    SAVNG     ;AND DON'T WRITE IT YET

                   ;      IF NON-BLANK ENCOUNTERED, GENERATE EATEN BLANKS

01AA  3A1524
01AA  3A1524       SUPRS  LDA    BLC
01AD  B7                  ORA    A
01AE  CAC101              JZ     NOBLNK    ;THERE ARE NONE EATEN
01B1  4F                  MOV    C,A       ;ELSE, GENERATE THAT MANY
01B2  3E20         GENBLK MVI    A,20H
01B4  C5                  PUSH   B
01B5  CD2002              CALL   PUTCHR    ;WRITE A BLANK
01B8  C1                  POP    B
01B9  0D                  DCR    C         ;IN A LOOP
01BA  C2B201              JNZ    GENBLK
01BD  79                  MOV    A,C
01BE  321524              STA    BLC       ;NO BLANKS EATEN NOW
```

```
                                 IF QUOTE ENCOUNTERED, TOGGLE TAB SUPPRESSION SWITCH

01C1   3A0324      NOBLNK    LDA     CSAV
01C4   FE27                  CPI     27H              ;SINGLE QUOTE?
01C6   C2D101                JNZ     SAVIT            ;NO, WRITE IT
01C9   3A0A01                LDA     FLAG
01CC   EE01                  XRI     01H              ;TURN ON/OFF SUPPRESSION
01CE   320A01                STA     FLAG

                             WRITE CHARACTER TO DMA BUFFER

01D1   3A0324      SAVIT     LDA     CSAV             ;RESTORE CHAR
01D4   B7                    ORA     A
01D5   CAF501                JZ      PUTEOF           ;END OF FILE?
01D8   CD2002                CALL    PUTCHR           ;WRITE THE CHAR
01DB   FE0D                  CPI     CR
01DD   C26101                JNZ     SAVNG
01E0   3E00                  MVI     A,00H
01E2   321424                STA     COLUMN           ;COLUMN BACK TO ZERO
01E5   3A0A01                LDA     FLAG
01E8   E680                  ANI     80H              ;TURN OFF QUOTE SUPPRESSION
01EA   320A01                STA     FLAG
01ED   3E0A                  MVI     A,LF             ;LF'S FOLLOW CR'S
01EF   CD2002                CALL    PUTCHR
01F2   C36101                JMP     SAVNG

                             FINISHED, WRITE EOF MARK AND CLOSE

01F5   3E1A        PUTEOF    MVI     A,1AH            ;CONTROL-Z TO END FILE
01F7   CD2002                CALL    PUTCHR
01FA   CD3202                CALL    DUMPIT

01FD   0E10                  MVI     C,CLOSE
01FF   115C00                LXI     D,FCB1           ;CLOSE THE FILE
0202   CD0500                CALL    BDOS
0205   FEFF                  CPI     0FFH
0207   C20000                JNZ     BOOT

020A   115302      SAVERR    LXI     D,ERRMSG         ;"ERROR"
020D   0E09                  MVI     C,PRINT
020F   CD0500                CALL    BDOS
0212   C30000                JMP     BOOT

                             GET A BYTE FROM PIE BUFFER

0215   2A06F0      GETCHR    LHLD    TXTS             ;GET POINTER
0218   7E                    MOV     A,M              ;GET NEXT BYTE
0219   E67F                  ANI     7FH              ;TURN OFF MSB
021B   23                    INX     H                ;BUMP POINTER
021C   2206F0                SHLD    TXTS
021F   C9                    RET

                             PUT A BYTE TO DMA BUFFER

0220   320224      PUTCHR    STA     ASAV
0223   2A0024                LHLD    BUFFP            ;GET BUFFER PTR
0226   77                    MOV     M,A              ;MOVE BYTE TO BUFFER
0227   23                    INX     H
0228   220024                SHLD    BUFFP
022B   3E01                  MVI     A,(DMA+80H)SHR(8) ;WHERE IT ENDS
022D   BC                    CMP     H                ;OVERFLOWED DMA?
022E   3A0224                LDA     ASAV
0231   C0                    RNZ

                             DMA BUFFER FULL, WRITE IT TO DISK

0232   218000      DUMPIT    LXI     H,DMA            ;START OF BUFFER
0235   220024                SHLD    BUFFP
0238   115C00                LXI     D,FCB1           ;DUMP BUFFER
023B   0E15                  MVI     C,WRITES
023D   CD0500                CALL    BDOS
0240   B7                    ORA     A
0241   3A0224                LDA     ASAV             ;RESTORE CHAR
0244   C8                    RZ
0245   E1                    POP     H                ;EXIT SUBROUTINE
0246   C30A02                JMP     SAVERR           ;AND STOP NOW

                             BACKEND DATA

0249   1B2A534156  SAVE      DB      ESC,'*SAVING $'
```

```
0253  070D0A4552  ERRMSG   DB    BELL,CR,LF,'ERR$'

                  ;        BE CAREFUL NOT TO LET BACKEND OVERLAP PIE EDITOR BELOW
0280                       ORG   TBASE+180H

        ------------------------------------------------------------
        |       PIE IMAGE LOGICALLY SITS HERE                      |
        |                                                          |
        |  PIE EDITOR IS AT $1280 (6502 ADDRESS MAP)               |
        |                OR $0280 (Z80  ADDRESS MAP)               |
        |                                                          |
        |  AND EXTENDS TO   $3300 (6502 ADDRESS MAP)               |
        |                OR $2300 (Z80  ADDRESS MAP)               |
        ------------------------------------------------------------

2400                       ORG   TBASE+2300H           ;ORG WAY PAST PIE IMAGE

                  ;        FRONTEND DATA

2400  0000        BUFFP    DW    0                     ;BUFFER POINTER
2402              ASAV     DS    1
2403              CSAV     DS    1
2404              OFILE    DS    16                    ;OUTPUT FILE NAME
2414  00          COLUMN   DB    0                     ;TAB COLUMN INDEX
2415  00          BLC      DB    0                     ;BLANK COUNTER

2416  0D0A43504D  HELLO    DB    CR,LF,'CPMPIE — APPLE PIE VERSION 2.0',CR,LF
2439  0D0A                 DB    CR,LF
243B  2843292043           DB    '(C) COPYRIGHT 1981',CR,LF
244F  2020202050           DB    '     PROGRAMMA INTERNATIONAL INC.',CR,LF
2471  0D0A                 DB    CR,LF
2473  46524F4E54           DB    'FRONT END BY DON WORTH (UCLA/OAC)',CR,LF
2496  0D0A0D0A24           DB    CR,LF,CR,LF,'$'

249B  204E4F5420  FILERR   DB    ' NOT FOUND$'

24A6  4C4F414449  LOADING  DB    'LOADING $'

                  ;        FRONTEND – READ FILE, INITIALIZE PIE EDITOR

24AF  216C00      FRONT    LXI   H,FCB2                ;COPY OUTPUT FILE NAME
```

# INTRODUCING




It's ironic that the latest breakthrough in the world of microcomputers doesn't use a single microchip or floppy disk.

It's The Book Company: a library of great books that will save you lots of time and money – whether you've owned your microcomputer five years or five days.

## GREAT REVIEWS

**The Book of Apple Software 1983 $19.95**

**Edited by Jeffrey Stanton, Robert P. Wells, Ph.D., and Sandra Rochowansky**

Software can be a significant investment. But many times it's difficult to compare similar programs to find exactly the program you want.

Enter *The Book of Apple Software 1983.* It's the only source that evaluates (not just lists) hundreds of programs available for Apple microcomputers.

Each evaluation is written by an expert in the type of program being reviewed, be it Utilities, Accounting, Education, Word Processing or Games and Entertainment. So you get top-notch, impartial information.

Not only does each evaluation tell you all the hard facts (like price, hardware requirements, language, etc), it also gives you a letter grade (A through F) in categories like Ease of Use, Reliability, Creativity, Visual Appeal and, most importantly, Value for Your Money.

What's more, the reviews are readable and entertaining, so you'll not only spend more wisely on software, you'll have a good time doing it.

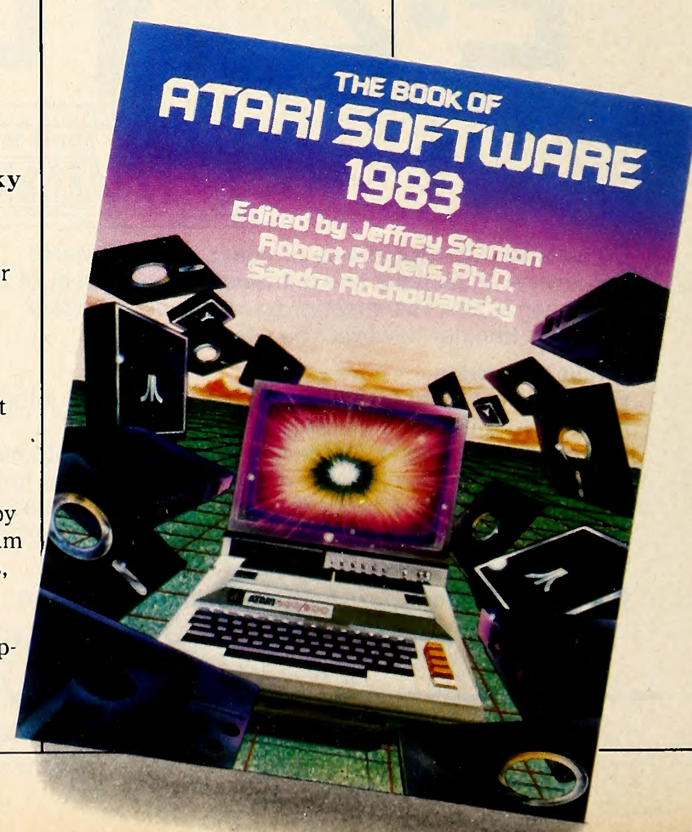
**The Book of Atari Software 1983 $19.95**

**Edited by Jeffrey Stanton, Robert P. Wells, Ph. D., and Sandra Rochowansky**

Attention Atari 400/800 or 2600 (VCS) owners. Atari has been famous for games, but they also offer a broad selection of software in other areas, such as Business, Education, and Word Processing.

Designed as a companion volume to the best-selling *Book of Apple Software, The Book of Atari Software 1983* contains hundreds of incisive reviews. Each evaluation features a concise description of the program, plus a letter grade rating system (A through F), so you'll know at a glance whether or not a particular program is right for you. And it will help to save you money along the way.

Pick up *The Book of Atari Software 1983* soon. It's not just the best consumer guide to Atari software. It's the *only* one.




Apple is a registered trademark of Apple Computer, Inc., Atari is a registered trademark of Atari, Inc.,

# BOOKNOLOGY.

## FUN AND GAMES

**Apple Graphics and Arcade Game Design**
**By Jeffrey Stanton**
**$19.95**

It's finally available. Everything you wanted to know about creating arcade games – from Space Invaders to Pacman – but didn't know who to ask.


Jeffrey Stanton takes you from game concept through Lo-Res and Hi-Res color graphics at the machine language level. And he gives you a thorough grounding in the Apple's screen architecture and the advantages of bit-mapped design.

Using flow charts and working examples he discusses scoring, laser fire, explosions, and bomb drops in both single screen and scrolling games.

This is the "must-have" book for anyone who wants to understand and create a computer game.

## PROFITABLE PROJECTIONS

**WHAT IF...? A Guide to Computer Modeling**
**By Tom Simondi**
**$19.95**

Almost every practical or business application of computers involves a complex scheme of projections called a model. Which explains why modeling programs are becoming extremely popular with microcomputer owners.

*WHAT IF...?* tells you about these powerful programs in terms that even the new computer-user can understand. You get a general explanation of model building, a comparison of the most popular modeling packages on the market (like VisiCalc and SuperCalc), and all the tools you need to build your own modeling programs. So you'll be able to create models for everything from real estate analyses to cash-flow projections for your own company.

We can't guarantee that you'll become a millionaire over-night, but understanding models sure won't hurt your chances.


THE BOOK COMPANY
A Division of Arrays, Inc.

Available at computer stores everywhere or directly from THE BOOK COMPANY.

☐ The Book of Apple Software 1983, $19.95
☐ The Book of Atari Software 1983, $19.95
☐ Apple Graphics and Arcade Game Design, $19.95
☐ WHAT IF...? A Guide to Computer Modeling, $19.95
(Place quantity in box if ordering more than one of each title.)

Visa, MasterCard, Check or Money Order excepted.

☐ Visa #__________(expiration date)______

☐ MasterCard #__________(expiration date)______

Signature__________

Total amount enclosed $______. (California residents add 6% sales tax. Postage will be paid by THE BOOK COMPANY.)

Name__________

Address__________

City__________

State__________Zip______

Mail to: THE BOOK COMPANY, 11223 South Hindry Avenue, Los Angeles, CA 90045. Or, call toll-free (800) 421-3931, or, in California call collect (213) 417-3003.

```
24B2  110424           LXI   D,OFILE
24B5  315FCB           LXI   SP,0CB5FH       ;DDT DOESN'T DO THIS
24B8  0E10             MVI   C,16
24BA  CD6126           CALL  MOVE
24BD  3E00             MVI   A,00H
24BF  327C00           STA   FCB1+32         ;CR=0
24C2  326800           STA   FCB1+12         ;EXT=0

                 ;     CHECK INPUT FILENAME FOR .ASM SUFFIX

24C5  3E00             MVI   A,00H           ;ASSUME NOT ASM FILE
24C7  320A01           STA   FLAG
24CA  3A6500           LDA   FCB1+9          ;SUFFIX IS ASM?
24CD  FE41             CPI   'A'
24CF  C2EC24           JNZ   SAYHI
24D2  3A6600           LDA   FCB1+10
24D5  FE53             CPI   'S'
24D7  C2EC24           JNZ   SAYHI
24DA  3A6700           LDA   FCB1+11
24DD  FE4D             CPI   'M'
24DF  C2EC24           JNZ   SAYHI
24E2  3E80             MVI   A,ASM           ;MARK IT AS .ASM FILE
24E4  320A01           STA   FLAG
24E7  3E00             MVI   A,00H
24E9  329402           STA   294H            ;UC LOCK FOR ASM FILE

                 ;     GREETING

24EC  111624     SAYHI LXI   D,HELLO         ;"APPLE PIE 2.0 ETC."
24EF  0E09             MVI   C,PRINT
24F1  CD0500           CALL  BDOS

                 ;     INITIALIZE PIE

24F4  218012           LXI   H,PIEINIT       ;CALLING PIE INITIALIZATION
24F7  22D0F3           SHLD  0F3D0H
24FA  2ADEF3           LHLD  0F3DEH          ;6502 SELECT ADDRESS
24FD  77               MOV   M,A             ;INVOKE 6502
24FE  2A00F0           LHLD  REALTS          ;PROTECT LOCATION 0 FROM CP/M
2501  2206F0           SHLD  TXTS

                 ;     SEE IF INPUT FILE EXISTS

2504  115C00           LXI   D,FCB1
2507  0E11             MVI   C,SFIRST
2509  CD0500           CALL  BDOS
250C  FEFF             CPI   0FFH            ;FILE NOT FOUND?
250E  C23A25           JNZ   FOUND

                 ;     INPUT FILE NOT FOUND

2511  115C00     NOTFND LXI  D,FCB1          ;PRINT FILE NAME
2514  CD3526           CALL  PRFILE
2517  119B24           LXI   D,FILERR        ;" NOT FOUND"
251A  0E09             MVI   C,PRINT
251C  CD0500           CALL  BDOS
251F  3A0524           LDA   OFILE+1         ;OUTPUT FILE MISSING?
2522  FE20             CPI   20H
2524  C20000           JNZ   BOOT            ;NO, MAKES NO SENSE
2527  3A5D00           LDA   FCB1+1
252A  FE20             CPI   20H             ;INPUT FILE NAME MISSING?
252C  CA0000           JZ    BOOT            ;HE IS IN ERROR
252F  2A06F0           LHLD  TXTS
2532  2202F0           SHLD  TXTE            ;TEXT END = TEXT START
2535  3600             MVI   M,00H           ;AND EOF AT FIRST BYTE
2537  C3E525           JMP   LOADED          ;PRETEND FILE IS LOADED

                 ;     INPUT FILE FOUND, OPEN IT

253A  11A624     FOUND LXI   D,LOADING       ;"LOADING "
253D  0E09             MVI   C,PRINT
253F  CD0500           CALL  BDOS

2542  115C00           LXI   D,FCB1          ;PRINT FILENAME
2545  CD3526           CALL  PRFILE

2548  115C00           LXI   D,FCB1          ;OPEN THE FILE
254B  0E0F             MVI   C,OPEN
254D  CD0500           CALL  BDOS
2550  FEFF             CPI   0FFH            ;HOW CAN THIS HAPPEN?
```

```
2552   CA1125                 JZ     NOTFND
2555   210001                 LXI    H,DMA+80H       ;BUFFER IS EMPTY
2558   220024                 SHLD   BUFFP
255B   3A07F0                 LDA    TXTS+1
255E   D610                   SUI    10H             ;CONVERT TXTS TO Z80
2560   3207F0                 STA    TXTS+1
2563   3E00                   MVI    A,00H
2565   321424                 STA    COLUMN          ;COLUMN = 00

                     ;        COPY BYTES IN A LOOP FROM FILE TO MEMORY

2568   CD0C26      READING    CALL   RDCHR           ;READ A CHARACTER
256B   FE1A                   CPI    1AH             ;CONTROL-Z?
256D   CAC425                 JZ     ATEOF           ;YES, END OF FILE
2570   FE0A                   CPI    LF              ;LINE FEED?
2572   CA6825                 JZ     READNG          ;YES, SKIP IT
2575   320224                 STA    ASAV            ;SAVE CHARACTER
2578   3A1424                 LDA    COLUMN
257B   3C                     INR    A
257C   321424                 STA    COLUMN          ;BUMP COLUMN POINTER
257F   3A0224                 LDA    ASAV
2582   FE0D                   CPI    CR              ;CARRIAGE RETURN
2584   C28C25                 JNZ    NOTCR           ;NO
2587   3E00                   MVI    A,00H
2589   321424                 STA    COLUMN          ;COLUMN = 0 AGAIN

                     ;        EXPAND TABS TO 8 COLUMN POSITIONS

258C   3A0A01      NOTCR      LDA    FLAG
258F   B7                     ORA    A
2590   CAB925                 JZ     STORET          ;NO TAB PROCESSING
2593   3A0224                 LDA    ASAV
2596   FE09                   CPI    TAB
2598   C2B925                 JNZ    STORET          ;TAB CHARACTER
259B   3A1424                 LDA    COLUMN
259E   E607                   ANI    07H             ;0-7
25A0   2F                     CMA                    ;COMPLIMENT
25A1   3C                     INR    A
```

```
25A2  E607              ANI   07H
25A4  3C                INR   A                ;TO GET NUMBER OF BLANKS
25A5  4F                MOV   C,A
25A6  3EA0      PADD    MVI   A,0A0H           ;PADD TO NEXT TAB POSITION
25A8  CD2C26            CALL  WRCHR
25AB  0D                DCR   C
25AC  CA6825            JZ    READNG           ;DONE
25AF  3A1424            LDA   COLUMN
25B2  3C                INR   A
25B3  321424            STA   COLUMN           ;COUNT COLUMNS
25B6  C3A625            JMP   PADD

                ;       STORE CHARACTERS AND CONTINUE

25B9  3A0224    STORET  LDA   ASAV             ;RESTORE CHARACTER
25BC  F680              ORI   80H              ;MSB ON FOR PIE
25BE  CD2C26            CALL  WRCHR            ;COPY CHAR TO MEMORY
25C1  C36825            JMP   READNG

                ;       END OF FILE

25C4  3E00      ATEOF   MVI   A,00H            ;MARK END OF BUFFER
25C6  CD2C26            CALL  WRCHR
25C9  115C00            LXI   D,FCB1
25CC  0E10              MVI   C,CLOSE
25CE  CD0500            CALL  BDOS             ;CLOSE FILE
25D1  3A07F0            LDA   TXTS+1
25D4  C610              ADI   10H              ;CONVERT TXTS TO Z80
25D6  3203F0            STA   TXTE+1
25D9  3A06F0            LDA   TXTS
25DC  3202F0            STA   TXTE
25DF  2A0601            LHLD  LOMEM            ;RESTORE START OF TEXT
25E2  2206F0            SHLD  TXTS

                ;       FILE LOADED, SET UP FCB FOR SAVE

25E5  3A0524    LOADED  LDA   OFILE+1          ;GET OUTPUT FILENAME
25E8  FE20              CPI   20H              ;BLANK?
25EA  CAF825            JZ    GOTFNM           ;YES, USE INPUT NAME

25ED  210424            LXI   H,OFILE          ;COPY OUTPUT FILE NAME
25F0  115C00            LXI   D,FCB1
25F3  0E10              MVI   C,16
25F5  CD6126            CALL  MOVE

25F8  CD4026    GOTFNM  CALL  FNTDMA           ;FILENAME TO DMA
25FB  3E00              MVI   A,00H
25FD  327C00            STA   FCB1+32          ;CR=0
2600  326800            STA   FCB1+12          ;EXT=0
2603  2A06F0            LHLD  TXTS
2606  2200F0            SHLD  REALTS           ;RESTORE PIE'S LOC 0
2609  C30B01            JMP   EDITOR           ;GO TO PIE

                ;       RDCHR: READ A CHARACTER FROM FILE

260C  2A0024    RDCHR   LHLD  BUFFP            ;BUFFER POINTER
260F  3E01              MVI   A,(DMA+80H)SHR(8)
2611  BC                CMP   H
2612  CA1B26            JZ    REFILL           ;BUFFER IS EMPTY
2615  7E                MOV   A,M              ;GET NEXT BYTE
2616  23                INX   H
2617  220024            SHLD  BUFFP            ;BUMP BUFFER POINTER
261A  C9                RET

                ;       BUFFER IS EMPTY, READ NEXT ONE

261B  218000    REFILL  LXI   H,DMA            ;RESTORE BUFFER PTR
261E  220024            SHLD  BUFFP
2621  115C00            LXI   D,FCB1
2624  0E14              MVI   C,READS          ;READ NEXT SEQ
2626  CD0500            CALL  BDOS
2629  C30C26            JMP   RDCHR

                ;       WRCHR: PUT NEXT BYTE IN TEXT BUFFER

262C  2A06F0    WRCHR   LHLD  TXTS
262F  77                MOV   M,A              ;MOVE TO NEXT PLACE
2630  23                INX   H
2631  2206F0            SHLD  TXTS             ;BUMP PTR
2634  C9                RET
```

```
                      ;          PRFILE: PRINT FILENAME

2635   CD4026   PRFILE   CALL    FNTDMA        ;FILENAME TO DMA
2638   118000            LXI     D,DMA
263B   0E09              MVI     C,PRINT
263D   C30500            JMP     BDOS          ;EXIT THRU PRINT

                      ;          FNTDMA: MOVE FCB1 FILENAME TO DMA

2640   215D00   FNTDMA   LXI     H,FCB1+1
2643   118000            LXI     D,DMA
2646   0E08              MVI     C,8
2648   CD6126            CALL    MOVE
264B   3E2E              MVI     A,'.'
264D   328800            STA     DMA+8
2650   216500            LXI     H,FCB1+9
2653   118900            LXI     D,DMA+9
2656   0E03              MVI     C,3
2658   CD6126            CALL    MOVE
265B   3E24              MVI     A,'$'
265D   328C00            STA     DMA+12
2660   C9                RET

                      ;          MOVE SUBROUTINE
                      ;            HL = INPUT PTR
                      ;            DE = OUTPUT PTR
                      ;            C  =LENGTH

2661   7E       MOVE     MOV     A,M
2662   12                STAX    D
2663   23                INX     H
2664   13                INX     D
2665   0D                DCR     C
2666   C26126            JNZ     MOVE
2669   C9                RET

266A                     END
```

# THE ANIMATED APPLE

## With GraForth

## Part 3

BY PAUL LUTUS AND PHIL THOMPSON

Last month's Maxwell animation can be broken down into a few simple steps: Three separate images are displayed, each followed by a short pause, altogether showing Maxwell take one step. The area that will not be overwritten by the next step is erased, and the block is positioned one character space to the left. This process is repeated and Maxwell takes one step after another, moving to the left.

Two aspects of this animation are worth exploring: the length of the pause and the number of images displayed before repeating the cycle. Both of these contribute to the overall smoothness of the animation.

The question of how long to pause between images is, of course, very subjective. In most cases, some kind of pause will be needed to view the image being displayed before it is erased or overwritten. A ghost effect can be created by printing and immediately erasing a character block. Too long a pause can also cause problems. It allows people to notice the individual steps in the animation, changing it from a movielike motion to a rapid slide show. The pause in the Maxwell animation is about a tenth of a second. Depending on the application, a good pause between images will usually range from one twenty-fifth to one quarter of a second. The pause can be created by drawing other graphics elsewhere on the screen, using a sound effect, or simply wasting time with an empty loop.

The second, and more important, contributor to smooth animation is how far to move the object before redisplaying it. (We'll call this the *step size.*) Obviously an image that moves a third of the way across the screen with each frame will not be very smooth. The most convenient step size to use is one character width (assuming we're moving horizontally). Here the process is simple: draw the block, wait, erase any portions of the image that will not be replaced by the next block, reposition one character space over, and repeat. Only one block image is needed, the program is short, and the animation is reasonably smooth.

The quality of the animation can be improved by using a step size smaller than one character width, however. Since character blocks can only be printed on character boundaries, two or more separate images are used, each subsequent image shifted a portion of a character width. This is what is done in the Maxwell demonstration. Three separate blocks are displayed on each character position, and each image shows Maxwell about a third of a character position farther to the left. Therefore, the step size is a third of a character width. You may want to have another look at the Maxwell images from the character editor. Type:

```
READ " CHAREDITOR "
HOME RUN
```

Get the file Chr.Maxwell, select a Blocksize of 2 by 3, then Read character numbers 64, 70, and 76, one after another. You can see that Maxwell moves to the left with each image while the block itself is printed in the same place.

Since there are seven pixels in the width of one character, the limit to step size is seven images per character position. With this step size, a new image could be drawn starting on every pixel. When using smaller step sizes, the wait pauses can usually be of a shorter duration, since each image will not be very different or far away from the previous one.

Step sizes need to be considered before creating the character sets. Suppose you want to create an image that fills a three by two character block and moves three steps per character width. Three separate blocks are needed to show the image in each of the three positions. If you draw


CALIFORNIA
MOVETO

your three by two image in the character editor using a three by two block, there will be no room within the block to shift the image from side to side. The solution is to define a block that is four characters wide, rather than three. The first image will leave some free space on the left, the next will be centered, and the last will leave space on the right.

If you wish to create duplicate images in the character set that are offset by a few pixels, redrawing each image can be time consuming. Fortunately, the combination of a new routine and a feature built into the character editor can make the task much easier.

When an image created in the character editor is saved into a character set, the editor reads the image directly off of the high-resolution screen. This means that anything that can be drawn in the upper-left corner of the screen can be written into a character set by the character editor. This is exactly why the editor does not automatically erase the screen as it starts up. It allows you to place an image on the screen, run the editor, and save the image as a character block. The screen area used starts at *1 vtab 1 htab*, which is the hi-res point (7, 8), and extends to the right and downward according to the block size selected in the editor.

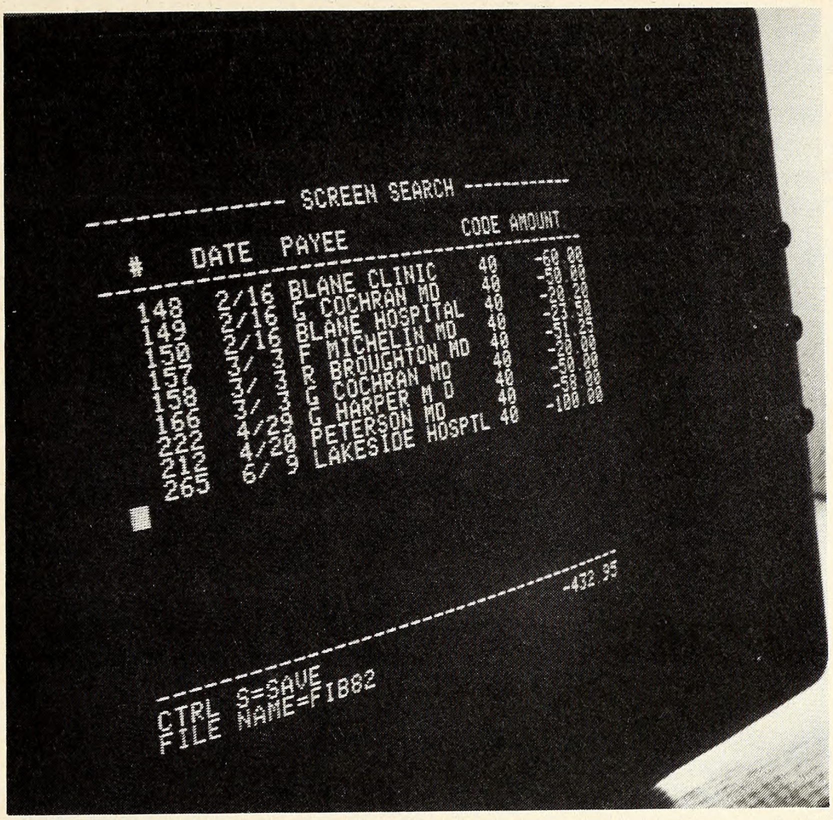
If we had a way of plotting character blocks starting on any pixel (instead of just on character boundaries), then we could save the shifted image back into the character set. Below is a routine to do just that. It reads a given character block from memory and plots it point by point at any position on the screen. Type *edit* to enter the GraForth text editor, type in the word definitions, save them to disk, then compile them into the word library. Typing the comments is, of course, optional.

```
VARIABLE BX          ( Horizontal block size )
VARIABLE BY          ( Vertical block size )
VARIABLE CPX         ( Horizontal pixel start for character )
VARIABLE CPY         ( Vertical pixel start for character )
VARIABLE BPX         ( Horizontal pixel start for block )
VARIABLE BPY         ( Vertical pixel start for block )
VARIABLE CN          ( Character number )
VARIABLE ADR         ( Address of character set )
: PIXELCHAR
8 0 DO                          ( 8 lines / char )
  DUP I + PEEK                  ( Read char byte )
  DUP 128 AND                   ( Check & )
  IF 7 ELSE 3 THEN COLOR        ( Set color bit )
  7 0 DO                        ( 7 pixels / line )
    DUP 2 MOD IF                ( If bit is on, plot it )
      CPX I + CPY J + PLOT
    THEN
    2 /                         ( Next bit, please )
  LOOP
  DROP
LOOP
DROP ;
: PIXELBLK
BY 0 DO                         ( Vertical loop )
  BX 0 DO                       ( Horizontal loop )
    I 7 * BPX + -> CPX          ( Set char X position )
    J 8 * BPY + -> CPY          ( Set char Y position )
    J BX * CN + I +             ( Find char # )
    8 * ADR +                   ( Find char address )
    PIXELCHAR                   ( Draw the char )
  LOOP
LOOP ;
```

Two routines are included here. *Pixelchar* plots a single character starting at any pixel, and *Pixelblk* calls *Pixelchar* to plot a block of characters. Single characters can also be plotted with *Pixelblk* by selecting a block size of one by one.

To run *Pixelblk:*

1. Load the character set.
2. Set:
   BX to the horizontal block size.
   BY to the vertical block size.
   BPX to the desired starting X coordinate for the block.
   BPY to the desired starting Y coordinate.
   ADR to the character set address.
   CN to the starting character number.




# New, Fun Money Manager

**The new Double Check!
It balances your checkbook and creates a valuable library of financial facts.**

**Sort checks or deposits by category codes.
See totals for expenditures or income.
Get 100 user-defined categories.**

You'll see totals for anything: household budgets, financial forecasts, or tax items. See detailed listings or get totals only. Either way you'll get up-to-date information about making and spending money. You'll save time and money at tax time too.

More: if you own a small business, you can generate income statements, get sales by category, or see cash flow. Best of all, most accounting is an automatic result of checkbook balancing. You can put Double Check in a non-checkbook mode and create "aged accounts receivables", assets, or petty cash accounts. It also works for credit card control.

**THE EASIEST BALANCER YET**

On your computer screen, you see a facsimile of your checkbook. You see 17 items per screen; you can scroll for more. You'll see checks, deposits, plus a running balance. . . it's a computerized replica of your checkbook. Balancing is easy because checkbook errors appear immediately.

Bank statement balancing is even easier. Just enter the numbers of "cleared checks". Instantly, the computer displays all the matching data (date, payee, amount, etc.). Compare with statement totals for super-fast balancing.

**POWERFUL CODING SYSTEM**

Double Checks coding system is simple, yet very powerful. As the computer balances your checkbook, you give each check or deposit its own category code. In the process, you build files of financial information which you can sort, list, or total.

Thus, Double Check creates a unique financial data base. You get totals for anything. You can label, change, or delete categories anytime. . . without effecting data. We even give you an on-screen category dictionary: Just press "Ctrl-O"; you'll see code numbers plus your labels.

You get 100 categories. . . enough for budgets or bookkeeping. You can sort by any category or sort by month. You can list the entire file, or selected codes. You can print it out or display on your screen.

- **Handles one checking account, or dozens.**
- **Split checks (or deposits) between categories.**
- **No complex budget set-ups; no monthly closings.**

**OUTSTANDING TAX TOOL!**

Get up-to-the-minute data anytime during the current tax year... while you still have time to act. Perfect with Howardsoft, Micro Lab, and other tax preparers.

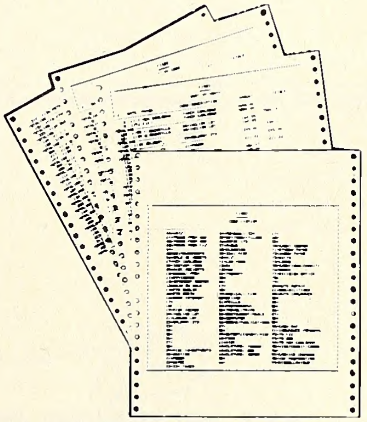
**YOU'RE IN CONTROL**

- **Edit anything, anytime.**
- **Fast, clean machine language.**
- **Crash proof program; no traps.**
- **Plain-English manual, no computer jargon.**

**NOT SOLD IN STORES**

We ship directly to you via first-class mail. We ship fast and pay postage. Back up disks are available for $20, also postage paid.

- **Free tutorial disk included. You'll learn the the program in thirty minutes.**
- **Works with one drive or two, with or without a printer.**
- **Tax deductible (if used to track income and deductions. See IRS publication #17, page 98).**



Seven ready-to-print reports: code dictionary, cleared and uncleared checks, lists by category, grand totals, and more.

Why not give Double Check a try? Order now and we'll include a free tutorial disk. You also get our money-back guarantee: If Double Check doesn't do everything we say, return it. We'll give you a prompt, courteous refund.

**INTRODUCTORY PRICE $49.95**

Price includes master disk, tutorial disk, and 30-page manual. For Apple II+, 48K, 3.3 DOS. Visa, Mastercharge, C.O.D., add $2.00. Phone 24 hours:

**800-648-4780**

**or mail check to:
Computer Tax Service
Box 7915, Incline Village, NV 89450**

3. Call *Pixelblk*. The block will be drawn on the screen.

The variables you need to set are not changed when *Pixelblk* is run, so they do not need to be reset every time.

Let's run the whole pixel-shift-and-save procedure through, assuming GraForth has just been booted. Since Maxwell is already a shifted set, we'll shift one of the helicopter images from the Chr.Stuff character set (discussed on page 7-9 of the GraForth manual), so that it starts midway between two character positions:

1. Read the pixel-shifter routines onto the word library. Use the file name you saved the pixel-shifter with:

```
READ "  filename  "
```

2. Read the character editor in above the pixel-shifter:

```
READ " CHAREDITOR "
```

3. Load the character set:

```
CR 132 PUTC PRINT " BLOAD CHR.STUFF,A2816 " CR
```

4. Set the appropriate variables:

```
5 -> BX 3 -> BY        ( 5 x 3 character block )
8 -> BPY               ( Start at correct vertical
                         coordinate for character editor )
11 -> BPX              ( Start at horizontal coordinate + 4
                         more to shift it 4 pixels
                         (7+4=11) )
2816 -> ADR            ( Set character set address )
33 -> CN               ( Set character number for first
                         helicopter )
```

5. In one line, erase the screen, draw the block, and enter the character editor:

```
ERASE PIXELBLK MAIN    ( MAIN is the word that runs the
                         character editor. )
```

6. Now that you're in the character editor, set a block size wide enough to fit the new image. Press B:

```
Enter Block Horizontal Size :6
Enter Block Vertical Size :3
```

7. Write the image into the character set. We're going to overwrite the three happy-face images here. Press W:

```
Enter character number
to be written : 78
```

From here you can either edit the image further or save the character set to disk. Pixel-shifting can also come in handy for changing the colors in character shapes, as we'll see in a few moments.

GraForth can also display characters in eight different sizes, using the word *chrsize*. Chrsize removes a number from the stack to determine what size subsequent characters will be displayed in. A character size of zero selects the normal text display. Character sizes one through eight use a different method for displaying characters; each dot, or pixel, of the character is plotted as a small rectangle, similar to the rectangles created with the fill command. Typing *1 chrsize* results in a normal size display; *8 chrsize* draws characters eight times larger.

The rectangle-type character plotting allows some additional capabilities. The color of the characters can be selected with the word *color*, while the normal text display will only display color if the characters were created with color. The larger character sizes can also be used in GraForth's *exclusive-or* mode (*exmode*), so that characters can be plotted over graphics and then erased without disturbing the underlying graphics! The price paid for these features is speed: The normal display can print characters much faster. The larger sizes don't have the speed necessary for smooth, animated character graphics. They are best used for displaying assorted messages or still character images.

Let's have a look at how GraForth keeps track of the characters on the screen, and how this affects character display. When your Apple is in text mode (no graphics), all of the characters on the screen are stored as ASCII values in a reserved area of memory. The hardware continuously reads these values and creates a character video display from them.

The hi-res graphics mode uses one of two other areas in memory for its display. Here, each dot on the screen is stored as a bit in memory. Changing the bits changes the display.

When GraForth displays a character on the hi-res screen, it stores the ASCII value of the character in the unseen text memory space. This allows GraForth to keep track of what characters are where without having to read through a lot of hi-res memory. It then looks up the shape of the character in the current character set and writes that shape into the hi-res space, which causes that character to appear on the screen.

As mentioned on pages 7-14 and 7-15 of the GraForth manual, before GraForth decides to print a character, it first checks the text area to see if that character is already on the screen. If it is, then the character is not reprinted. This speeds hi-res scrolling considerably. However, character set changes and unblk commands do not affect the contents of the text page and can produce discrepancies between what you see and what you get. This is best clarified with an example. Enter the following lines:

```
CR 132 PUTC PRINT " BLOAD CHR.SLANT,A2816 " CR
2816 CHRADR
HOME PRINT " ONE LINE " CR
CHRSET CHRADR
0 VTAB PRINT " TWO LINES " CR
```

We first printed the line "one line" using the slant character set, then printed the line "two lines" over the top of it using the standard character set. Note that the "line" in "lines" is still in the slant set. GraForth checked the text screen, and since it found the characters already there, it did not reprint them in the new character set.

One solution is to clear the text memory. If this is done, GraForth will not find any identical characters and will always reprint. The command *−936 call* will clear the text window to spaces without affecting the graphics screen:

```
2816 CHRADR
HOME PRINT " ONE LINE " CR
CHRSET CHRADR
-936 CALL 0 VTAB PRINT " TWO LINES " CR
```

The call cleared the text window and allowed the entire line to be reprinted in the new character set.

If you ever find characters not printing when you think they should, then their ASCII values are probably already in the text memory, preventing them from being reprinted. This is especially true if you want to print a space character (ASCII 160, or character number 0) that has been redefined to be a visible graphics image, since there are many space characters already lurking in the text page of a screen that is mostly blank.

The large size characters also use the text screen. This means that (for example) if you print a normal size character at *2 vtab 1 htab*, and then print a character in *8 chrsize* also at *2 vtab 1 htab*, GraForth will lose track of the first character even though the characters occupy different portions of the hi-res screen. This won't cause any problems unless you try to scroll or reprint the character.

An amusing and enlightening effect can be created by going into text mode before running a program that uses character graphics. All character blocks will be printed simply as groups of letters on the text screen. The following example runs the character graphics portion of the GraForth demonstration program from text mode. If you've been following the above examples, you'll first need to clear the character editor from memory with *forget X* to prevent some word names from being duplicated.

```
READ " GRAPHICS2 "
TEXT RUN
```

Press reset when you want to exit; otherwise, the demo will continue.

Another aspect of animated character graphics worth exploring is color. The larger character sizes allow you to select color as characters are printed, but, as mentioned above, the slower speed can get in the way. The best method is to use the normal character display and design the color right into the character sets. This also means that intricate color patterns can be used, rather than a single color. Using the GraForth character editor, you can select colors as you're creating the image, subject to the usual color limitations of the Apple.

Pages 19 and 20 of the *Apple II Reference Manual* discuss the basics behind the Apple color limitations, and the *Softalk* column Assembly Lines explored the subject in depth recently. Let's take some time out here to look more closely at how the Apple stores pixels and keeps track of color in memory. Then we'll see how it affects character display in GraForth.

The pixels in each horizontal row of the Apple screen are stored in memory in groups of seven, one group per byte of memory. A single bit in each byte determines what colors the pixels in the group can have. A byte can contain either green and violet pixels or orange and blue pixels. If for some reason the color bit is changed, then all of the pixels in that byte will change color. If any two adjacent pixels are plotted, their colors will combine to form white. A true white is always two pixels wide.

Here's how to determine the color of an individual pixel: If the color bit is set to one (orange and blue pixels), then a pixel will be orange if it is in an odd numbered column, and blue if in an even numbered column. If the color bit is cleared to zero (green and violet pixels), then a pixel in an odd column will be green, and a pixel in an even column will be violet. Thus the actual color of an individual pixel is determined by the setting of the color bit for the byte in memory the pixel occupies and the column the pixel is plotted in.

For most kinds of plotting, GraForth takes care of all of this for you. If you plot, for example, a green dot, GraForth will clear the color bit to zero, then check what column the dot is being plotted in. If the column is odd, GraForth will simply plot the pixel. If the column is even, GraForth will automatically move the pixel one column to the right, since green dots cannot lie in even columns. In general, when a color is specified, GraForth shifts the dots if necessary; if the color is set to white, the dots are always left in place.

Let's try a few examples. If you haven't specified a color, GraForth will use white (*3 color*, which has its high bit cleared to zero). Enter:



```
0 40 18 24 WINDOW ERASE
0 10 PLOT
```

A violet dot appears, since it was plotted in an even column.

```
1 10 PLOT
```

Another dot plotted adjacent to it changes it to white.

```
1 15 PLOT
```

A single dot in this column is green.

```
1 COLOR
0 20 PLOT
```

We're forcing this dot to be green. Since a green dot could not be plotted in an even column, GraForth moved it into column 1.

```
5 COLOR
5 20 PLOT
```

Surprise! Plotting an orange dot near the green dot changed the green dot to orange, too. Here's why: Since orange was specified, GraForth set the color bit for that byte to one. But the green dot occupies the same byte (the same group of seven pixels) and required the color bit to be zero to keep it green. When the color bit changed, the pixel then satisfied all the requirements of being orange, namely being in an odd column with the color bit set to one.

Let's pull back from the digression on bytes and color and apply it to character graphics. The first revelation can be found by remembering that character shapes in GraForth are seven pixels wide by eight pixels tall. The character width of seven pixels happens to coincide with the seven pixels per byte (with one color bit) discussed above. Therefore, each hi-res character is one byte wide and eight bytes tall, and each row of the character has its own color bit.

Suppose that you've created and printed at the left a single character shape that is a solid green block. This means that the color bits for the character are zero and all of the odd columns (1, 3, 5) contain pixels. Suppose that the same shape is then printed one character space to the right. That means it will be offset by seven pixels from the original. The odd pixels (1, 3, 5) will now fall into even ($1+7=8$, $3+7=10$, $5+7=12$) columns, changing the green block to violet! Second revelation: colored character blocks change color when moved between even and odd columns. Actually, character blocks do not have a true color while in the character set. Color and color changes only become apparent when the blocks are plotted on the Apple screen.

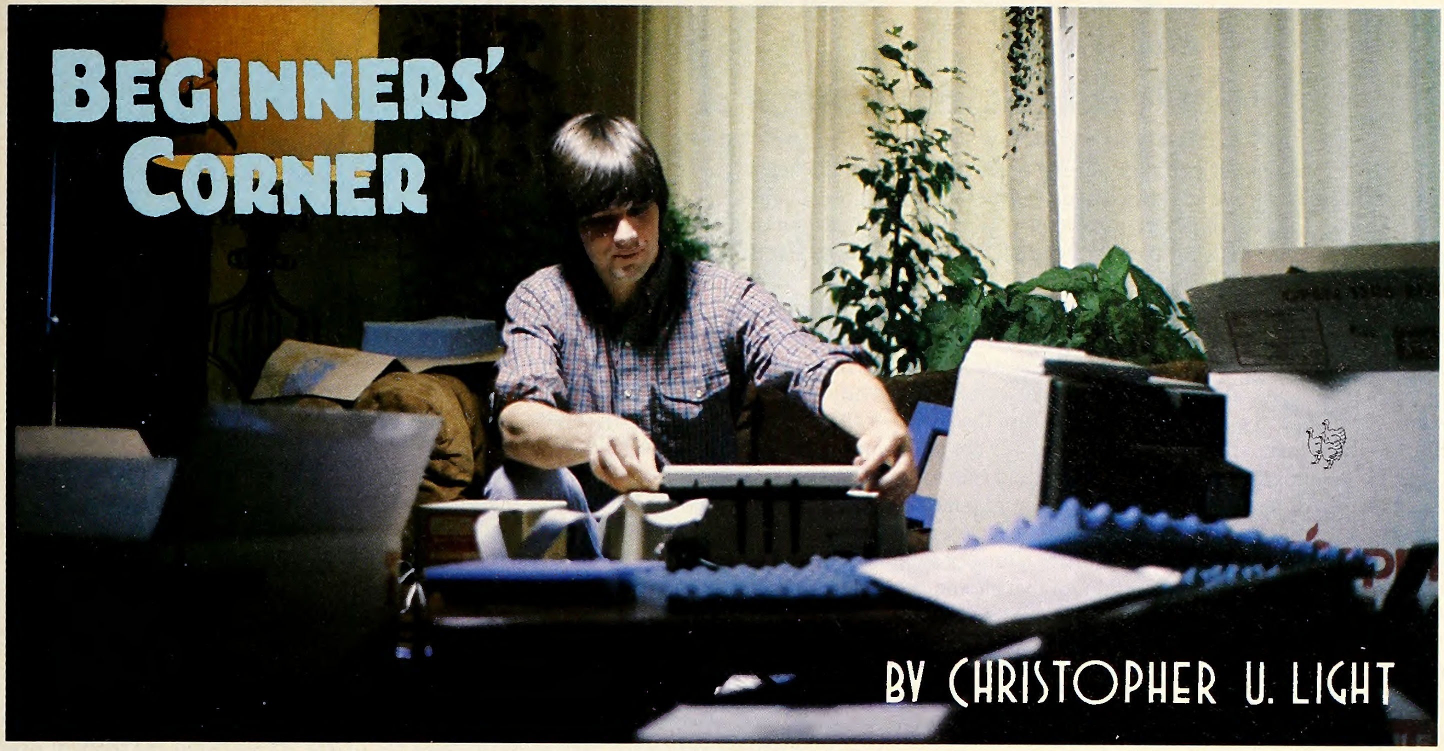
This brings us to the realization that if we want to keep the colors constant while moving horizontally across the screen, two sets of character shapes will be needed. They will be identical, except that one will be offset by one pixel. One set will be used in even columns, the other in odd. As the blocks are moved one character space, or seven pixels, at a time, the one pixel offset in every other block will actually make the distance either six pixels or eight pixels. Even columns will stay even, and odd will stay odd.

The pixel-shifter routine we used earlier can easily be used to create the other color images. Two things should be kept in mind: First, since the character editor edits the images starting on an odd character column (*1 htab*), then the colors will coincide with the editor when the block is printed on any odd column. Printing a colored character block on an even column will show colors reversed from those in the character editor. Second, when we created a shifted character image above, we needed a wider block to save it. The same is true for shifts of one pixel. The actual image must be at least one pixel narrower than the block it is saved in, to leave room for the one-pixel shift.

Designing character sets and animating character blocks is, of course, a very new art. Questions of what looks best and how best to do something are always subjective and depend on the particular application being written. We've tried to give you the tools you need for working confidently with GraForth's character graphics, and some ideas on smooth animation.

Next month, we'll turn to the three-dimensional capabilities of GraForth, showing how to create, save, and manipulate 3-D shapes. See you then.


Nothing like it before. Nothing else like it now
. . . brings you continuous Hi-Res action-animation in every adven turous moment! And, real running, leaping, crawling. Real fighting shooting, stabbing, dynamiting. Real wounding, poisoning killing. Real action, excitement, mystery! All in a real-tim challenging adventure that's the wave of the future
Paul Stevenson's graphic genius, first displayed in hi best selling "Swashbuckler" sword fighting game outdoes itself in AZTEC. You're inside an ancien Aztec pyramid searching for the golden ido Descend deep into the heart of the temple— meet cobras, scorpions, giant lizards hostile Aztec guardians and more. Watc for hidden trapdoors and strange death rooms. Be ready to fight, or run, crawl o jump to possible safety. The menace i real, the options and strategy are yours You've never seen an adventure like Aztec! You'll never tire of its amazin action-animation and exciting challenge $39.95 for the Apple II* At you computer store or
DATAMOST INC
9748 Cozycroft Ave., Chatsworth, Ca 91311. (213) 709-1202
AZTEC
VISA/MASTERCARD accepted. $2.00 shipping handling charge. (California residents add 6½% sales tax.)
*Apple II is a trademark of Apple Computer, Inc.




Last time we looked at the traps that are built into Applesoft and DOS to prevent a program from, for example, launching a space shuttle without loading the shuttle's landing instructions. We'll look now at error traps you can create and include in your own programs so that users who hit keys by mistake won't have to run your programs again or, worse yet, be required to reboot them. Error traps, at least those that are intended to trap simple errors of carelessness, are the mark of a pro. They're also just common courtesy toward your users.

Error traps make sense in any program. Even if you expect to be the only user of your own programs, you'll be better off if you've at least devised ways of catching and correcting keyboard errors. This is twice as important, of course, if your output will be going to a chairman of the board who has no idea what *break in nn* means.

**To Trap or Not To Trap?** The two extremes in error trapping are neglect and bludgeon. The first approach depends solely on Applesoft's error messages and lets any user with clumsy fingers be damned. The second uses Applesoft's error message cancellation routine for everything and begins every program with an "onerr goto" command. Neither approach is likely to be appreciated by users of your programs.

The best approach to error trapping is to figure out what errors your users are likely to make and devise appropriate methods of handling these errors if they should occur. Good error trapping should involve the minimum disruption in the running of a program. If a print-out is part of the program, you shouldn't allow your error trapping to mess it up.

For certain potential errors, however, neglect may be the preferred solution. Suppose, for example, that your program attempts to load another program from disk. The most likely error a user could make that would affect this part of the program would be to insert the wrong disk or leave the wrong disk in the drive. Apple's DOS message "file not found" conveys the information very well in these instances. If there's no disk at all in the drive, the message "I/O error" is a bit more cryptic, but users soon realize that this always indicates a problem in reading from or writing to a disk. You may feel that anybody who leaves a drive door open or inserts the wrong disk not only deserves Apple's DOS messages but also deserves having to run your program again from the beginning.

**Disk Drive Dropouts.** On the other hand, you might feel that it's only common courtesy to have the program stop itself at that point and print a message such as "Insert correct disk. When ready, type return to continue," and then go on. Let's do it. First create a small program so you have something to load. Type:

```
NEW
10 HOME
20 PRINT "OH, WHAT A BEAUTIFUL MORNING."
```

Save the file under the name Morning. Now write your main program.

```
NEW
10 HOME
20 INPUT "IS IT MORNING OR AFTERNOON? "; A$
30 IF A$ = "MORNING" THEN GOTO 50
40 GOTO 60
50 PRINT CHR$(4); "RUN MORNING"
60 PRINT "IT'S A LOVELY AFTERNOON"
70 END
```

Save this one under the name Example, then run it to make sure it's entered correctly. You will have to load Example again if you answered "morning."

By the way, this program accomplishes in one line something we used two instructions for last month. In last month's program, we put a DOS instruction inside an Applesoft program with the commands *D$ = chr$(4)* and *print D$; ". . ."*, which is more efficient than our new way, if you need to refer to DOS several times. However, both work. It's worth noting in passing that *print " "; ". . ."* also will work if you type control-D between the quotes when you enter the line. Unfortunately, you won't be able to read the control-D and may forget what was there later.

You'll notice that this program has no error traps of any kind. If users type "night," or "yellow," or "hello," they'll still get the afternoon response. To correct that, add

```
35 IF A$ <> "AFTERNOON" THEN GOTO 99
99 PRINT "YOU GOOFED. START OVER": GOTO 20
```

The addition of these two lines will take care of any incorrect response from the keyboard except a reset. Be sure to save again after making any changes.

Suppose a user has inserted the wrong disk or none at all. To see what will happen, open the door to your drive after typing "morning" but before hitting return. You should get the message "I/O error break in 60." Although you should avoid the indiscriminate use of onerr, in this case it's the only thing that can save you from an I/O error without requiring that you rerun the entire program. So add the lines:

```
50 ONERR GOTO 95
55 PRINT CHR$(4); "RUN MORNING"
65 POKE 216,0
95 INPUT "INSERT THE CORRECT DISK, THEN HIT RETURN.";B$:
   GOTO 55
```

Note that instead of putting a general onerr command at the beginning of the program, we've inserted it just before the line that could cause a DOS error. If the error occurs, a prompting message is printed, and the program stops and waits for the user to signal (by hitting return) that he has inserted the correct disk.

The only reason for asking the user to signal when ready is to stop the program and provide time to insert the correct disk. Actually, hitting any sequence of keys followed by return will work. Since the variable B$ is never used again, it doesn't matter what value it has, and it doesn't seem worthwhile to put in a test to make sure that return is the only key hit. Return was chosen, of course, because it's the minimum necessary.

Then the program goes back to line 60 and tries to load from the disk again. If there is a successful load, the program goes to line 65, although the user is unaware of this. The command in line 65, *poke 216,0*, turns off the onerr command so that later errors in a long program won't also go back to line 60. Onerr is a powerful tool that should be used for specific errors and turned off when the possibility of those errors occurring is past.

**Coded Messages.** Applesoft's error handling can also distinguish between several possible errors because, when an error is encountered, the code number of the error is stored in location 222. Applesoft's syntax error, for example, is number 16, while DOS's is 11. Thus, if you want to have separate error messages for different errors, you can. Let's make the following changes in our program.

```
50 ONERR GOTO 80
80 Z = PEEK (222)
85 IF Z = 6 THEN GOTO 95
87 IF Z <> 8 THEN GOTO 99
90 INPUT "CLOSE THE DOOR, DUMMY. THEN HIT
   RETURN";B$:GOTO 55
```

Error message 6 is "file not found," while message number 8 is "I/O error." The result is the same—another attempt to load Morning—but the messages are specific to the error. Try running the program now with the wrong disk, the door open, and so on.

Now, replace the goto 55 in lines 90 and 95 with the command *resume*. When the error flag, set by onerr goto, is on and an error occurs, resume returns the program to the line in which the error occurred and tries again on the assumption that the user has corrected the error. Try running the program now with the wrong disk or the door open.

Although resume is handy, it's not really recommended for general use. For one thing, you must make certain that your program branches around it if there's no error. If a running program reaches resume without encountering an error first, unpredictable things can happen, the most likely of which is a hung system. If there's an error in the error handling routine with resume, the program will go into an infinite loop.

Another reason for caution in the use of resume is that it's lazy programming. It lets you get by without anticipating the errors a user might make and deciding for yourself what you want the program to do when it encounters an error. The use of goto, on the other hand, directs the program exactly where you want it, and you remain in full control of it.

**Variable Resistance.** The onerr goto command cancels Apple's built-in error messages, allowing you to substitute your own. It also lets you continue running from any point in the program you wish (presumably the point where the error occurred), but it does not cancel the error itself. Thus, as with any error message, the values of all variables are retained, but certain pointers and stacks are reset to zero. These include the internal counter used in for-next loops and the pointer that tells a subroutine which gosub line to go back to when it finds return. Therefore, if you have an error in a loop or subroutine, you must restart the loop or the subroutine. In other words, your error handling routine must go to the line that contains the for or gosub command, not to the one containing next or return.

It's worthwhile to take a look at what happens to an error in a loop. Type *new* and then enter:

```
10 HOME
20 FOR I = 1 TO 3
30 INPUT "TYPE A NUMBER "; X
40 PRINT X
50 NEXT I
60 END
```

As usual, run the program to make sure it's okay. Notice that the variable X in line 30 is numeric and not a string; it does not have a dollar sign and therefore cannot represent words or letters, only numbers. For the inputs, first enter 1, then 2, and 3.

The next time around, enter 1, then Q or some other letter. You'll get the built-in input error handling response, *?reenter*. If you enter a number, say 2, the program will print this and then look for the third number.

Suppose, for some reason, that you wanted to have the program simply repeat "type a number" rather than ?reenter and wanted to use the onerr command to direct the program. There are three alternatives, none of which works satisfactorily. The first format is to go back to the for statement; the second is to return to somewhere within the loop, and the third is to go to the next I. Try all three with the line:

```
25 ONERR GOTO 20
```

and, after running the program, try making it goto 30 and then goto 50. When you go back to the for in line 20, you start the loop all over again and have to reenter all data entered prior to the error. When you jump to next in line 50, the system hangs. When you go to your input line inside the loop, you end up with an infinite loop and can enter numbers forever without exiting it.

**Indexing the Loop.** There is a way to have only three inputs before the end of the program and yet avoid the ?reenter message. That is to keep your own counter for the loop and to reset to the value of that counter after the error. It's really quite easy. Just make the following changes:

```
15 N = 1
20 FOR I = N TO 3
23 N = I
25 ONERR GOTO 20
55 POKE 216,0
```

Your counter in this case is the variable N. Run the program and you'll see that it will continue asking for input if you type a letter by mistake. It will end after the third number it receives even if you enter several letters in error first.

Because Applesoft has ?reenter as an error handling routine already in ROM that does essentially the same thing as this method, you may never want to restart a loop in the middle after an input error. Nevertheless, the technique may be useful for covering other error situations in which it would be disadvantageous to start the loop all over again. The basic trick is to set up your own counter, use that counter to initialize the loop, and, after an error, restart the loop at the value of your counter. From Applesoft's point of view, you are beginning the loop at the beginning again (that is, with for), so there's nothing wrong. As before, we've added the poke command to turn off the error flag when the particular error trapping routine is over, even though this precaution is unnecessary in this short program.

These are a few suggestions. Undoubtedly you'll come up with many more error traps. The specific techniques are not very important; in any program, there will usually be half a dozen methods of doing a job. What is important is to develop the habit of anticipating ways that users can mess up your program while it's running and then to force them to correct their errors in a manner that allows the program to continue where it left off. After all, if the only thing you want users of your programs to do after they make a mistake is to run them over again, you don't need any error traps. Applesoft will quit and print its own error messages, or it'll get hung up, or something. But it won't make you any friends and it'll probably create a few enemies.

Next month: free software!

# A TEACHER FOR THE APPLE

## Learn *How to Operate the Apple II Plus* with the same self-study course Apple has chosen to train its own field sales reps!

Imagine having a personal tutor "talk you through" the practical operations of an Apple II Plus computer. In just 3 "hands on" training sessions of one or two hours each, you'll learn how to:

- Use each special key and essential command
- Load and run programs
- Save programs and data
- Copy programs and diskettes
- Protect programs
- Modify programs
- Use your computer as a super calculator
- And much more!

**No Programming Required.** You don't need to be a programmer to operate a computer, any more than you need to know how to design a car to drive one.

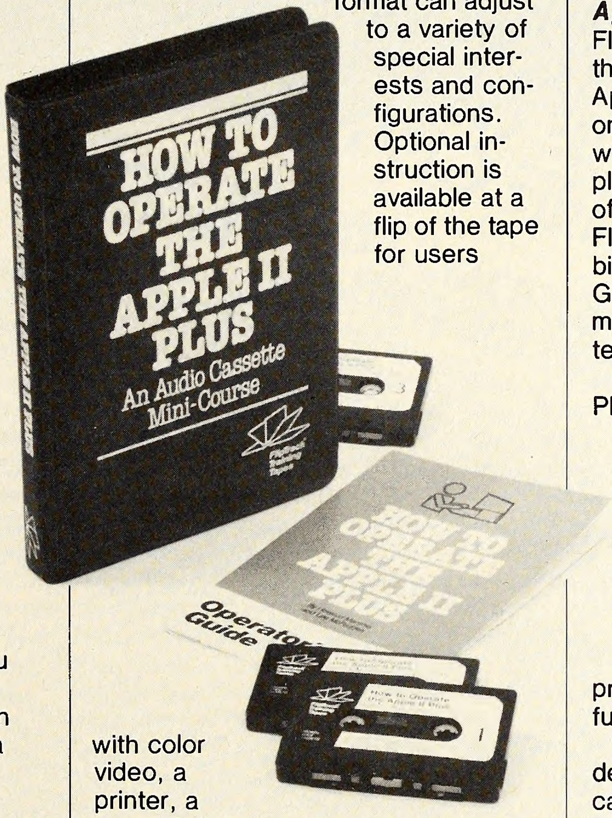
Our purpose is to put you in the driver's seat of your Apple computer so that you can confidently use existing programs . . . enter your own data . . . organize your files . . . and protect them. If you later decide to take up programming, you'll have the necessary background for doing so.

**Teach Yourself. Train Others.** *How to Operate the Apple II Plus* is designed for the first-time computer user. No technical knowledge is assumed. Perfect for training others: Staff . . . colleagues . . . clients . . . customers . . . students . . . family. Use it again and again.

**Why Audio Instruction?** Simple. It's human. It's interactive. It's multi-sensory. It's patient. It's inexpensive. And it works!

FlipTrack® Training Tapes deliver carefully sequenced instruction in a relaxed, conversational manner. You concentrate on the screen and keyboard, not a book. You learn at your own pace, trying every command and observing the results.

**The FlipTrack® Difference.** Best of all, our patented FlipTrack® cassette format can adjust to a variety of special interests and configurations. Optional instruction is available at a flip of the tape for users with color video, a printer, a second disk drive, or game controllers.




You save time and avoid confusion by learning just the procedures that relate to your system, or listening to just the topics that interest you. With FlipTrack® Training Tapes, you choose whether to follow a "full" track or a "fast" track every step of the way.

**Examine *How to Operate the Apple II Plus* FREE for 15 days.** Let FlipTrack® Training Tapes speak for themselves by dropping by your local Apple dealer for an on-the-spot demonstration. Or if that's not convenient, we'll be happy to send you the complete mini-course on a 15-day "right of return" basis. You'll receive all 3 FlipTrack® cassettes in a handy vinyl binder, along with an Operator's Guide, which will serve as your permanent reference to the procedures, terms and concepts of the course.

Your only prerequisite is an Apple II Plus computer with the System Master 3.3 diskette and a standard audio cassette player. No computer hookup is necessary for the cassette player.

Try a lesson or two yourself, or share them with others. If you are not delighted at how quickly and easily you master the operation of your Apple, simply return the program in its original condition for a full refund. No questions asked.

**To order . . .** See your local Apple dealer. Return the coupon below. Or call us toll free. Here's our 24-hour toll free number for credit card orders only:

**800/227-1617, Ext. 439**

In California, call 800/772-3545, Ext. 439.
You have nothing to lose, so act today.

---

**YES!** Please rush me *How to Operate the Apple II Plus*. I understand that if I'm not delighted, I may return the mini-course undamaged within 15 days for a full refund. No questions asked.

☐ My check is enclosed for $52.45 ($49.95 plus $2.50 shipping & handling) Illinois customers include sales tax.
☐ Charge my credit card:
☐ VISA ☐ MasterCard Interbank No.________
Card No.________________ Exp. Date________

Name________________________

Address________________________

City________________ State________ Zip________

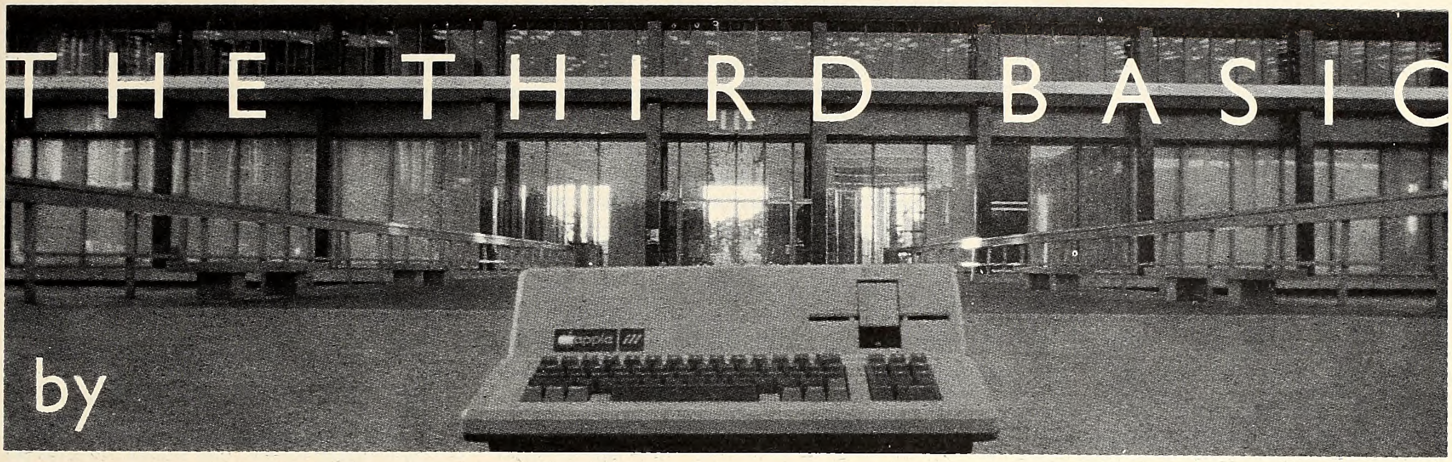
Signature________________________

Outside USA add $10 per unit. Airmail extra.

**FlipTrack® Training Tapes**
Division of Mosaic Media, Inc.
Dept ST-6, P.O. Box 711,
Glen Ellyn, IL 60137
312/790-1117

Apple and Apple II Plus are trademarks of Apple Computer, Inc.

# THE THIRD BASIC



by

Taylor Pohlman

## Exploring Business Basic, Part 14

Welcome back to Basicland. Before we plunge waist-deep into today's exciting episode, it's time to congratulate Arnold Bailey of New York State for his intrepid solution to our challenge on two-byte reads from the console. For those who missed the last two cliffhangers, we wrapped up the discussion of nifty *.console* features by challenging you, the studio audience, to come up with a routine in Business Basic that allowed a program to tell the difference between the enter and return keys. Last month a solution was furnished in this column, but because of publishing lead times, it had to be submitted before the challenge itself could be issued (maybe a better name for the column is Tomorrowland!).

The published solution relied heavily on the desire to read one keystroke at a time and sample each one, simulating the get statement in Basic. Arnold took advantage of the fact that the console still terminates a read on an ASCII 13 (which return and enter both generate) and correctly identified that an input# statement was required to correctly read both bytes. It was an excellent solution, and well documented (embarrassingly well documented compared to the abbreviated listings we normally foist upon you in this column). For being first with a valid solution, Arnold wins a copy of *Quickfile III* and the thanks of a grateful nation.

**Sorting It All Out.** One of the key ingredients of most business and scientific programs that handle large amounts of data is the ability to arrange that data in an ordered sequence. "Arrange data in an ordered sequence" is, of course, a windy way of saying "sorting." There are as many sort techniques as there are people to think them up, but for the purposes of this column and its Christmas (December) cousin, we will stick to four or five fundamental methods. Business Basic has several nifty features that make sorting more efficient, and the huge memory space makes it practical to sort large collections of strings or numbers in memory. For that reason, merging, the flip side of most sort techniques, will be covered only briefly in the December issue. For now we'll stick to in-memory sorts to illustrate the techniques.

**We're Forever Showing Bubbles.** The most common sorting technique, and the one guaranteed to show up in every elementary textbook, is the *bubble* sort. So named because of the technique of taking a value and shuffling (bubbling) it up to its proper place in the list, it depends on comparing each value to the one next door and exchanging them if the order is wrong. If you compare each element with its neighbor enough times, eventually the list will be sorted. For the purposes of the sample program, and all the rest of the programs in this series, we will assume the desire is to sort the lists in ascending (lowest to highest) order. Let's plunge into the first example to illustrate how this works:

```
10    REM sort using bubble technique
20    REM n is number of elements
30    REM sarray is the array to be sorted
50    REM
100   DIM sarray (1000)
110   INPUT"Sort Routine. Number of
      elements to generate: ";a$
120   n=CONV(a$):IF n<2 THEN 200
130   FOR i=1 TO n:sarray(i)=RND(1):NEXT
135   PRINT"Start of Sort"
140   GOSUB 1000
150   PRINT"Sort complete. First 10
      elements are:"
160   FOR i=0 TO 10:PRINT
      sarray(i);" ";:NEXT
170   PRINT:PRINT:GOTO 110
200   END
1000  top=n-1
1010  madeswap=0
1020  FOR i=1 TO top
1030  IF sarray(i)>sarray(i+1) THEN SWAP
      sarray(i),sarray(i+1):madeswap=1
1040  NEXT i
1050  IF madeswap THEN
      top=top-1:GOTO 1010
1071  RETURN
```

This first program sorts a numeric array, rearranging it in memory. For purposes of testing it, lines 110 through 140 ask for the number of elements to sort, load sarray with random values, and then call the subroutine at line 1000 to perform the actual sort. After returning from the sort, lines 150 through 200 print out the first ten elements (just to demonstrate that the data is sorted) and end. Remember that this framework is deliberately simple, so we can concentrate on the sort technique itself.

Line 1000 establishes the upper limit of our check for correct order and line 1010 establishes a variable madeswap, which is a flag we'll use later to determine if more sorting needs to be done. That leads us to the main routine in lines 1020 through 1040. Line 1030 scans each element and compares it to the next higher element. If the first element is greater than the second, then the swap statement is used to exchange them, and the madeswap flag is set to show that an exchange was made. Then the process repeats, with that next higher element compared with its next highest companion, until the top is reached. That's why top is set equal to the number of elements minus one.

When a complete scan is made, line 1050 checks to see if any swaps were made in the last pass. If there were none, then the array must be in order. If not, the array might not be in order, and it needs to be processed again until all possible swaps have been made. Notice that line 1050 resets top, the limit of checking, since after each pass the smallest number found anywhere in the array will be forced to the top of the list (try it out if you don't believe it). This is something that some versions of this routine miss, and it leads to a lot of unnecessary scanning. Notice also that we take advantage of the swap statement. Some Basics don't have this statement, and the result is that you have to assign one value to a temporary variable and do the reassignments. This is a nuisance and also takes a lot of time, whereas swap is very fast.

As long as things are being noted, it's probably worth pointing out that this routine could be rewritten for descending order by rearranging the sense of the if statement and having the search go from a varying bottom to a fixed top instead of the other way around. One last comment: Although this sort technique is the simplest, it is also the slowest, except for those situations where the list is very short or almost sorted already. Simple timing tests will convince you that the routine slows down nonlinearly as the size increases. "Nonlinearly" means it goes from bad to awful without passing through worse.

**Getting the Point.** The routine below is another variation on the bubble theme, with one important exception. Sometimes it doesn't make sense to actually rearrange the data, but rather only to create a list that describes what the order would be if the data were physically sorted. For example, consider the following list (as it might have been read from a file):

| Record number | Item |
|---|---|
| 1 | Henry |
| 2 | Bill |
| 3 | Gloria |
| 4 | Alphonse |
| 5 | Gaston |

One way to sort this list is to simply arrange it in the following sequence:

Alphonse, Bill, Gaston, Gloria, Henry

That's ascending alphabetical order. However, we could represent that same sequence by listing the record numbers:

4 2 5 3 1

Not only is this second representation more compact, it may actually represent less work to rearrange the record numbers than the actual data itself. Such numbers are called *pointers*, since the value that is listed is just a pointer to the actual data location, not the data itself. If you want to construct the sorted list of names, it is easy to look them up using the record number (pointer) list. Some languages and systems make a great deal out of this pointer concept. For now, we suggest that you use the technique wherever it makes sense from a performance, storage, and convenience standpoint. The following example is an adaptation of the first program to incorporate this pointer sort technique:

```
10   REM sort using bubble technique
20   REM N is number of elements
30   REM sarray is the array to be sorted
40   REM parray contains the pointers to
     the sorted array
50   REM
100  DIM sarray(1000),parray%(1000)
110  INPUT"Sort Routine. Number of
     elements to generate: ";a$
120  n=CONV(a$):IF n<2 THEN 200
130  FOR i=1 TO n:sarray(i)=RND(1)
     :parray%(i)=i:NEXT
135  PRINT"Start of Sort"
140  GOSUB 1000
150  PRINT"Sort complete. First 10
     elements are:"
160  FOR i=1 TO 10:PRINT
     sarray(parray%(i));" ";:NEXT
170  PRINT:PRINT:GOTO 110
200  END
1000  top=n-1
1010  madeswap=0
1020  FOR i=1 TO top
1030  IF sarray(parray%(i))>
      sarray(parray%(i+1)) THEN SWAP
      parray%(i),parray%(i+1):madeswap
      =1
1040  NEXT i
1050  IF madeswap THEN
      top=top-1:GOTO 1010
1060  RETURN
```

Notice that we have introduced a new array in the program at line 100. Parray% contains the pointers to the actual locations in sarray that represent the sorted order. Notice in line 130 that parray% is set initially to the sequence 1, 2, 3 . . . , the same sequence as we find the data in sarray initially. However, since the pointer array contains only references to locations in the original array, it can be declared an integer array (maximum value 32,767) to save space. Having set up all the values, the gosub to line 1000 sorts the data, as in the first example; except that now, in line 1030, instead of testing the actual sarray values directly, we use parray%(i) and parray%(i+1) as pointers to where the real data is. Once the comparisons are made, the pointers (not the values themselves) are physically exchanged. When the sort is finished, the routine returns to line 150 to print out the sorted data. The print statement in line 160 illustrates how the pointer array is used to look up the correct sequence of values, even though they are actually scattered around within sarray.

One of the interesting possibilities of this technique is that it is possible to have more than one pointer array to a given data array. In that way, you could have an sarray that had associated with it a parrayup% and a parraydown% pointer array, so that listings and searches could be done in either order, depending on the program requirements, without re-sorting. These are sometimes called indexes, and are very useful in many applications. More typically, pointer arrays are used in situations where the original array consists of string values, or fields within disk records. In that case, the economy of storage of an integer array is very valuable, and the pointers consist of actual disk record numbers, which are then easy to look up in any specified order. Next time we will consider some sort techniques and pointer arrays that are particularly suited to disk look-ups. These techniques are sometimes generalized under the heading of access methods.

**A Mild Speed Lift.** All this is fine, but the original warning is still worth considering. Bubble sorts are simple and easy to understand, but they are painfully slow on any reasonable amount of data. The fundamental problem is that the algorithm requires lots of comparisons, and even if the comparison indicates that the data needs to be moved, a move of one cell at a time is all that is possible. That means that for a small value to get from the top to the bottom on an ascending sort requires lots of exchanges (one for each value in the array). One way to speed up this process is to make comparisons across larger distances and thereby cut down the number of compares and exchanges necessary. A sorting algorithm called the Shell-Metzner sort accomplishes this. Usually called the Shell sort (which is appropriate considering its similarity to the shell game switching technique), it depends on long-distance comparisons and swaps to speed up the sorting process. For simplicity, we'll look at the Shell sort in standard form, without the additions for pointer sorting. A typical Shell sort routine looks like this:

```
10   REM sort using Shell-Metzner technique
20   REM n is number of elements
30   REM sarray is the array to be sorted
50   REM
100  DIM sarrray(1000)
110  INPUT"Sort Routine. Number
     of elements to generate: ";a$
120  n=CONV(a$):IF n<2 THEN 200
130  FOR i=1 TO n:sarray(i)=RND(1):NEXT
135  PRINT"Start of Sort"
140  GOSUB 1000
150  PRINT"Sort complete. First 10
     elements are:"
160  FOR i=1 TO 10:PRINT
     sarray(i)" ";:NEXT
170  PRINT:PRINT:GOTO 110
200  END
1000 IF n<2 THEN RETURN
1010 span=INT(n/2)
1030 newspan=n-span
1040 FOR i=1 TO newspan
1050 temp=i
1060 upper=temp+span
1070 IF sarray(temp)>sarray(upper) THEN
     SWAP sarray(temp),sarray(upper):
     temp=temp-span:IF temp > 0 THEN
     1060
1080 NEXT i
1090 span=INT(span/2)
1100 IF span THEN GOTO 1030:ELSE
     RETURN
```

As you can see, the initial routine in lines 10 through 200 is essentially the same as in the other routines. The subroutine in line 1000 implements the Shell algorithm, starting with a quick check in line 1000 that there is more than one element to be sorted. Once that is established, line 1010 divides the array into halves, and line 1030 establishes the center point around which swaps will be made. It would be a good idea to create a small array on paper and follow through on exactly how this routine works for your own satisfaction. Briefly, the newspan variable establishes a pivot point, with the loop in lines 1040 through 1080 facilitating comparisons and swaps between the upper and lower portions of this pivot point. After each swap, the range of search is narrowed by decreasing the temp value in line 1070, and the process is repeated. After each major pass with a span value, line 1090 cuts the span value in half and repeats the process, until the span is one, at which point the array is sorted. Line 1100 checks for that happy occurrence, and returns if it has come to pass.

Several things are worthy of note in this routine. First, the Shell sort will work faster on sorted data than unsorted data, and in nearly every unsorted case, it will outperform the bubble sort, dramatically so in cases above fifty to one hundred values. Also, this routine can easily be adapted to a pointer sort, using the techniques outlined in the second example above. String arrays can be sorted simply by replacing the numeric comparison in line 1070 with a string array compare. The swap statement works equally well with string and numeric data.

**Sort of a New Way To Sort.** The preceding algorithms are fairly standard and safe and will reliably sort any kind of data. Sometimes in application programs we are more fortunate and can have special knowledge of what kind of data we are sorting. This allows for special techniques which are faster than any general-purpose routine could be. Although the Apple III is one of the fastest personal computers around, it's not exactly a Cray I, so it often pays to be able to use special tricks. Imagine a situation where there are lots of records to sort, but there are only a few unique values among all the records. One classic example is sorting mailing address records on the basis of the value of the state field. Obviously, there may be thousands of records to sort, but there are only fifty possible states, each typically represented as a two-character code. There are many other examples, but that is one that is easy to imagine.

The following program generates random string values and then lets you test the practicality of a sort technique based on a concept called an *inverted list*. Inverted lists are a favorite topic among access method and database experts, but the same principles can apply to sorts. Basically, an inverted list is not an upside-down version of a regular list, as you might expect from the title. Rather, you can think of it as a list of all the unique values in another list, with sublists that contain the record numbers of all the records sharing the same field value. In our example, if we had the following situation:

| Record number | State |
|---|---|
| 1 | CA |
| 2 | MD |
| 3 | CA |
| 4 | NY |
| 5 | MO |
| 6 | CA |
| 7 | MD |

Then the inverted list of this data would look like this:

| Value | Locations |
|---|---|
| CA | 1, 3, 6 |
| MD | 2, 7 |
| MO | 5 |
| NY | 4 |

The two representations contain the same information, but in a considerably different form. Note also that the inverted list is assembled in ascending alphabetical order. That's not necessary to the example, but once the unique values are established, it is generally easy to sort them. This is especially true if the number of unique values is much smaller than the total number of values. The following program lets you generate random string arrays, with up to a thousand values, and then pick a group of columns on which to sort. Try it initially by sorting only on the first column. This will guarantee that there are only twenty-six unique values (since the routine generates only upper-case letters), no matter how many strings you generate. First, the main routine:



```
10   DIM pntr%(1000),startval%(255),
     endval%(255),spointer%(255),
     sarray%(1000)
20   zero$=CHR$(0):zero%=0
30   DIM array$(1000),value$(128)
40   INPUT"Number of strings to
     generate: ";a
50   INPUT"Number of characters per
     string: ";b
60   FOR i=1 TO a
70   FOR j=1 TO b
80   array$(i)=array$(i)+
     CHR$(INT(RND(1)*26)+65)
90   NEXT j,i
100  FOR i=1 TO a:PRINT array$(i);" ";:
     NEXT i
130  PRINT:PRINT:INPUT"Start and end
     columns for sort: ";c1,c2
132  FOR i=1 TO a:pntr%(i)=zero%:NEXT:
     FOR i=1 TO 255:startval%(i)=zero%:
     endval1%(i)=zero%:NEXT
133  sortval$=""
140  FOR rec%=1 TO a
150  item$=MID$(array$(rec%),c1,c2-
     c1+1)
160  GOSUB 2000
170  NEXT rec%
190  GOSUB 3000
196  PRINT"Number of unique values: ";n:
     PRINT"Ratio of total records to unique
     values: ";a/n
197  INPUT"Press return for the sorted
     list ";a$
200  FOR i=1 TO a:PRINT
     array$(sarray%(i));" ";:NEXT i
210  PRINT:PRINT:INPUT"Sort the array
     again? ";a$
220  a$=MID$(a$,1,1):IF a$="Y" OR
     a$="y" THEN 100
230  END
```

Lines 10 and 20 set up a lot of values that will be used later, while line 30 sets up the main string array, array$, and the array that will contain unique values, value$. After prompting for the number of strings and the size of each string, lines 60 through 90 build the string array by randomly creating character strings composed of upper-case ASCII characters. Line 100 prints out the created array, and line 130 requests the columns on which to sort. Unless you create very few strings, it is best to sort on only one column, since the unique combinations possible in sets of more than one column are probably too great for the routine to work properly. Note that in a controlled (nonrandom) set, like the states, this might not be a problem.

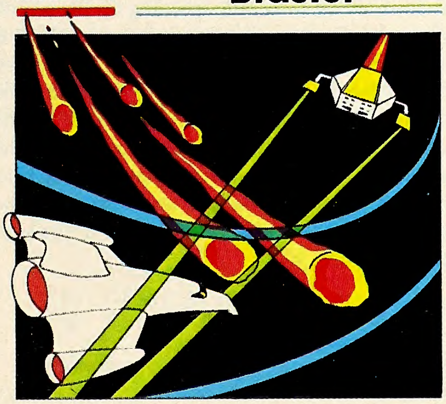
In any case, once the columns are chosen, lines 132 and 133 initialize values and prepare for the main sort loop in lines 140 through 170. Note that for each record (rec%) to be sorted, the variable item$ contains the value extracted from the main record that will become the sort key for that record. The routine at line 2000, which we will examine shortly, adds the current value to the list of unique elements if necessary and inserts its record number in the general list. The subroutine at line 3000 then orders the unique value list, creates the sorted pointer list in sarray%, and returns to lines 196 through 220 to print the list on demand and start the process over, if desired.

Let's look now at the routine that creates and adds to the inverted list:

# Great Games From Piccadilly Software Inc.

Piccadilly Software Presents

**Star Blaster**



Piccadilly Software Presents

**Falcons**

**COMING SOON!**

New titles to be released shortly!

• • • •

INVASION OF EVERYTHING

• • • •

MARTIAN SOIL

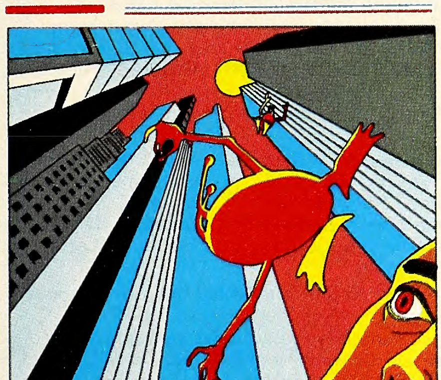
Piccadilly Software Presents

**Survival**

COMING IN THE 2ND HALF OF 1982

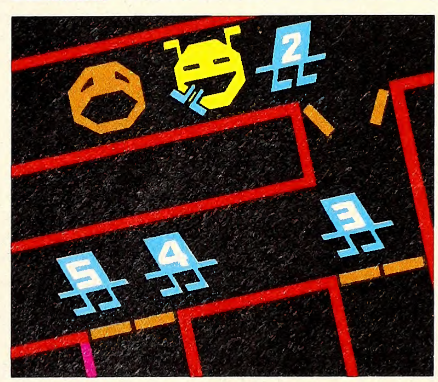
Piccadilly Software Presents

**Suicide!**



Piccadilly Software Presents

**Succession**




## Available at your local dealer.

Requires: APPLE II/APPLE II PLUS with 48K in Dos 3.3 or 3.2 • APPLE III

**Piccadilly Software Inc. • 89 Summit Avenue • Summit, N.J. 07901 • (201) 277-1020**

**If you're unable to obtain Piccadilly Games from your local dealer, send $29.95 per game plus $1.50 per order for postage and handling. NJ residents add 5% sales tax. Outside US, send $3.00 per order for shipping.**

*Apple is a registered trademark of Apple Computer Inc.

```
2000  x=INSTR(sortval$,item$)
2010  IF x THEN pntr%(endval%(x))
        =rec%:endval%(x)=rec%:RETURN
2020  x=LEN(sortval$)+2
2025  IF x+LEN(item$)>255 THEN PRINT
        "sortval$ overflow. sort aborted":STOP
2030  sortval$=sortval$+zero$+item$
2040  startval%(x)=rec%
2050  endval%(x)=rec%
2060  RETURN
```

This routine makes good use of the INSTR function to search a string called sortval$. Sortval$ contains all the unique values, separated by the ASCII value 0. This means that the unique values can easily be identified, assuming that none of the values contain an ASCII 0 themselves. Once INSTR either finds or doesn't find the item$ value in sortval$ the rest of the routine is set into motion. In the case that item$ already exists in sortval$, line 2010 updates the pntr% array by putting the current record number into the space reserved for the last record number in the list of that unique value. That is, endval% is an array that remembers the index in pntr% of the last occurrence of any particular unique value. That means that the next occurrence of that value gets automatically put into the location in endval% that matches the start location of the value in sortval$. This is another good one to try on paper until you get a feel for how it works. Assuming that the value was found, the routine's work is finished for now, and it returns to look at the next value.

If the item was not found in sortval$, that means it is a new, unique value. Line 2020 gets the next possible location for storing the new value, and line 2025 checks to see if there is room for the value. You could probably come up with something more friendly than the stop statement to solve the problem. In any case, if there is room, line 2030 adds the value to the sortval$ string, with the zero$ spacer, and lines 2040 and 2050 establish start and end locations for this new value and then return to get the next record.

This process continues until all the records are examined and all unique values are added to sortval$ and their beginning and ending pointers are established in the appropriate arrays. At this point pntr% contains a linked list for each unique value of sortval$, with the starting point of the list pointed to by the appropriate element of startval% and the end point defined by a zero in the location pointed to by endval%. Now the fun begins. Having assembled the list of pointers to all values, it is necessary to sort the unique values themselves into the appropriate order and then assemble the individual linked lists into a total sorted list. The routine to break out the unique values and sort them looks like this:

```
3000  sortval$=MID$(sortval$,2)
3002  FOR n=1 TO 255
3005  x=INSTR(sortval$,zero$)
3010  IF x=0 THEN 3050
3015  value$(n)=LEFT$(sortval$,x-1)
3020  sortval$=MID$(sortval$,x+1)
3025  NEXT n
3050  value$(n)=sortval$
3055  last=1
3060  FOR i=1 TO n
3070  FOR j=last TO 255
3080  IF startval%(j)<>0 THEN spointer%
        (i)=startval%(j):GOTO 3100
3090  NEXT j
3095  PRINT"ERROR, startval not found":
        STOP
3100  last=j+1
3110  NEXT i
3120  GOSUB 4000
```

Lines 3000 through 3050 scan the sortval$ array and break out each value into a separate element of value$ for ease of look-up and sorting later. It relies on zero$ as a delimiter between values in sortval$. One note here may make things clearer. The whole reason why sortval$ was used instead of going with a string array for the values was because INSTR is an infinitely (nearly) faster way of searching for a given string value than a for-next loop plowing through value$, and since that operation has to be done for each record, speeding up the search was a critical issue. In any case, once value$ is built, a corresponding list of the start values for each string is built in spointer% by lines 3055 through 3110. This leaves us with a list of the actual values and the beginning values of the linked list for each. Now all that remains is to sort the values themselves and rearrange the spointer% list to match. That's done in the subroutine at line 4000:

```
4000  IF n<2 THEN RETURN
4010  span=INT(n/2)
4020  newspan=n-span
4031  FOR i=1 TO newspan
4040  temp=i
4050  upper=temp+span
```

```
4060  IF value$(temp)>value$(upper)
      THEN SWAP value$(temp),value$
      (upper):SWAP spointer%(temp),
      spointer%(upper):temp=temp-span:
      IF temp>0 THEN 4050
4070  NEXT i
4080  span=INT(span/2)
4090  IF span THEN GOTO 4020:
      ELSE:RETURN
```

That's right, campers, it's our old friend (well, not *that* old) the Shell sort. The only change is that we are sorting string data, and in addition to swapping the string values, we swap the spointer% values as well. The important thing here is that even though we may have processed thousands of records, we only have to sort the unique values among them. As long as we have lots more records than unique values, this routine will save significant amounts of time. Anyway, to finish up, once the values are sorted, we can assemble the whole list of record pointers by following the start values in the spointer% array and loading all the elements of the linked lists in the new order. That will finish the subroutine, and it looks like this:

```
3130  k=0
3140  FOR i=1 TO n
3150  index=spointer%(i)
3160  k=k+1:sarray%(k)=index:IF
      pntr%(index)<>0 THEN
      index=pntr%(index):GOTO 3160
3190  NEXT i
3200  RETURN
```

Notice that each linked list starts with an index in spointer% and ends when the value in pntr% is zero. By assembling the lists one by one in the sorted sequence determined by spointer% we guarantee that the whole list is in order.

Glancing way back up to the original main program, you can see that lines 196 through 220 can now take the sarray% list as a pointer list into the original records and print out that list in order.

It's hard to believe that the original bubble sort program in the beginning of this tome can be so short and so simple and yet take the longest to execute, while the last program, which seems so complicated and so long, can do certain types of sorts at least a hundred times faster. Many times it's not how much code is in the program, but how many times it must be executed, that really makes the performance difference. For that reason, in sorting as well as any other activity, it really pays to examine your loops and repetitive code and to think of the best algorithms possible. Remember, too, that the last program is useful only if there are only a few unique values, and the safest bet in the general case is the Shell sort.

Next time we will take up some other interesting sort techniques, including an improved Shell sort called the Quick sort and a completely different sort called the Binary Tree sort. The Binary sort, like the inverted list sort discussed in this article, can also be the basis for an access method. In fact, Apple III's *Record Processing Service* package uses a modified version of this algorithm. We hope to get the chance to get into those techniques as well. Until then, don't get out of sorts!

# Contest Winners

from page 8

ple, Jake went home to his warehouse to take a break and watch the *Wild, Wild West* on TV.

The two men were right next door to Jake working on a toilet seat heater when they heard a window-shaking, bone-jarring growl. At first they thought it was just the neighbor's Chihuahua, but it was coming from Jake's warehouse. They went next door to find Jake lying on his Persian rug grasping his stomach. The men quickly deduced that he was hungry, but the problem was what to feed him. They decided to try feeding him silicon chips, microprocessors, wires, printed circuit boards, and some other odds and ends. Jake happily wolfed down the goods.

They kept Jake on this diet for several days and also followed him on his usual route. One day on his route, they found a box with a keyboard on it. They picked it up and found several others on Jake's trail.

When they arrived at their lab, they hooked it up to a TV set. It looked like a computer! It worked like a computer! It smelled like a computer! It *was* a computer!

They rejoiced and went to tell Jake about their great find. When they arrived, they were amazed at all the computers lying around Jake's house. When they put two and two together, they realized that Jake was somehow reprocessing the odds and ends into computers.

"So," they said to each other, "these must be horse Apples!"

They all played Ring Around the Rosie and went into partnership selling Apple computers. Jake and those two geniuses are now making money together; occasionally the men feed Jake some higher quality parts and higher technology to make the Apple II, Apple II Plus, and the Apple III.

And that is how the Apple computer got its name. (Besides, did you ever hear of a horse kumquat!?)

When Samuel Swersky of Rockville, Maryland, sent in his entry for *Softalk*'s Etymology contest, he warned us, "My entry for the Kumquat contest is so good it really won't be worth your while to read any others. Though I've already received six-figure prepublication offers from such other journals as the *National Midnight Star* and the *National Enquirer*, I will probably mail it to you guys. So, get ready to fork over the loot."

Well, we received Swersky's entry, *Simon's Song*, and it was long. Really long. This sucker came in bound volumes. We are *not* talking short story. *Simon's Song* made the finals because of its sociopolitical color and international intrigue and was a favorite of many of those at the Kumquatorium. In this sampling, "Simon" refers to Simon Applebaum, the protagonist of the epic and the man to whom Swersky tells us we owe so much:

Postscript:
But though thousands now key in their programs,
And some are addicted to *Zork*,
If it weren't for Simon Applebaum,
We'd all be watching Mindy and Mork.

**The Also-Rans.** Well, that's it for the finalists. Here are some other entries that merit mention and a giggle or two.

Douglas Titchenal of Columbus, Ohio, tried to con us into picking his as the winning entry. His song, sung to the tune of Groucho Marx's *Hello! I Must Be Going*, ends thus:

> A is for abilities far-reaching. P is for potential yet untried. P again, this time for size and setup—compact, efficient, yet personalized. L is for the fine command of language. E for rating—excellent overall! Put them all together they spell Apple, and from its tree 100 bucks will fall!

Goodbye, Douglas, you must be going.

Jean Coulombre from North Quincy, Massachusetts, is our computer widow of the month. While her husband tapped away at the keyboard, he challenged her to enter the contest. Of course, Jean would have none of this "$100 in advertisers' goods" tomfoolery. If she had won, her husband would have selected the Videx Videoterm. Jean wanted $100 worth of new clothes, "and I want them now!" Come on, Mr. Coulombre, give her a break. Looks like you lost twice this month.

Some contestants added visual touches to their poems. Stephen Heywood (Framingham, MA), Ben Moore (Carpinteria, CA), and John D. Redfield (Forth Smith, AR) are credited for their hi-res verse and added artistry. Eleven-year-old Clint Wilson (Olathe, KS) sent us a hand-drawn computer with his story on the screen and the story's hero, Harold the Worm, wearing a cap with tassel at the keyboard. These entries are now on display in the *Softalk* Museum.

The also-rans will receive kumquats for their efforts. Be sure to contact us before stopping by to pick up your prize.

**Be True to Your School.** In the schoolhouse division, it was East versus West for the title of National Educational Kumquat Champion. In this corner, from the East, weighing in at nineteen double-spaced pages on the green and white computer paper, from Farrell, Pennsylvania, is the entire Farrell School District. And their opponent, representing the West, from the hometown of Apple Computer itself, Cupertino, California, weighing in at half of a double-spaced page and one-half hand-written page, are two fourth-graders from Faria School.

The match-up never even got to the first round because the Farrell District was using entrants in grades from elementary through high school. So, each school will reign as its respective regional champion.

From Faria, Martha Fulton's fourth-grade class was represented in the boys' division by Sam Chi. Here's his story:

> Why they named it Apple? I think because it looks good and it's fresh, and almost everybody likes apples, and it's shiny, and it tastes good. I think that's why they called it Apple Computer.

Representing the girls is Kelly L. McConnell, and here's her version:

> I think that the people that work with computers didn't want to have any of the other fruits like banana computer or orange computer. I think that "banana" computer sounds gushy, and "orange" computer sounds bumpy, but I think "Apple" computer sounds shiny, like when you first get a new computer it's really shiny. "Apple" computer also sounds very bright like when you turn on an Apple computer. Apple computer also sounds very strong like when you bite into an apple it is very hard. So why not a name that's shiny, bright, and strong, like an apple, for a great new computer. These are some of the many reasons why I think Apple Computer got its name.

Look out, Kelly; Chiat/Day, Apple's advertising agency, may be giving you a call with a job offer.

Over on the East Coast, Farrell really wanted the prize badly. Under the command of computer instructor Donna DeBonis, Farrell barraged us with more than fifteen entries. Leading the assault were the sibling troops of Frank, Gabriel, Michael, and Mina (two entries) Gargiulo.

But the best entry from the Farrell group was an atypical Adam-and-Eve story whose approach and style departed from other entries starring the Bible's first humans. Anita Shaw was the author of the masterpiece; but when we talked to the folks at Farrell, we discovered that Anita is the school nurse. We should have known that from her prize selection:

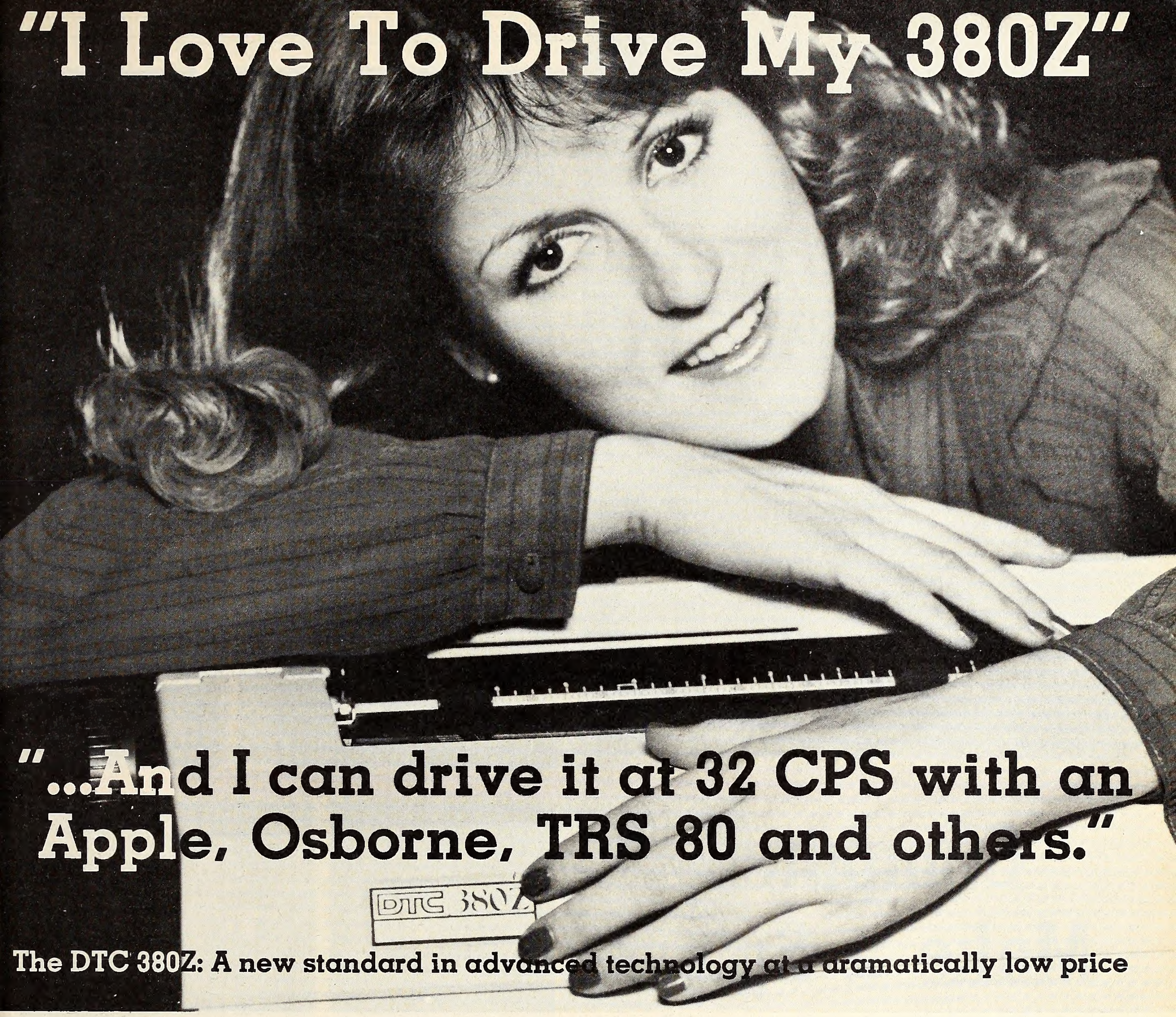
"If I win, I don't want a meager $100 worth of software. I want an entire Apple II system, complete with a printer, Disk II drive, a color CRT, and a selection of database management software programs. And I want it now!"

**Let's Get Personal.** And, finally, here's a letter we received, written on personalized stationery, that sums it all up:

Dear Mr. Kumquats:

> I would appreciate the chance to set the record straight. Surely Woz and Jobs selected "Apple" because the name denotes dignity, strength of character, grace under pressure, unusual intellectual capacity, sensitivity to the downtrodden, comfort in the company of the powerful, and humble, unassuming ways.

Sincerely, Cass Apple, Atherton, CA



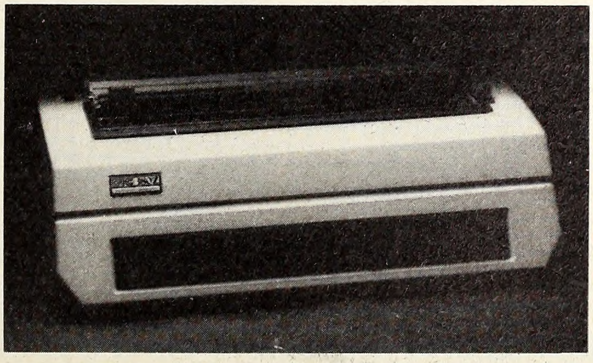

- **48K buffer for high speed throughput**
- Diablo 1640/1650/630 software compatible
- Serial and parallel interfaces
- Serial baud rates to 19.2K
- Automatic baud rate selection
- Four CPU's, no cables, belts, wheels or pulleys

- **Letter quality daisy wheel printer**
- 12 print styles, 15 languages, 96 char. wheel, standard ribbon
- Automatic bi-directional printing
- Proven reliability 100,000 printer mechanisms produced

- **Up to 32 CPS print speed in typical applications**
- Interconnecting cables for all major micro-computers
- Automatic proportional spacing
- Parallel 6K bytes per sec. transfer rate
- Optional extras: forms tractor, cut sheet feeder.

**MONEY BACK GUARANTEE**

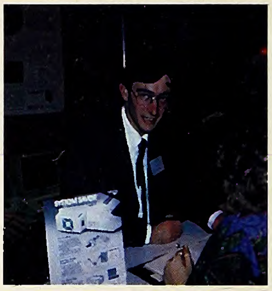
ORDER NOW
VISA • MASTERCARD
800-962-8185
in CA 800-538-9294

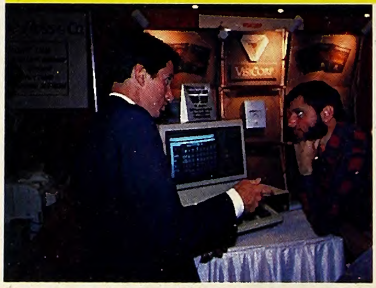


The DTC 380Z
**DAISY**
Wheel Printer
**$1199.**

DTC Data Terminals & Communications
590 Division St., Campbell, CA 95008









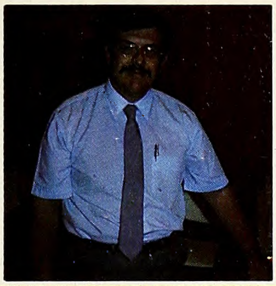

# Computers Make a Splash in the Land of 10,000 Lakes

BY DAVID HUNTER

When most people think of Minnesota they remember the bone-crushing Vikings's defense, or the migration of the Lakers, or Rod Carew, who came to fame with the Twins. Very few realize that Minnesota, especially Minneapolis, is quite active in the world of computers. Sports is big there, but computers might be bigger.

The home of Control Data (whose Plato is finding its way onto Apples), Honeywell, and Cray Research, Minneapolis seemed like a logical location for holding an Applefest. It's not quite Chicago, but it's closer than Boise, Idaho, Columbus, Missouri, or Gary, Indiana.

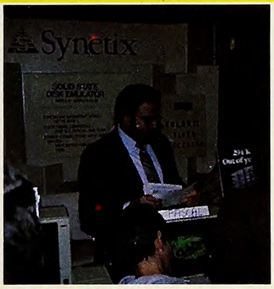
Mid-September was too early to go ice-fishing, a major winter pastime in Minnesota, and the Vikings were on strike along with the rest

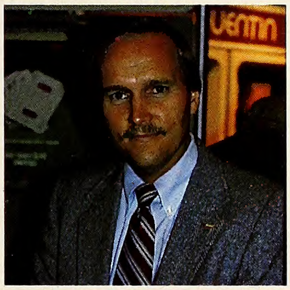
The show was held in the Convention Hall in refurbished downtown Minneapolis. This is the home of the Minnesota Educational Computer Consortium, but it was the Banana Educational Software booth that stole that show; Mitch Mitchells of distributor Micro D shows old favorite *Apple-Vision*; Joe Clark gave Savvy a public airing; Mark Machtems, Ken Staust, and Jim Libersky repped the Agricultural Data Services Inc.'s farm software; John Kuehn showed Blue Lakes's software.




Gordon Beckmann flew in from England to help demo Kensington's *Format II*; John Kolacke and Bob Vesaas showed business software from Goci Software; Bob Orr represented the popular BPI line of accounting packages; Frank Taft needed a breather as crowds kept the Hayden Software booth active; Mark Pelczarski and Dave Albert sprung a line of games on the unsuspecting public at the Penguin booth.




Phil Graves of VisiCorp shows *VisiCalc* to potential customer Ron Rothstein; Steve Freeman had more than RAM expansion boards to show at the Legend Industries booth—he also showed a five-megabyte hard disk; Charlie Harrison and Ken Bush hoopla'd *VisiCalc Formatting Aids* from Data Security Concepts; Synetix's Bob Shuman showed boards to emulate one or two floppies; B.S.G. Corporation's Dick Colness and Ann Hanson had software, disks, and sundry accoutrements.




Bob Ezzard kept crowds interested in the Alien Group's speech synthesizer; Mark Hoffman and Kim Gillespie displayed Quentin's line of floppy and hard disk drives; Brian Scott and Margie Reisner repped Cameo's hard disk with removable cartridges; Tim Hanson touted Micro-Sci's floppy disk alternative; Mike Larson had a full line of Wizard hardware expansion products from Wesper Microsystems.



F. Lee McFadden of Flip-Track Training Tapes had audio cassette instructional materials; Barry Fleig showed WIDL's line of Apple directories; Ben Larson of Kraft drew crowds by demoing joysticks with *Choplifter*; Anne Kurth showed gawkers the value of a buffer card at the Practical Peripherals booth; music was in the air when John Paulson was at the keyboard in the Syntauri booth.




Rick Geshwiler of Yukon Computing talked over business with Bill King and Rich Jenkins of Orange Micro; Paul Wengler and Glenn Dunk showed Cascade's graphics development system; a big hit was the Genius, an eighty-column, sixty-six line display device from local company Micro Display Systems; Alan Baker, Sally Rush, and Steve Schultz presented the *Key Wiz VisiCalc* aid from Creative Computers; Gordon Mitchell and Carolyn Marquardt showed Bit 3's eighty-column board.


Left to right, first row: Joan Hayden showed the PKASO board from Interactive Structures; Chuck May boosted Epson printers; the Mini'app'les Users Group made a bid for members; Dave Simpson pushed the line of surge suppressors from Electronic Protection Devices. Second row: Chuck Anshell, Pearl Anshell, and Mitch Johnson of Com-Pute Services; Bob Olson of National Business Systems was touting BASF disks; Mike Hartstein, the research and development director of Software Connections, viewed the action. Third row: Chicago visitors included Steve Shendelman from Data Domain; Donald Brown of C.E. Software and *SwordThrust* fame was an interested observer; Dolores Hagen of Closing the Gap chats with attendee Carol Cole; Susan Mackert repped *Printout* magazine.

of the National Football League. The Twins were in last place. In short, there wasn't much else to do in Minneapolis that weekend. Barring any unpleasant surprises, the timing seemed perfect.

The big surprise at the Minneapolis Applefest was that it was also the Twin Cities Computer Show. Running one on one with Apple exhibitors were the likes of Radio Shack Computer Centers, Datamark Business Systems, Northwestern Bell, and CompuScribe. Total attendance for the four-day show exceeded all estimates, but there were still grumbles from exhibitors who expected an exclusively Apple affair.

Big surprises behind us, at Minneapolis there was little of the excitement and panache that have characterized the previous Applefests in Boston. Exhibitor turnout was noticeably light. The West Coast was represented by only a handful of companies; R.H. Electronics, Cameo Electronics, Syntauri, Passport Designs, and Orange Micro were about it. The East Coast pulled a little better, while the majority of exhibitors originated from the Midwest, mainly Illinois and the Minneapolis area.

Despite the poor showing overall, this Applefest was successful on other levels. Local exhibitors added a new flavor to the show not always found in such computer-conscious environments as San Francisco. The pace was slower and the stakes smaller. Exhibitors and attendees achieved greater communication than in larger, more competitive affairs.

Perhaps it was the Midwest and its residents' reputation for being straightforward, hard-working, and honest. Without a doubt, the average attendee at this fair was serious about computers and not afraid to ask questions. Even those who didn't own a computer seemed like they sincerely planned to get one.

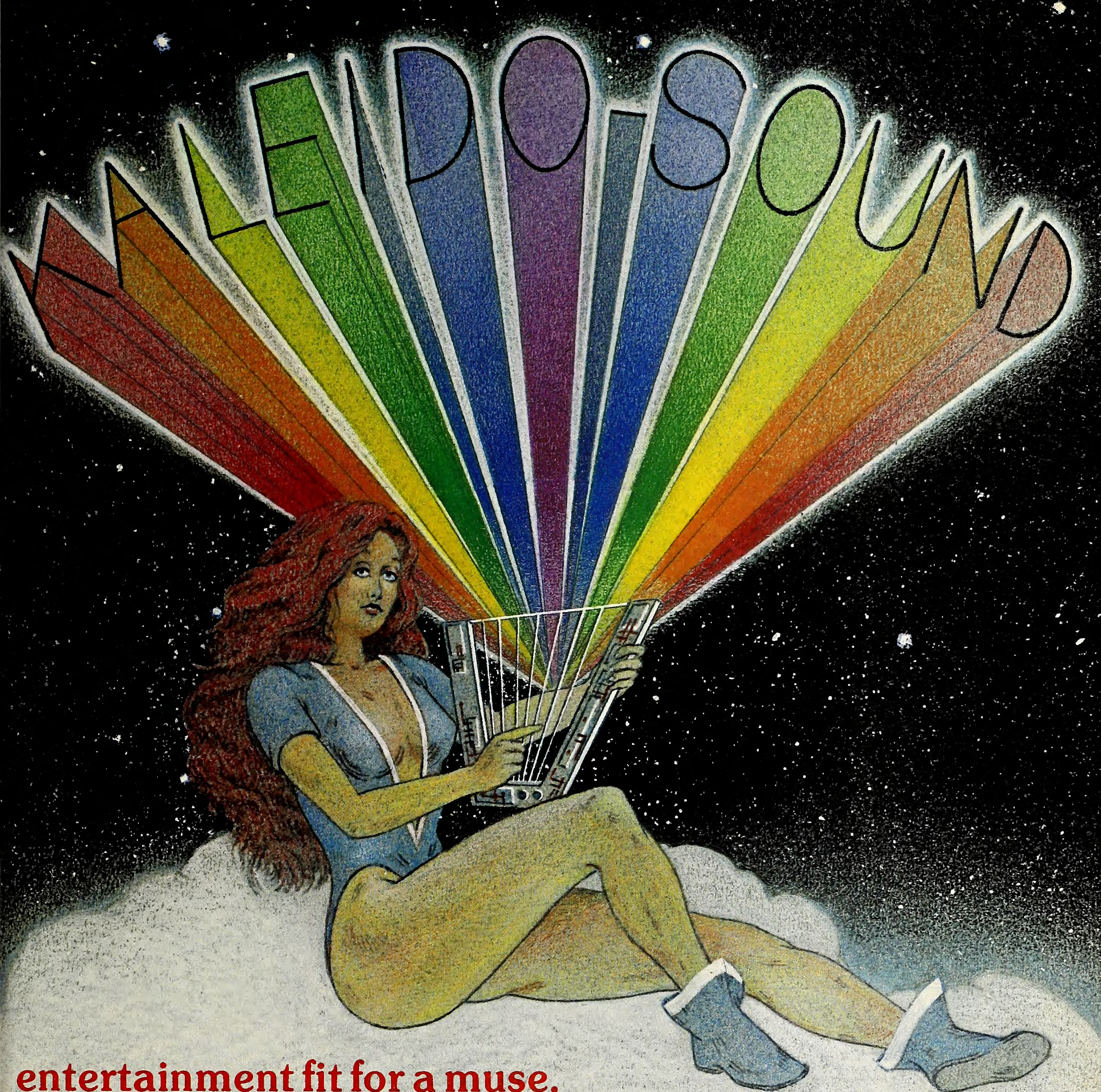
Business programs and peripherals were in abundance, while there was only a smattering of games. Hardware highlights were provided by Micro Display Systems (Hastings, MN), with its generic full-page display monitor called the Genius, and Cameo Electronics's new Winchester hard disk with a removable cartridge. Rhino Robots (Champaign, IL) drew quite a crowd around their booth all four days. Their pick-and-place tabletop robot arm, called the XR-1, is meant for educational and research applications.

Dayton's, a top-of-the-line department store chain in the Midwest, had an impressive booth, punctuated with ten-foot-tall gray monoliths. Selling systems and software, Dayton's Computer Centers dramatically demonstrate the possibility of offering more than just computer games in big department stores.

A generous spirit was displayed by some of the exhibitors. Northwestern Bell gave away cigarette lighters embossed with their slogan "The Knowledge Business." *Softalk* gave away InvisiTabs. R.H. Electronics held a drawing for nine top prizes. All nine winners in the R.H. contest came from the Minneapolis/Saint Paul area, including small towns like Chaska, Burnsville, and Maple Grove.

Minnesota is a major agricultural state and the concerns of farmers were addressed by Countryside Data, a farm software distributor. More than one educator gathered under the bright yellow banner of Banana Educational Software and discussed the present and future of learning with Apples.

When all was said and done, the Minneapolis Applefest was enjoyable because the pressure was off. It was a long four days, but it had none of the madness of the West Coast Computer Faire or the National Computer Conference. Minneapolis will be remembered for sophisticated attendees and their irrepressible curiosity about Apples. Plenty of money passed hands and yet Mammon paled next to the gods of computing. ■




## entertainment fit for a muse.

KALEIDO-SOUND. A new concept in entertainment software from Passport Designs. Simply plug your stereo, cassette player or sound system into your Apple II™ and let KALEIDO-SOUND turn your music into brilliant full color graphics.

Imagine – four different kaleidoscopic patterns with selectable color schemes pulsating in perfect harmony with your favorite music. Add a fantastic new dimension to parties, concerts or your home entertainment system.

KALEIDO-SOUND runs on a 48K Apple II plus. Plug an audio signal into the cassette port and see what you've been missing.

KALEIDO-SOUND. Made for mortals but fit for a muse.

**$39.95**

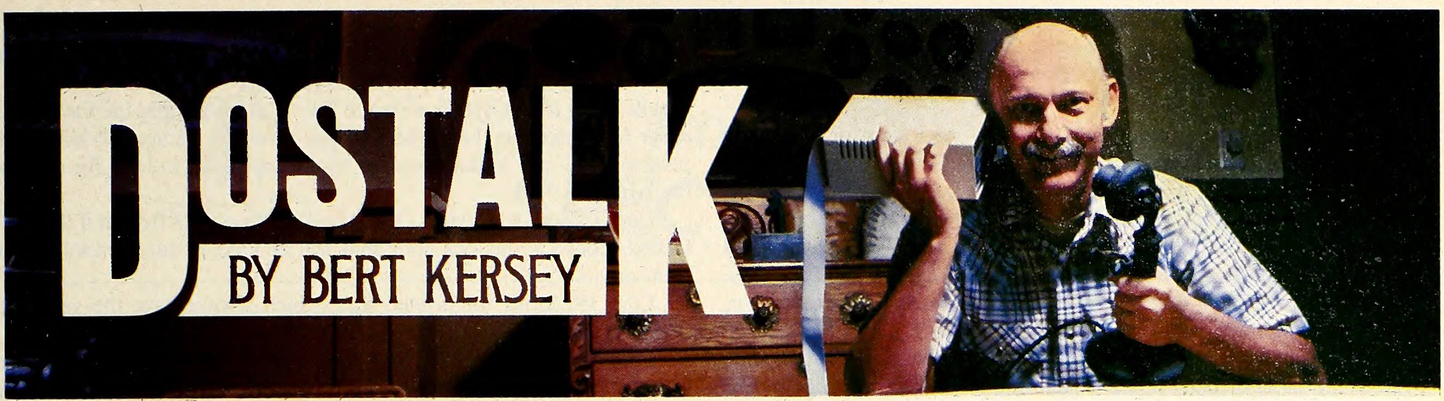
Add $1.00 for direct orders.
VISA, MASTERCARD, check or COD accepted.

Get it at your local dealer or order direct from:

PASSPORT DESIGNS INC

116 North Cabrillo Hwy., Half Moon Bay, CA 94019
(415) 726-0280


When all hell breaks loose.
Pandora's box is open. And all the evils of the past are loosed upon mankind. Armed with bolts of lightning, you have the chance to recapture and return these corrupt creatures of doom to the prison of Pandora's box. Time is short. The world is already changing for the worst. Your skills and courage are needed.
PANDORA'S BOX
Pandora's Box—a new generation type game with overtones of Hi-res adventure graphics.
Only $29.95 for the Apple II*
At your computer store, or:
DATAMOST INC.
9748 Cozycroft Ave., Chatsworth, Ca 91311 (213) 709-1202
VISA/MASTERCARD accepted. $2.00 shipping/handling charge.
(California residents add 6½% sales tax.)
*Apple II is a trademark of Apple Computers, Inc.



# DOSTALK

BY BERT KERSEY

Let's continue last month's sector-reading experiments and write a useful utility that sorts catalogs by file type and puts available file slots where you want them. We'll actually be writing data to a disk this time, so make some back-up copies of a disk that's seen lots of action (loading, saving, deleting, and so on) and join the fun!

**A Quick Review.** A disk's catalog is stored on track 17, with seven file names in each 256-byte sector of that track. The first seven file names are in sector 15, the second seven in sector 14, and so on, down to sector 1. The first eleven bytes of each sector are wasted, with the exception of bytes 2 and 3, which point to the next sector of the catalog. The last 245 bytes of each sector contain seven thirty-character file names and five bytes of file information, with thirty-five bytes allotted per file. Got it? If not, then goto October.

The objective of this month's utility is to regroup the files on a disk by file type (B, A, I, and T). We also want to group deleted file names (which remain in a catalog until overwritten) so that we can predict where the next file names written to the disk will appear. Running the *Bait Sort* program reads track 17 of the disk in the drive sector by sector, prints a sorted catalog on the screen, and then lets you decide whether to "permanently" write the changed catalog to disk.

Armed with the techniques outlined here, you should be able to expand the program so that it can alphabetize files and do all sorts of interesting things. For now, we'll leave alphabetizing to someone else's colums (SortTalk?).

Here's the program; have at it.

**Bait Sort**

```
100  FILE$ = "AIDTB"
110  REM FILE$=ORDER OF FILE-TYPES (MUST CONTAIN
     B A I T & D)
120  FOR I = 1 TO 5:X$ = MID$ (FILE$,I,1):A(I) = 5 - (X$ = "B") - 2
     * (X* = "D") - 3 * (X$ = "A") - 4 * (X$ = "I") - 5 * (X$ = "T"):
     NEXT : REM T=0, I=1, A=2, D=3, B=4
130  FOR I = 1 TO 5:B(A(I)) = B(A(I)) + 1: NEXT : FOR I = 0 TO 4: IF
     B(I) = 1 THEN NEXT : GOTO 150: REM CHECK FILE CODES
140  PRINT "ILLEGAL FILE ARRANGEMENT: ";FILE$; CHR$ (7):
     END
150  F$(0) = "T":F$(1) = "I":F$(2) = "A":F$(4) = "B":F$(3) =
     "<DEL>"
160  POKE 47083,0: POKE 47091,0
170  POKE 768,32: POKE 769,227: POKE 770,3: POKE 771,76:
     POKE 772,217: POKE 773,3: REM JUMP TO RWTS
180  DIM L1(15),L2(15): FOR SEC = 1 TO 15:L1(SEC) = 10000 + (15
     - SEC) * 256:L2(SEC) = L1(SEC) + 10000: NEXT : REM
     STORAGE LOCATIONS
190  POKE 47092,1: REM 1=READ SECTORS
200  TEXT : HOME : VTAB 24: PRINT "READING SECTOR:";
210  FOR SEC = 1 TO 15: HTAB 16: PRINT SEC;: POKE 47084,17:
     POKE 47085,SEC: REM TRACK 17, SECTOR SEC
220  L = L1(SEC): POKE 47088,L - INT (L / 256) * 256: POKE 47089,
     INT (L / 256): CALL 768: REM STORE FIRST TIME
230  L = L2(SEC): POKE 47088,L - INT (L / 256) * 256: POKE 47089,
     INT (L / 256): CALL 768: REM STORE SECOND TIME
240  NEXT SEC
250  NF = 0:NS = 15: REM FILE AND SECTOR COUNTERS
260  LOC = L2(NS) + 11
270  HOME : VTAB 24: PRINT "SORTING CATALOG:": PRINT : REM
     START SORT
280  FOR N = 1 TO 5:TYPE = A(N)
290  FOR SEC = 15 TO 1 STEP - 1
300  FOR X = L1(SEC) + 11 TO L1(SEC) + 221 STEP 35: REM LOOK
     AT EACH FILE NAME
310  IF PEEK (X) = 0 THEN 420: REM IF END OF CATALOG
320  P = PEEK (X + 2):LOCK$ = " ": IF P > 127 THEN P = P -
     128:LOCK$ = "*": REM SUBTRACT 128 IF LOCKED
330  IF PEEK (X) = 255 AND TYPE = 3 THEN 360: REM IF DELETED
     FILE
340  IF P > 4 THEN PRINT : PRINT "NONBAIT FILE: ";: FOR I = X + 3
     TO X + 32: PRINT CHR$ ( PEEK (I));: NEXT : STOP : REM IF R
     OR S TYPE
350  IF (P <> TYPE) OR (P = TYPE AND PEEK (X) = 255) THEN
     410: REM IF NOT RIGHT FILE TYPE, THEN SKIP
360  FOR I = X TO X + 34: POKE LOC, PEEK (I):LOC = LOC + 1:
     NEXT : REM POKE FILE NAME INTO NEW LOCATION
370  PRINT LOCK$;F$(TYPE);: IF TYPE = 3 THEN 390
380  PRINT SPC( 1); RIGHT$ ("00" + STR$ ( PEEK (X + 33) + PEEK
     (X + 34) * 256),3);: REM PRINT SECTOR NUMBERS
390  FOR I = X + 3 TO X + 32: POKE 2004 + I - X, PEEK (I): NEXT :
     PRINT : REM PRINT FILE NAME
400  NF = NF + 1: IF NF - INT (NF / 7) * 7 = 0 THEN NS = NS - 1:
     LOC = L2(NS) + 11: REM DECREASE SECTOR EVERY
     SEVENTH FILE NAME
410  NEXT X: NEXT SEC
420  NEXT N
430  PRINT : PRINT "WRITE TO DISK? (Y/N):";: GET A$: PRINT A$:
     IF A$ <> "Y" THEN END
440  PRINT : PRINT "WRITING SECTOR:";
450  POKE 47092,2: REM 2=WRITE SECTOR
460  FOR SEC = 1 TO 15: HTAB 16: PRINT SEC;
470  POKE 47084,17: POKE 47085,SEC: REM TRACK 17, SECTOR
     SEC
480  L = L2(SEC): POKE 47088,L - INT (L / 256) * 256: POKE
     47089,INT (L / 256): CALL 768
490  NEXT SEC
500  PRINT : PRINT CHR$ (4);"CATALOG"
```

If you don't care how the program works, no problem; just type it, save it, and run it with an expendable disk in your drive. If you want to learn a few things, however, read along. . . .

**The Bait Sort Rundown.** Line 100 contains the string file$, which determines the order of our sort. File$ must contain only the letters B, A, I, T, and D, standing for Binary, Applesoft, Integer, Text, and Deleted. Each letter must appear only once and must fall in the order that we want our catalog's file names to appear. Lines 120 through 140 proofread file$ for us.

Line 150 sets the array F$( ) according to the file-type codes predetermined by DOS. This string array will be used in our dummy print-out of the sorted catalog. Remember, byte 3 of each file's thirty-five bytes says what file type it is:

0 = Text
1 = Integer
2 = Applesoft
4 = Binary

If the particular file is locked, the value 128 is added to these values. Since the code 3 is unused, we've used that number to look for deleted files. DOS actually designates deleted files by a 255 in a file's first byte.

Line 160 pokes a couple of default zeros into the *Input/Output Control Block*. *Poke 47083,0* accepts any volume number. *Poke 47091,0* tells the program to read each sector in its entirety. Other numbers in this block (locations 47080 to 47096) will be poked soon. Stand by. . . .

Line 170 pokes in our little machine language jump to DOS's RWTS (Read-Write-Track-Sector) routine at location 768 ($300):

```
300-   20 E3 03 JSR    $03E3
303-   4C D9 03 JMP    $03D9
```

Just think of this as computer code for "Go read from or write to a disk." From Basic, *call 768* is the trigger.

**Data Duplication.** Line 180 determines where in memory to store the data to be sorted. For sorting purposes, we'll store two copies of the catalog, so two sets of locations are chosen: sector 15's 256 bytes will be read from the disk and stored first at location 10000 or L1(15), and then at location 20000 or L2(15). Sector 14 will be stored twice, 256 bytes higher, and so on, continuing through sector 1.

Line 190 sets the pointer telling RWTS to read the disk rather than write to the disk. Be careful; a 2 poked into 47092 in this program line would probably ruin your disk (not to mention your day).

Line 210 starts the loop that reads our fifteen sectors. A 17 is poked into location 47084, designating track 17. This poke could have been done prior to the loop, by the way. Each sector number (SEC) is poked into 47085. Lines 220 and 230 first tell RWTS where to store data. For poking purposes, L1$( ) and L2$( ) need to be broken into two bytes each, high and low order. *Call 768* triggers the reading of sector SEC.

Now our sort search begins. We'll look for each file type in the first storage area, L1(15) through L1(1), and store each file name found and its accompanying data in the second storage area, L2(15) through L2(1). Every file name on the disk must be accounted for (once only!) in order for our sort to be successful. Line 250 starts our sort search by setting the number of files (NF) and number of sectors (NS) counters to zero—tidy, but not really necessary since they haven't been used yet.

The variable LOC in line 260 holds the location of the byte where the next piece of found data will be stored. LOC is increased by one after each poke in line 360 and by eleven after each sector is full in line 400.

Line 280 starts the actual search through the five types of files. The value of the variable *type* tells the program what value to look for in the third byte of each file.



Line 290 starts a search loop on sector 15 and counts down to sector 1. Line 300 finds the first byte of each file's thirty-five bytes, then line 310 checks to see if that byte is a zero. If it is, that means that the file slot has never been used, so there won't be any more file names following. The program will quit looking and go to the beginning to look for the next file type (*next N*).

Line 320 looks at byte 3 (X+2) and subtracts 128 from it if the file is locked. Lock$ is the lock code that will be used in our dummy catalog print-out.

Line 330 looks for deleted files, which always have the value 255 stored in byte 1.

Line 340 looks for R, S, and other odd file types and stops the program if they are found. We're avoiding these rare file types mainly because doing so makes this program much simpler to explain. Include them if you want. Unlocked S files have a value of eight in byte 3. R files use a sixteen.

Line 350 checks for the correct file type and skips to the end of the loop (line 410) if it's not found. If we have successfully located a file, line 360 copies the file's thirty-five bytes to location LOC and consecutive bytes.

Lines 370 and 380 print the file locked code, file type, and sector numbers for our dummy catalog. Then line 390 pokes the file's name (bytes 4 through 33) onto the bottom line of the screen, or into memory locations 2007 through 2036. This line could read:

```
390  VTAB 24: HTAB 7: FOR A=X+3 TO X+32: PRINT
     CHR$(PEEK(A));: NEXT: PRINT
```

except that the program would then be unable to print non-normal (flashing, inverse, and control) characters. Finally, line 400 counts found file names. On every seventh name, we must jump eleven characters to allow for the eleven bytes at the beginning of each new sector. Lines 410 and 420 complete the loops.

**No More Fooling Around.** Line 430 makes sure we like what we see on the screen. So far, we've only played around in memory. You can fool with the program and debug it to your fingers' desire up to this point, but now things get dangerous!

We are about to (fanfare?) write to the disk! If there are *any* mistakes in this program code other than errors that halt execution, it's almost guaranteed that lines 450 through 490 will render the files on your disk inaccessible (although the disk itself can be reinitialized). So use a disk you didn't really like anyway, fearlessly run the program, and answer "Y" to the query in line 430.

The next few lines set the parameters for the RWTS routine. Line 450 pokes a big 2 into location 47092, telling RWTS to write instead of read, and line 460 starts the write loop. Then line 470 pokes the track and sector values into 47084 and 47085, the same as line 210 back when we were reading the disk. Finally, line 480 tells RWTS where in memory to get the data that will be written to the disk.

Line 500 is the moment of truth.

There you have it. The deleted files that had a "<DEL>" printed next to them in the dummy catalog won't show when you actually catalog the disk, of course, even though they are there. Deleted file names are significant, however, because the next file saved, bsaved, or written to the disk will appear in the slot occupied by the first deleted file name (or at the bottom of the catalog if there are no deleted files). If you put deleted file names, for example, right after your Applesoft file names, your next saved file name will appear, logically, at the bottom of that group. If you want, you can put a loop in the program that will insert as many deleted file names as you want into your new catalog. Just make the program repeat the same deleted file name several times. Or store a bunch of dummy file names on the disk and then delete them before running *Bait Sort*, like so:

```
10  FOR FILE=1 TO 25: PRINT CHR$(4); "OPEN FILE ";FILE
20  PRINT CHR$(4); "CLOSE": NEXT
30  FOR FILE=1 TO 25: PRINT CHR$(4); "DELETE FILE ";FILE
40  NEXT
```

That's all for now.

# HARDTALK

## BY JEFFREY MAZUR

While there is certainly a wide range of printers to choose from, there may be even a bigger selection of printer *interface* boards for the Apple. The March installment of this column described how these boards work and evaluated most of the products that were available at that time. Since then, quite a few additional devices have emerged, many of which provide a lot more features than just a connection between printer and computer.

**PKASO Printer Interface.** The PKASO printer interface boards from Interactive Structures provide comprehensive text and graphics capabilities for most popular printers. The on-board firmware allows numerous printer operation and formatting commands. Figure 1 lists many of these commands.

As you can see, all the standard Apple-convention commands, such as control-I 80 N to set the line length to eighty characters, are available. Some of the other commands merit explanation as well.

The B command allows you to send any ASCII character to the printer by entering its equivalent decimal value. This capability is used by several of the printer mode commands (see figure 2). For text and graphics screen dumps, several options can be selected, including size, polarity, screen, and so on. These options are controlled by the number preceding the command character and may be combined for various effects. Thus, a rotated print of hi-res screen 2 could be generated by typing *control-I 9 H.* The 9 is derived by adding the value for screen 2 (1) and the value for the rotate command (8).

The F command is a simple way of setting the font size for printers that support multiple fonts. In the PKASO board we tested, which was intended for use with the NEC 8023A and C. Itoh 8510 printers, there were also commands that enabled the user to select the normal and double size proportionally spaced fonts.

Control-I M is a lead-in to another series of commands that control the printer mode. These commands vary according to which printer version of the card you have. Figure 2 lists the commands for the NEC/C. Itoh printers.

Some of the commands listed in figure 2 must be followed by one or more parameters. These parameters can be sent with the control-I B function. For example, the vertical line spacing can be set to 16/144-inch by means of the command sequence control-I M T control-I 16 B.

Besides being able to do hi-res graphics dumps, the PKASO board can produce lo-res pictures. In this mode, the board produces sixteen shades of gray—one for each of the lo-res colors. Since some of the Apple's colors have the same brightness level, a bit of translation is required to produce the distinct gray scale. While the end result may look different from what is seen on the screen, it can be quite pleasing when viewed from a distance.

Figure 3 shows how a hi-res picture with sixteen gray levels can convey more information than a normal hi-res picture. If you find this hard to believe, consider the television picture you're accustomed to looking at. The horizontal resolution of a black-and-white television picture is not that much greater than the Apple's 280 pixels. One reason the television picture looks so much better is that it has a continuous gray scale (equivalent to more than two hundred discrete levels between white and black) that carries a significant portion of the information you take in when viewing a single frame. When color and real-time moving images are added, you can begin to appreciate how much further computer graphics must progress in order to match the quality of film or television.

Another unique feature of the PKASO board is the ability to print special characters. Loading a small driver routine and character set makes possible the embedding of special symbols, subscripts, superscripts, and so on within ordinary text. These special characters are printed in the same way ordinary text is printed; thus, they can be accessed by your favorite word processor with little trouble. A copyright symbol, for example, might be produced by typing control-I 1 S 5 control-I S. In this

| | |
|---|---|
| Control-I n B | Sends binary code n directly to the printer. |
| Control-I D | Prints text screen 1. |
| Control-I 1 D | Prints text screen 2. |
| Control-I 2 D | Prints lo-res graphics screen, inverted. |
| Control-I 4 D | Prints only the lower four lines of the screen. |
| Control-I 10 D | Prints the lo-res graphics screen. |
| Control-I 14 D | Prints lo-res mixed graphics and text (four lines). |
| Control-I E | Ejects the page (top-of-form). |
| Control-I n F | Sets character font to fit n columns within an eight-inch line. |
| Control-I n G | Gray scale mode (prints n pixels). |
| Control-I H | Hi-res graphics print, standard screen 1. |
| Control-I 1 H | Hi-res graphics print, screen 2. |
| Control-I 2 H | Large size hi-res print. |
| Control-I 8 H | Ninety-degree rotated hi-res print. |
| Control-I 16 H | Left-justified hi-res print. |
| Control-I 32 H | Inverse (white on black) hi-res print. |
| Control-I I | Turns on video display, sets line length to 40. |
| Control-I K | Disables automatic line feeds. |
| Control-I n M | Sets printer modes (see figure 2). |
| Control-I n N | Turns off video display, sets line length to n. |
| Control-I S | Selects standard character set. |
| Control-I 1 S | Selects special character set. |
| Control-I T | Tabs to next field (in even multiples of eight columns). |
| Control-I n T | Tabs to column n. |
| Control-I n X | Switches to special characters using external driver routine. |
| Control-I control-X | Changes command lead-in character to control-X. |

Figure 1. Commands summary for the PKASO printer interface boards.

| | |
|---|---|
| Control-I n M | Sends the escape character, followed by the character whose ASCII value is n (if n is zero, then only the escape character is sent). |
| Control-I 102M | Selects forward line-feed direction. |
| Control-I 114M | Selects reverse line feed. |
| Control-I M $ | Selects alphanumeric character set. |
| Control-I M # | Selects internal graphics character set. |
| Control-I M ! | Selects enhanced mode. |
| Control-I 34 M | Clears enhanced mode. |
| Control-I M S | Selects bit image graphics mode. |
| Control-I M A | Sets vertical line spacing to six lines per inch. |
| Control-I M B | Sets vertical line spacing to eight lines per inch. |
| Control-I M T | Sets vertical line spacing to n/144-inch per line. |
| Control-I M ( | Sets printer's internal horizontal tab stops. |
| Control-I M ) | Clears selected internal tab stops. |
| Control-I M 2 | Clears all internal tab stops. |
| Control-I M X | Selects underline mode. |
| Control-I M Y | Clears underline mode. |
| Control-I M L | Sets printer's left margin. |

Figure 2. Printer mode commands for the NEC 8023A/C. Itoh 8510.

# Compatible color . . . monitors


## . . . for demanding graphics or text display.

For high resolution (560H X 240V) color graphics, you can't beat the Amdek Color-II Monitor. And if you're looking for economy, the Color-III Monitor with 260H X 300V resolution is a superb buy.

Both monitors feature RGB video input for computer controlled color . . . and Amdek's easy-to-install Digital Video Multiplexor (DVM) board permits interface with the most popular 80 column boards.

Just call, or write for full details.

- Color-II Monitor has RGB input and 560(H) X 240(V) line resolution.
- Color-III Monitor has RGB input and 260(H) X 300(V) line Resolution.
- Digital Video Multiplexor (DVM) assures color graphics interface with most popular 80 column boards, such as: Videx "Videoterm", Advanced Logic "Smart-term", M&R "SUP 'R' TERMINAL", Bit-3 Computer Corporation "FULL VIEW-80", and the "Doublevision" boards.

2201 Lively Blvd. • Elk Grove Village, IL 60007
(312) 364-1180 TLX: 25-4786

AMDEK CORP.

**Amdek** . . . your guide to innovative computing!

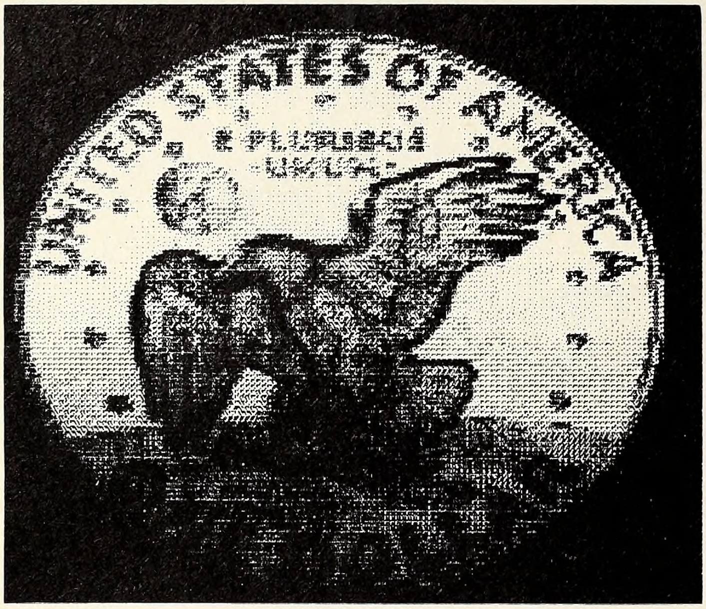


Figure 3. Halftone graphics with the PKASO board.

instance the new symbol was previously defined as the special character for the number 5. If your word processor does not allow control characters in the file, the lead-in character can be changed from control-I to an unused printable character such as ∧ (shift-N).

Any commands that are sent to the PKASO board remain intact when the board is deselected and reselected. This makes it possible to set up the various parameters for the card before running a program that uses it. Another feature of the PKASO board is printer status indication. If the printer runs out of paper or is in the local mode, a flashing P or L will appear in the lower right corner of the screen. This is a nondestructive display; anything that was previously on the screen in that location will be replaced when the printer condition is remedied.

The printer must be on and ready to print whenever the Pascal or CP/M operating system is booted. When the board is operating in these systems, the lead-in character automatically defaults to control-Y since control-I is used as the tab command. Another useful feature of the system is the capability to set strobe polarity and to enable interrupt-driven operation from machine language. This information is supplied in a well-written and produced owner's manual.

All in all, the PKASO board is a very powerful interface for your printer. There are only two aspects of the board that could cause trouble. One is that the graphics dump command formats are unique to this board. Most graphics boards have conformed to the format used by the Grappler from Orange Micro. These commands, such as control-I G D I R to print out a graphics screen double wide, inverted, and rotated, are a little easier to remember. Because this format has become somewhat of a standard, programs designed to interface with graphics boards may have to be modified to work with the PKASO.

The other possible drawback to the PKASO board is that several common functions, such as page length control (when to skip over page perforations) and high order bit control, are not provided. These functions are not often used and may be provided external to the interface board (for instance, from a word processor); therefore, in light of the many positive features of the PKASO, it is a very good buy at $165. This price includes a cable and a utility disk containing sample pictures.

**PKASO ID12 Color Printer Interface.** The PKASO ID12 interface provides all of the functions we've just been talking about plus some additional commands that can be used to control the color printing capabilities of the Prism printer from Integral Data Systems. This printer uses a four-color ribbon to create color print-outs. Any color (yellow, magenta, cyan, or black) can be selected manually with the control-I n R command. Each of the Apple's lo-res and hi-res colors can be produced by using a combination of ribbons (see figure 4), or you can let the firmware dump full-color print-outs of the graphics screen automatically.

Mixed colors are printed by making several passes of the printhead across the paper. The lightest color is printed first. Then the vertical position of the ribbon is changed to expose the next color. Another pass is made, placing a dot of the new color over the previous one. To avoid contamination, colors are always printed starting with the lightest color ribbon. Contamination happens when the second color is printed directly over a previous dot. As the ribbon contacts the paper, a small amount of ink from the first color can get transferred back onto the ribbon (of the wrong color). Obviously, it's much better to get a little yellow ink on the black portion of the ribbon than vice versa. The PKASO board also has a special Mosaic mode that reduces this problem even further. More on that later.

Notice in figure 4 that the board prints black for white and white for black. This arrangement is similar to the way other graphics dumps swap the two to prolong ribbon life. This also corresponds more naturally to the video picture, even though the background on the screen is black as opposed to the white background of paper. With a simple command, the true black and white can be restored. For lo-res graphics dumps, two other color schemes are possible. Besides the equivalent-color representation, a set of spectrum colors or sixteen levels of a single color can be selected. This is shown in figure 5.

| Hcolor | Color Displayed | Color Printed | Combination of Ribbons |
|---|---|---|---|
| 0 | Black | White | None |
| 1 | Green | Green | Cyan + Yellow |
| 2 | Violet | Violet | Magenta + Cyan |
| 3 | White | Black | Black |
| 4 | Black | White | None |
| 5 | Blue | Blue | Cyan |
| 6 | Orange | Orange | Yellow + Magenta |
| 7 | White | Black | Black |

Figure 4. Printing of Apple's hi-res colors.

# EVERYONE NEEDS A . . .


## UNINTERRUPTABLE POWER SOURCE

**A**DVANCED DESIGN
**N**EW RELIABILITY
**G**UARDIAN PROTECTS
**E**LECTRONIC SAFETY
**L**IFESAVER FOR DATA

**$595**

**SAVE YOUR DATA FROM POWER OUTAGES!**

**BACKUP FOR YOUR COMPUTER, MONITOR, PRINTER AND 5¼″ FLOPPY AND HARD DISC DRIVE**

• Automatically stops annoying problems from power line interruptions and brown outs • You need standby power to save data • Maintenance free backup power available in 115 volt or 220 volt • 50 or 60 HZ • 150 watts • Complete versatility — operate your system from a 12 volt source, i.e., automobile cigarette lighter, boat or airplane • Rugged self contained gel cell battery • No voiding warranty — no cutting wires • Automatic audio alarm warning tone during commercial power failure or interrupt • UL listed • FCC approved • Transient voltage suppressor gives added insurance from line voltage spikes, utilizing Zener Ray™ • Green/red LED power status indicator • Green — normal AC line power • Slow blinking red — at least 6 minutes of remaining standby power • Fast blinking red — approximately 2 minutes of remaining battery power • Solid state technology unexcelled by any UPS power unit in its class.

RH ELECTRONICS, INC.

566 IRELAN, BUELLTON, CA 93427
(805) 688-2047

**SEE YOUR RH ELECTRONICS PRODUCTS DEALER**

**FOR YOUR APPLE II*:**

| | |
|---|---|
| SUPER FAN II™ | $ 74.95 |
| SUPER FAN II™/ZENER RAY™ | $109.00 |
| SUPER RAM II™ | $125.00 |
| RH 12 VOLT TRANSVERTER | $149.00 |

**FOR MICRO COMPUTERS:**

| | |
|---|---|
| GUARDIAN ANGEL™ | $595.00 |

| Color= | Screen Color | Spectrum Color | Single Color |
|---|---|---|---|
| 0 | White | White | White |
| 1 | Magenta | Magenta | Darkest |
| 2 | Dk.Blue | Red | . |
| 3 | Violet | Orange | . |
| 4 | Dk.Green | Yel-Orange | . |
| 5 | Gray | Yellow | . |
| 6 | Blue | Yel-Green | . |
| 7 | Lt.Blue | Green | . |
| 8 | Brown | Chartreuse | . |
| 9 | Orange | Cyan | . |
| 10 | Gray | Blue | . |
| 11 | Pink | Violet | . |
| 12 | Lt.Green | Red-Violet | . |
| 13 | Yellow | Gray | . |
| 14 | Aqua | Brown | Lightest |
| 15 | Black | Black | Black |

Figure 5. Printing of Apple's lo-res colors.

For large print-outs of low resolution, it is possible to "paint" pictures from a sixty-four–color palette that the PKASO board can manage for you. Each color pixel consists of a 6 by 5 matrix of dots. The way to produce any color in the palette is specified by a four-byte definition. This allows each color to be specified by an eight-level degree of saturation in each of the four ribbon colors. That's more than four thousand possible colors (although many would be indistinguishable from one another).

To deal with the contamination problem mentioned earlier, the PKASO board has the special Mosaic mode. Instead of being printed on top of each other, different color dots are mixed by being placed very close to each other. To produce an area of a given color mixture, the area is filled with dots in the proper proportion to achieve the desired shade. The Mosaic mode uses exactly the number of dots needed to fill the space, and no two dots are ever printed on top of one another.

Another advantage of the Mosaic mode is that fewer dots are printed and therefore less ink is used. This can as much as double ribbon life (no small matter considering the price of the four-color ribbon). Although the Mosaic mode produces pictures with a more pronounced dot pattern, it is not objectionable when viewed from a distance. In fact, if you look closely enough at any printed picture or at a color television screen you'll see the same thing.

Another feature is the reduced color option. This is useful when printing graphs and charts. When this option is selected, any isolated dots will be printed in black instead of color. This can also be helpful in overcoming some of the shortcomings of the Apple's color video generation scheme.

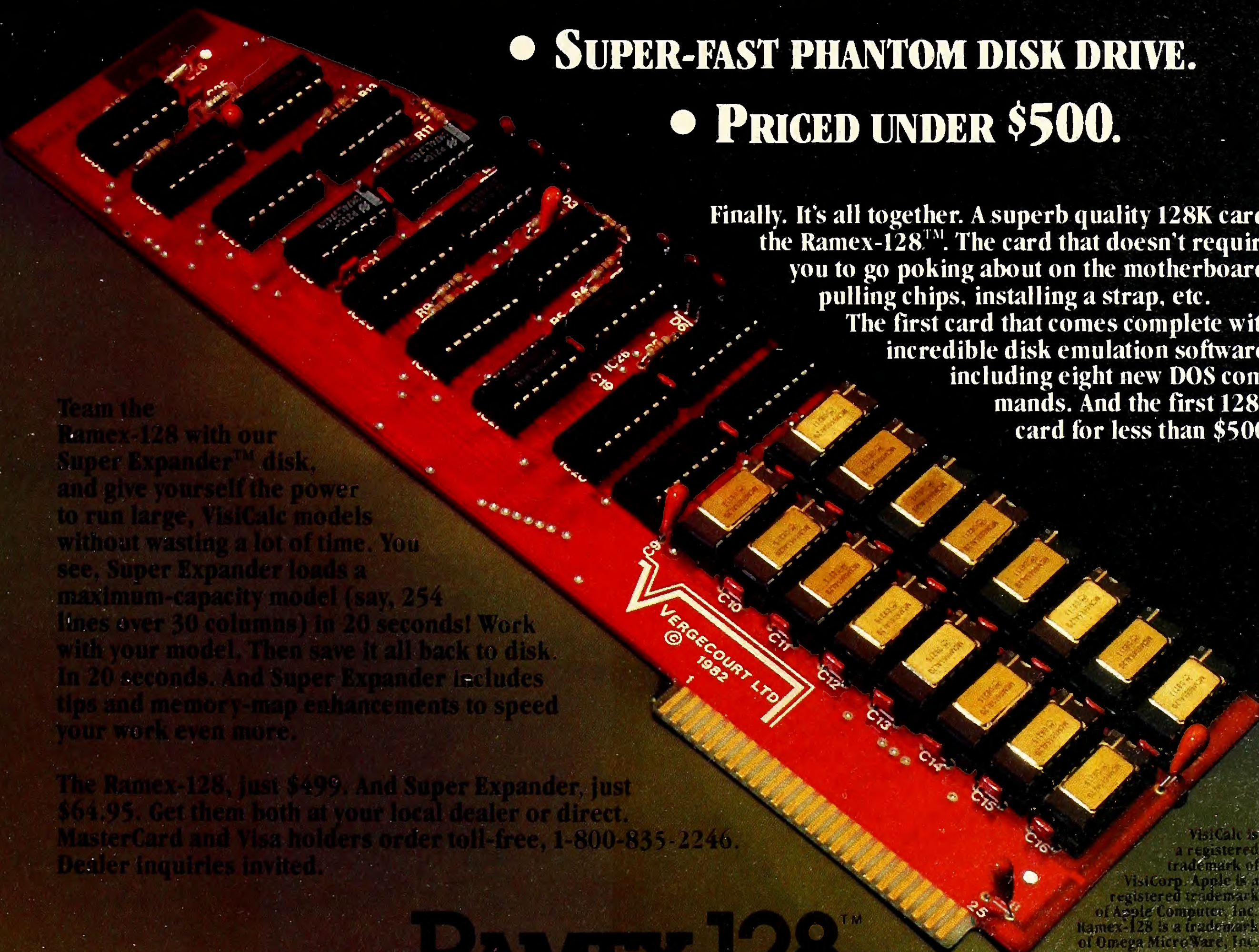
As if all this were not enough, the PKASO ID12 has one more major function—the Super-Res Color Graphics System. This system aids in the generation of images that are much too complex even to be displayed on the Apple screen. It allows full access to the color graphics capabilities of the Prism printer—namely, horizontal and vertical resolution of eighty-four dots per inch. The Super-Res system uses a 4K-byte buffer in memory to construct one line of the printed output (that is, a 7-dot high by 1,024-dot wide by four-color deep buffer). This buffer is first set up with the proper data (using plot and clear commands) for each of the four colors. Then the buffer is printed out, in four passes, to create one line of the finished picture. The buffer is then repeatedly set up and printed until the entire image has been created.

A software driver routine available on disk will interface the ID12 with the Apple III's SOS.

**Skyman 2001 Printer Interface.** The Skyman interface from Nutek International is a little different from the other interface boards. To begin with, it must be installed in slot 1; it won't work in any other slot! Although this is quite an uncommon requirement for this type of board, it really makes little difference since almost all Apples have their printer interface installed in slot 1. Another rarity is the use of several dip switches to select what type of printer is being used. Currently, graphics dumps

# Now here for your Apple®:
# A card that blows all the others away!

- **Loads or saves a full VisiCalc® model in 20 seconds. (Not 16 minutes like the others)**
- **136K for a VisiCalc model.**
- **Super-fast phantom disk drive.**
- **Priced under $500.**




Finally. It's all together. A superb quality 128K card, the Ramex-128™. The card that doesn't require you to go poking about on the motherboard, pulling chips, installing a strap, etc. The first card that comes complete with incredible disk emulation software, including eight new DOS commands. And the first 128K card for less than $500.

Team the Ramex-128 with our Super Expander™ disk, and give yourself the power to run large, VisiCalc models without wasting a lot of time. You see, Super Expander loads a maximum-capacity model (say, 254 lines over 30 columns) in 20 seconds! Work with your model. Then save it all back to disk. In 20 seconds. And Super Expander includes tips and memory-map enhancements to speed your work even more.

The Ramex-128, just $499. And Super Expander, just $64.95. Get them both at your local dealer or direct. MasterCard and Visa holders order toll-free, 1-800-835-2246. Dealer inquiries invited.

VisiCalc is a registered trademark of VisiCorp. Apple is a registered trademark of Apple Computer, Inc. Ramex-128 is a trademark of Omega MicroWare, Inc.

## Ramex-128™

**Under $500. What are you waiting for?**

Omega MicroWare, Inc.™

222 So. Riverside Plaza · Chicago, IL 60606 · 312-648-4844

are available only for Epson and Centronics printers.

With the exception of the control-I, N, and K commands, all the Skyman functions are accessed through a unique set of codes. Most of these begin with a special escape character that defaults to control-W. This escape character can only be changed by poking. Figure 6 shows the other available commands.

Although dumping graphics is certainly one of Skyman's major features, its rather cryptic command format leaves something to be desired. It also suffers from compatibility problems with graphics software. On the other hand, Skyman offers one more graphics feature. This feature is made possible by Skyman's graphics logical register, which allows logical *and*, *or*, and *exclusive-or* operations to be performed between the two graphics screens. One example of how this might be used would be the creation of a composite picture from two separate images. The background could be loaded into one hi-res screen while the foreground image was placed in the other. The *and* operation would print the composite image of the foreground and background (for those purists among you: sorry, hidden lines don't count!). A special overlay function is also provided to enhance this effect; this places a two-dot wide border around the foreground image. Other functions of the graphics logical register are to select normal versus inverted images and to change colors into a gray scale representation.

Although this board works with Pascal and CP/M, it has a few quirks. For example, it must be initialized from Basic before booting either operating system. The Skyman 2001 Parallel Printer Interface sells for $169 including cable and demonstration disk. While it has some impressive capabilities, its several limitations should be taken into account by those who are considering it.

**MPC AP-80 and AP-SIO.** MPC Peripherals Corporation offers two low-cost interface boards for the Apple—the AP-80 parallel and AP-SIO serial interfaces. The AP-80 is almost a clone of the Apple parallel printer interface. It has the same jumper configuration block to set strobe and acknowledge polarities and offers a limited number of commands due to its small 256 byte PROM. It's not fancy; it offers no graphics or text formatting features. But then it doesn't present any compatibility problems either. The best part is the price—$89.50, including cable.

| | |
|---|---|
| Control-W P | Enables hex mode. Treats all subsequent characters as hex data and sends the data directly to the printer. A period terminates this mode. For example, *print "control-W P 41"* prints the letter A on the printer. |
| Control-W V | Enables decimal mode. Similar to the hex mode but uses decimal values. |
| Control-W V7E | Enables elongated print mode. |
| Control-W VE | Disables elongated print mode. |
| Control-W VnL | Sets left margin to column n. |
| Control-W VnG | Sets up graphics logical register. |
| Control-W T | Initiates text dump. The current text screen is printed up to and including the line containing the cursor. |
| Control-W W | Text writing mode. Allows text to be typed on the screen and edited before printing. |
| Control-W S | Converts all text to lower case unless immediately preceded by an apostrophe (for example, *print "control-W S 'SOFTALK"* will print Softalk). |
| Control-W U | Converts all text to upper case. |
| Control-W Y | Enables dual mode. Allows two forty-column lines to be printed as one eighty-column line. |
| Control-W X | Disables dual mode. |
| Control-W Z | Prints graphics dump. This command is followed by several parameters: control-W Z n1n2:n3 n4 n5 n6 n7, where n1 and n2 select the type and size of the picture (normal, inverted, horizontal magnification [1-4], vertical magnification [1-3], and so on). The parameters n3 through n7 are optional and allow partial screen dumps. They specify the starting dot position, ending dot position, starting line number, ending line number, and left margin position, respectively. |

Figure 6. Skyman 2001 command format.

The AP-SIO provides the Apple with a serial communications port. This can be used to drive printers, plotters, modems, external terminals, and so on. Any device using the RS-232 standard can be connected to the AP-SIO for one-way or two-way communication with the Apple. This board features fourteen switch-selectable baud rates from 50 to 19,200 and provides for hardware handshaking. Another handy feature is the Serial Protocol Block. This is a small IC header that configures all of the connections to the DB-25 connector. Thus it is possible to swap input and output data lines or to select the proper handshaking signals without taking apart the interface cable.

There is a limited amount of firmware provided that allows the card to be accessed with PR# and IN# commands. The card will also function with Pascal and CP/M. It does not have a terminal mode the way the Apple's communications card does, so when the AP-SIO is used to connect a modem, a separate terminal program must be employed. There are four options that the firmware does support. These are lower to upper case translation, delay after carriage return, line feed after carriage return, and Apple video echo. Each of these options defaults to a value selected by dip switches but can be altered from software. The price of the AP-SIO is $129.50.

**Intra Computer PSIO.** The Programmable Serial In/Out board from Intra Computer provides both RS-232 and current loop operation. One possible use for this card would be for connection to the Telex/TWX network. The design of the board allows direct connection to the network in most cases; some installations may still require a Telex Line Adapter, however. Other features of the PSIO's current loop operation are active or passive modes, nonpolarized connections, and automatic dialing on the Telex/TWX networks. Of course, either the EIA (RS-232) or current loop interface could be used to connect a printer or other serial device.

On-board firmware connects the board from Basic, Pascal, CP/M, and so on, but no other commands are offered. Data format, baud rate, line width, and auto line feeds can be set by means of pokes or from machine language. One unique feature provided by the firmware is automatic baud rate selection. This allows the PSIO automatically to detect the speed of the incoming data and to adjust itself for the proper rate. Interrupt-driven operation is also possible with the appropriate machine language programming.

The documentation provided with the PSIO is fairly comprehensive and comes in a three-ring binder. A complete hardware description and ROM listing are included. One of the sample programs provided shows how the Apple can emulate a Baudot Teletype machine.

**Practical Peripherals Serial Microbuffer II.** The Microbuffer II was described fully in the March column. At that time, Practical Peripherals had only a parallel interface card for the Apple. The Serial Microbuffer is designed to offer the same features (up to 32K bytes of buffering and extensive text formatting commands) to those people who own serial printers. Graphics dumps are not available, however.

Note also that the Serial Microbuffer is a printer interface board only; it is not meant for duplex operation with devices such as modems. For this reason, the command structure is identical to the parallel Microbuffer. These boards support numerous text formatting commands, most of which were described in the earlier article. Since our initial reviews, the Microbuffer firmware has undergone several revisions, including the addition of some new commands. Figure 7 is *not* a complete description of all the Microbuffer commands; it is only a summary of the new ones.

Another change in this board is the default lead-in character used in Pascal and CP/M. This has been changed from control-Y to control-Q. The new character makes it possible to send commands directly to the card from the keyboard. This is accomplished by using the Filer Transfer command and specifying CONSOLE: to PRINTER:. The Serial Microbuffer II sells for the same price as the parallel versions—$259 with 16K buffer, $299 with 32K. A little expensive, but well worth it!

**Prometheus Graphitti Card.** This board is another Grappler lookalike. It uses the same command structure but includes a double hi-res dump (page 1 and page 2 printed contiguously). It also has two jumper plugs for selecting strobe and acknowledge polarities.

Another improvement has been made in the graphics dump routine that allows pictures to be printed in approximately the same aspect ratio as they appear on the Apple's screen. Most other graphics cards print a dot-for-dot representation of the hi-res screen. The space between vertical dots on most printers is not the same as the space between horizontal dots, however. For example, an Epson prints sixty dots per inch horizontally and seventy-two dots per inch vertically. Thus when a graphics dump is made of a circle on the Apple screen, it usually comes out looking like an ellipse. The Graphitti firmware has a special routine that interpolates extra points throughout the picture to compensate for these discrepancies. The result is an image on paper that has the same proportions as the original picture. Another fine point of the Graphitti card is its price—only $125.

**Prometheus Versacard.** Combine a parallel and serial interface board with a clock and BSR interface and you've got the makings of the

| | |
|---|---|
| Control-I D | Sets all text formatting parameters to their default values. This is used to clear the Maintain mode. |
| Control-I E | Sends an escape character to the printer. Useful for setting up printer modes. |
| Control-I F | Formats Basic program listings. This command breaks up multiple statement program lines so that each statement is printed on a new line. |
| Control-I M | Enables Maintain mode. In this mode, all print-formatting parameters are left intact when leaving and reentering the card. |
| Control-I T | Sets the Transparent mode. This command turns off all text formatting and command recognition by the card. This is useful when using a program such as a graphics dump that sends characters through the interface that might be falsely interpreted as commands to the interface. |

Figure 7. Summary of new commands in updated Microbuffer firmware.

Versacard. Like other multifunction cards, this board combines four functions onto one board using phantom slot addressing. This means that while the board may physically be in slot 2, the parallel port may be assigned to appear as if it were in slot 1. Before getting the idea that this will free some extra slots in your Apple, note that you cannot place another board into a phantomed slot (Z-80 type boards excepted). The serial port always appears in the physical Versacard slot and the clock and BSR devices always share the same slot.

The parallel port can be configured for either polarity of strobe and acknowledge signals. Aside from the control-I n N command to set line length, all of the other commands use a nonstandard format. These include enable/disable line feed, turn on/off video display, automatic paging, and high-order bit control. The Versacard now also performs graphics dumps in a similar fashion to the way the Graphitti card does.

The serial port handles all popular baud rates, data formats, and so on and has hardware handshaking. The firmware supports the same commands as for the parallel port. A terminal mode is also available for using the Versacard with a modem. In the terminal mode, you can toggle the printer on and off with a control-P, select half or full duplex with control-D, send a software break via control-B, or exit the terminal mode with a control-R.

The clock/calendar is set up to emulate a Thunderclock Plus. Onboard batteries keep the clock running when the computer is off. Seven different formats for the clock data are available depending on the exact command used to read the clock. For example, in response to these Applesoft commands:

```
IN#n
INPUT "%"; T$
```

the variable T$ will be assigned a string in the form of:

```
SAT FEB 09 10:23:55 PM.
```

The clock can also generate interrupts at one of three frequencies: 1,024 per second, 1 per second, or 1 per minute. The interrupt rate is selected by a small jumper on the board.

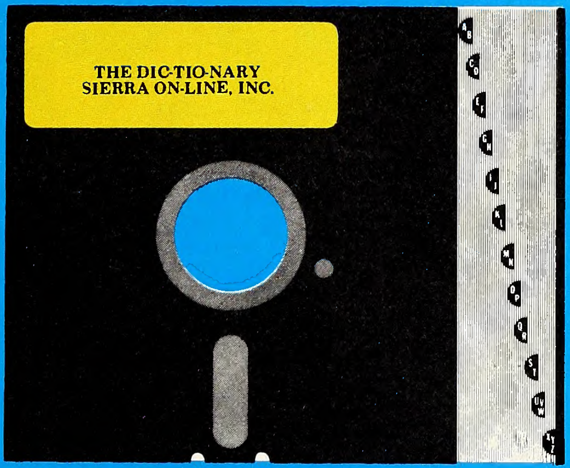
The BSR X-10 wireless AC control interface is also designed to emulate the Thunderclock Plus. An optional ultrasonic transducer must be connected to the Versacard and placed near a BSR command console. The Apple can then control those appliances that are connected to the BSR system by sending ultrasonic signals to the command console. Up to sixteen devices can be controlled including variable brightness on the lamp modules. A time and event scheduler program is now available to set up an automated system for controlling these devices.

The Versacard packs a lot of power onto one peripheral card. It suffers from some of the same drawbacks as other multifunction cards (see the May 1982 Hardtalk) but is still a bargain at $199.

**Other Changes.** Some of the other boards have been improving also. For example, the Grappler board we looked at in March has since been renamed the Grappler+. This board now has graphics routines for all of the printers on one EPROM. Thus the board can be configured for any printer by moving a few dip switches instead of removing and swapping EPROM chips. It also adds dual hi-res graphics with a side-by-side print-out of both screens.

It seems safe to say that this is not the last word on printer interfaces. As newer and better printers are announced, along with continually dropping prices, more and more Apple owners will be buying printers. This creates a great demand for interface boards ranging from the simplest, bare-bones parallel port to full-featured graphics boards with buffering. You should have little difficulty finding a board to suit your needs and budget. ⬛

*Interactive Structures, 146 Montgomery Avenue, Bala-Cynwyd, PA 19004; (215) 667-1713. Intra Computer, 101 West 31st Street, New York, NY 10001; (212) 947-5533. MPC Peripherals, 9424 Chesapeake Drive, San Diego, CA 92123; (714) 278-0630. Nutek International, 3020 East Camelback Road, Suite 200, Phoenix, AZ 85016; (602) 956-1123. Orange Micro, 3150 East La Palma, Anaheim, CA 92806; (714) 630-3620. Practical Peripherals, 31245 La Baya Drive, Westlake Village, CA 91362; (213) 991-8200. Prometheus Products, 45277 Fremont Boulevard, Fremont, CA 94538; (415) 490-2370.*




# Dić~tio~nāry

## Saves you from creative speling

No matter how good your Apple's word processor, the user is human. So, the odds are that spelling mistakes will occur in almost every letter, document, or proposal. Many errors will be caused by those who never won a spelling-bee and are addicted to 'creative' spelling...while others will be due to keystroke 'typos'. Either way, spelling errors can be embarrassing to you — to your image — to your business.

DIC-TIO-NARY **solves** the spelling problem for you, and does it at computer speed, easily and almost effortlessly. How easily, how effortlessly? Well, with DIC-TIO-NARY you have the option to correct a word immediately . . . right at the time you're doing the spelling check. No waiting necessary. And if, by chance, you don't know the correct spelling of the word, you can go into the wordbook right then and there and have it displayed for you! (Ask for these features on competitive programs, no matter their price, and see how fast they change the subject.) We think you'll find that for helpfulness our DIC-TIO-NARY is on par with having your favorite English teacher or even Mr. Webster at your elbow!

### words, words, words

The DIC-TIO-NARY starts you off with more words than you'll probably ever need: 25,000 of the words used most often. **Plus** it lets you add 3000 words of your own choosing, so you can customize it to your specific needs. (By making back-up "wordbooks" you could have many additional lists of 3000 words!)

### how it works

In simplest form, the DIC-TIO-NARY races through your document at a 2+ page-per-minute speed. It looks at **every** word in your text. At the same time it compares each word with those in its 28,000 wordbook . . . checking for any misspelling. When it catches a goof, you get to see the word as it was used in your text, **on the screen**. The DIC-TIO-NARY then asks how you want that particular word handled.

### choices, choices

The DIC-TIO-NARY will display for you any word not listed in its wordbook and any misspelled word. It then offers you a choice of word-editing options. For example: You may decide to ignore the word. You may have it marked in your text and/or on the printout, for later identification, correction or replacement. You may have the word automatically added to your wordbook. Or, you may immediately correct the spelling (using the wordbook for verification, if desired) . . . and all occurrences of that word will automatically be corrected throughout your text, keeping the **original** capitalizations or lower case letters!

### extras, extras

For ease of use, for efficiency and true proofreading effectiveness the DIC-TIO-NARY provides you with many other useful goodies. You may add or delete individual words from the wordbook...or do the same with an entire list of words. You may have any portion of the wordbook listed, and may search it even while processing documents. When you add it up, you'll find you can't get our total combination of features in any other spelling program at any price.

All in all our DIC-TIO-NARY may well be the most valuable and useful program since the little red schoolhouse disappeared. Yet, you can enjoy all its benefits for only $99.95. If you buy our special Screen Writer Professional, you get the $129.95 SCREEN WRITER II word processor described on the left, plus The DIC-TIO-NARY, at the package price of $199.95. Be creative! Save $29.95!

DIC-TIO-NARY works with ScreenWriter II, Super-Scribe II, Apple Pie and Apple Writer. Requires 48K Apple II or II+ with DOS 3.3. Available at your local computer store or directly from . . .

Visa, Master Charge, C.O.D., or Checks accepted. Add $1.00 for direct orders.

Sierra ON-LINE INC
36575 MUDGE RANCH ROAD COARSEGOLD CA 93614
TLX 910 362 5025 209 683 6858


LASER
BOUNCE
TM


FINAL
CONFLICT

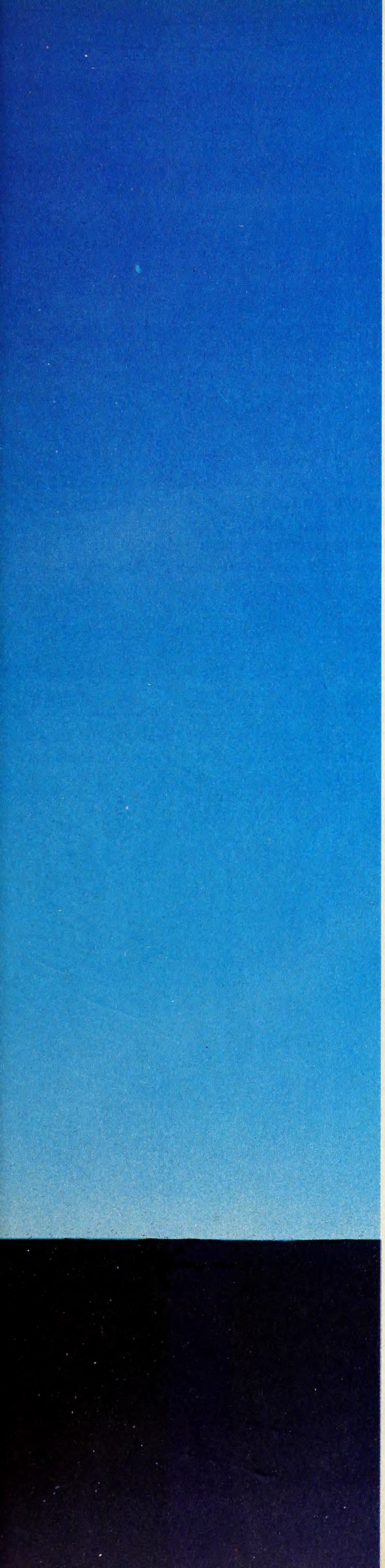


Illustration by Kevin McKeon

S·T·O·R·Y·T·A·L·K F·I·C·T·I·O·N

# The Man Who Changed His Mind

At 12:45 p.m. Harlan's head fell off, for the last time. He had lost his head before; it had teetered and tottered, it had rolled away; but it had always returned. Snapped back into place. Home for lunch, as it were.

Not this time. It was gone for good. Split, missing, inescapably off; to be forgotten. After all, it did have a mind of its own. The only problem was that it was Harlan's mind too, and he was going to miss it.

He shrugged his three shoulders. Not really three shoulders, but that's what it looked like he had. Three shoulders to cry on if his head ever came back.

His nerves began to panic. It was over, the deed was done. Scared, headless, alone, what had he done? He had been of two minds lately; had the conflict been too much? At least it had been a clean break, he thought.

He thought? Wait a minute, what was he thinking with? He thought again. He was still on his feet, his pulse rushing as thoughts poured from somewhere into the nowhere of his mind. He couldn't see, hear, or speak, but he was thinking. How could that be, he thought. (A dizzying, wonderful feeling, to be able to think on one's feet without benefit of a brain.)

A phantom head. That's what it had to be. Of course. The last scrambled dreams, desires, and con-

BY MICHAEL FERRIS

volutions of the body in the few seconds after the final plug is pulled. A few sputtering, smokeless moments of brilliant clarity courtesy of the central nervous system.

People around him were pointing, he realized. Horse laughs, blank stares, incredulous gasping. They were seeing it and it was unbelievable, despite the first-hand proof. It could really happen. Nobody had any idea what was going on inside the man, but from the outside nothing had really changed. He just didn't have a head. He must have had one just a minute ago, but nobody could say what it looked like, so what was the difference?

Lots of people are loose in the world without a brain in their bodies. Harlan took his mind off the crowd and walked away.

He thought he might be using his precious last moments, and he didn't want to waste them on idle speculation. Ease up, think quiet thoughts, keep the juice running low. He could actually live like this, he thought. For how long, he couldn't guess.

A three-shoulder goon, that's what he felt like. A hulking, headless, hamfisted guy, the type who grabbed at life—no pleases or thank-yous—and never thought for a moment there was any other way. He was one of them now. With only his fingers to do the talking.

He had upset his delicate mental process for the last time and lost his head over it. He slowly scrolled over the last few years: all the movies, magazines, television, and computers he had slowly given his mind over to. He had forfeited his imagination to technology and he had paid the price. Detachment. Of the final, ultimate kind.

He smiled.

He smiled? With what? Well, he thought he smiled. A thin, ironic smile snaked across what he remembered of his lips. Robbed, that's how he felt. Robbed of himself, by himself, for the good of himself. In the name of progress. Even for the good of mankind. A funny, flimsy line of reasoning. Fitting that it matched the negligent style of the crime.

A deep sigh somehow filled his lungs with air. Nourishment flooded his veins; he wasn't hungry anymore. A good sign. He was learning to take care of himself.

Beep.

Bolstered by that small fact, he chanced a speculation. There was a time when it was natural for him to ponder deep meanings, weigh alternatives, consider moves, articulate all his choices, then make them smoothly. Like everybody, he cried through the rough spots and laughed when things went easy. Without a clear head and with four senses less to connect him to the outside world, could he ever hope to do all this again?

Getting his hopes up, that's what he was doing. A noble and fine thing to do, even possible without really thinking. Perfect for a headless person.

Beep.

There was plenty to hope about.

Somewhere a computer booted a program with Harlan's name on it. All by itself.

He hoped he would live a long life and never go hungry. Whirrr. He hoped for friends and lovers to share his dreams with. Clackity clack. He hoped for inspiration to fuel his ambitions. Hummm.

Up and running, the program sealed itself in its disk drive and locked its power switch on. This was one disk that was never going to crash.

Beep. He heard it again. Beep.

Harlan felt a tingling, then a chill, as he realized what was happening. A network of personal computers was booting inside his phantom mind. Giving him hope. His new head was suddenly a collision of charts, bars, mazes, treasure rooms, and dancing graphics. Endless menus to choose from, ever-expanding subroutines, all the ROM space he could ever hope to use. Words in process, access to files, calls via modem. Powerful stuff. All of it—crunch! rumble! collide!—shooting straight through him.

He was connected to the invisible world of the microcomputer. He was one of them somehow.

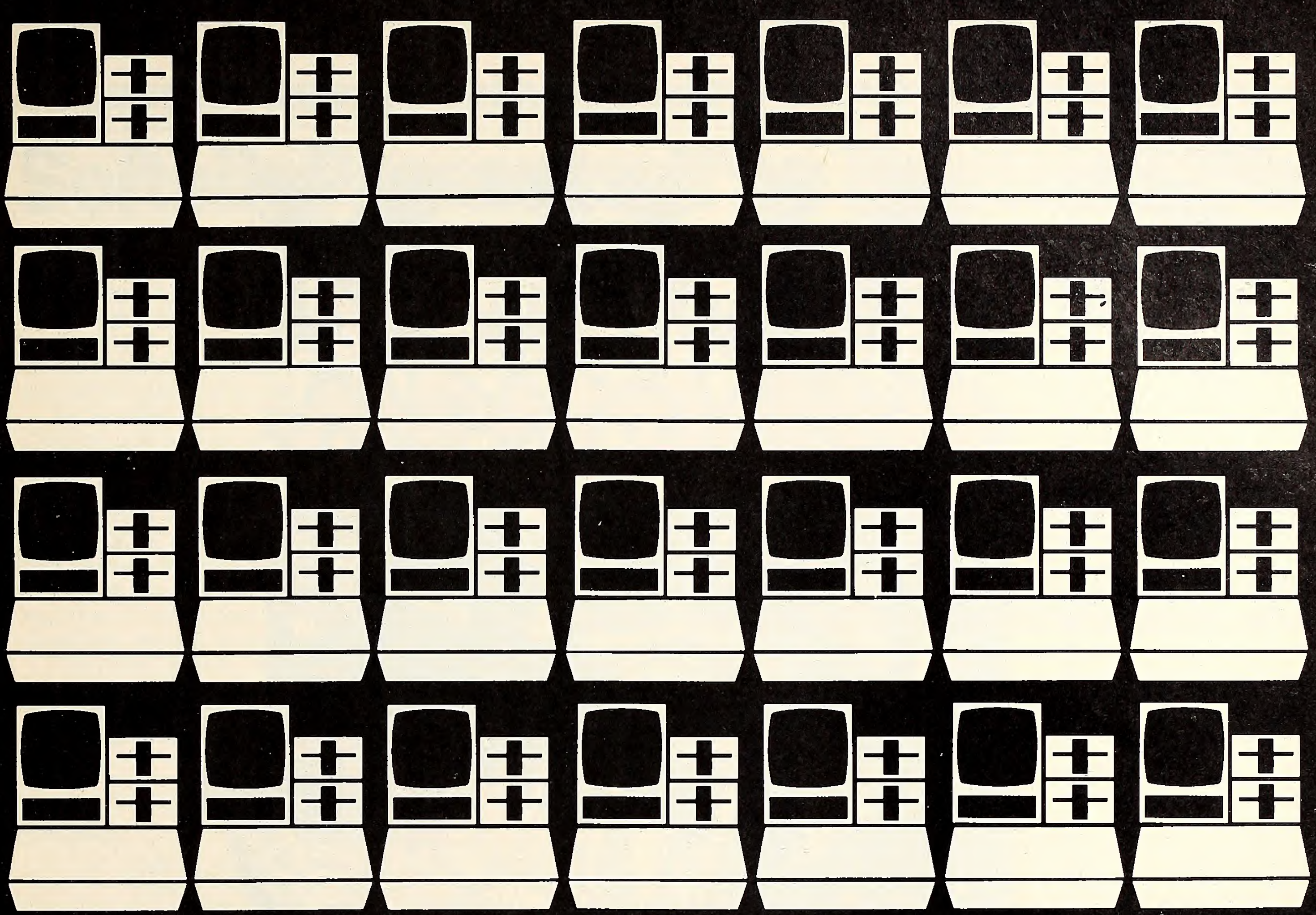
Harlan was overjoyed. It was more than a headless person could have hoped for.

Beep.

Somewhere another computer booted. Whirrrr. Its cursor glowed with impish perversity. Clackity clack. A buffer dumped out some garbage across its monitor screen. Hummm. It looked like. . . .

Yes.

It looked like it was smiling.



NOW, also available in Atari format!!!


# You haven't lived until you've died in space.


And here's your chance.

Software author Peter Folkos has created Alien Ambush, a space age nightmare. It's a hi-res, full-color arcade game, fiendishly written to give those nasty aliens every advantage.

So if you have access to a 48K Apple* with DOS 3.3 or a 16K Atari 400/800** with a disk drive, and you're hot for some new thrills, Alien Ambush was written for you. But be warned: it just got a lot tougher to survive in space.

*Apple is a trademark of Apple Computer, Inc. **Atari, Atari 400 and Atari 800 are trademarks of Atari, Inc.
Distributed exclusively by Micro D, 17406, Mt. Cliffwood Circle, Fountain Valley, Ca. 92708 (714) 540-4781

# Softalk Presents The Bestsellers

September was the best of times for the folks from VisiCorp, but October brought a bag of mixed blessings. *VisiCalc*, the company's bellwether program, regained the lead as the bestselling piece of software in the two largest software markets in September. And a special promotion started October off strongly for the industry leader. But the announcement that one of their former employees was in strong competition with them didn't augur well for the future.

In the Apple market, *VisiCalc* regained the lead from Broderbund's *Choplifter* after running second for three months. The lead is decisive, but it's not an indication that *Choplifter*'s about ready to fold up its tent and steal away into the night. Dan Gorlin's hostage scenario was so strong that the *Softalk* convention of applying an index rating of approximately one hundred was applied to the third rated program. *Choplifter* was far ahead of third-place *Personal Filing System*.

*VisiCalc* had also fallen temporarily on hard times in the IBM market, where it was displaced in August by *WordStar*. But the pioneer spreadsheet resurged in that market as well to regain the lead.

Then came October. VisiCorp unveiled a new marketing plan—buy any two *Visi* products and get one free. That's fueled sales of *VisiTrend/VisiPlot*, *VisiFile*, and *VisiSchedule* and all was looking reasonably rosy, even with the advent of Microsoft's *MultiPlan* into the Apple market as a spreadsheet competitor.

But along came Mitch Kapor, now president of Lotus Development, author of *VisiTrend/VisiPlot*, and formerly a VisiCorp product manager, with the announcement of *1-2-3* for the IBM. The product is a combination spreadsheet, text processor, and database that bids fair to give VisiCorp a run for its money. Everyone takes aim at the champs.

Alas for the Apple II market, it's not likely ever to get a chance at *1-2-3*. The program requires at least 128K of RAM and Lotus recommends 256K to achieve maximum performance. That clearly leaves the Apple III as a viable machine for the product, however.

September saw an increase in software sales, but most game publishers would be hard pressed to vouchsafe that fact. In a sea of essentially undifferentiated arcade and hi-res adventure games, few titles were able to rise above the average. The result was that the greater sales were spread more evenly over a larger number of titles.

An exception to that rule was Broderbund product, which consistently outperformed the rest. Not only did *Choplifter* have an iron grip on second, *The Arcade Machine* rose from a tie for twenty-seventh to eighth. *Serpentine* rose from twenty-sixth to a tie for twelfth. *Star Blazer* fell from eleventh to a tie for nineteenth, but the newest Broderbund entry, *Seafox*, jumped into twenty-seventh. That means that one-sixth of the Top Thirty came from one publisher and, interestingly enough, each of the programs has a different author, which indicates breadth as well as talent in the Broderbund stable.

Although new packages are entering the market on an almost daily basis, it's notable that there's not one new product on the Top Thirty other than *Seafox*. Other programs that are listed as new to the list this month actually are regaining the Top Thirty from prior months. These include *VisiFile*, *Super-Text 40-56-70*, and *Zoom Grafix*. It was a pretty ho-hum month at the top.

There was a minor surprise in the specialty Adventure 5 list. Sirius's

## Apple III

| This Month | Last Month | |
|---|---|---|
| 1. | 1. | **VisiCalc**, Software Arts/Dan Bricklin and Robert Frankston, VisiCorp |
| 2. | 2. | **Apple Writer III**, Paul Lutus, Apple Computer |
| 3. | 6. | **Apple III Business Graphics**, Apple Computer |
| 4. | 3. | **Personal Filing System**, John Page and D. D. Roberts, Software Publishing Corporation |
| 5. | 7. | **Access III**, Apple Computer |
| 6. | 10. | **VersaForm**, Joe Landau, Applied Software Technology |
| 7. | 4. | **PFS: Report**, John Page, Software Publishing Corporation |
| 8. | 5. | **Apple III Business Basic**, Apple Computer |
| 9. | — | **Word Juggler**, Tim Gill, Quark Engineering |
| | — | **VisiCalc—Advanced Version**, Software Arts/Dan Bricklin and Robert Frankston, VisiCorp |

## Word Processors 10

| This Month | Last Month | |
|---|---|---|
| 1. | 2. | **Apple Writer II**, Paul Lutus, Apple Computer |
| 2. | 1. | **Screen Writer II**, David Kidwell, Sierra On-Line |
| 3. | 4. | **Word Handler**, Leonard Elekman, Silicon Valley Systems |
| 4. | 7. | **Super-Text 40-56-70**, Ed Zaron, Muse |
| 5. | 6. | **Magic Window**, Bill Depew and Gary Shannon, Artsci |
| 6. | 3. | **WordStar**, MicroPro |
| 7. | 8. | **PIE Writer**, Softwest, Hayden |
| 8. | 5. | **Sensible Speller**, Sensible Software |
| 9. | 9. | **Dictionary**, Tom Cain, Sierra On-Line |
| 10. | 10. | **Executive Secretary**, John Risken, Sof/Sys |

*Kabul Spy* jumped back into contention while its running mate, *Escape from Rungistan*, continued as the leader. Infocom's three entries stayed on the list and the former undisputed adventure champ, Sierra On-Line, had to be content with a single contender. And that was *Hi-Res Adventure #2: The Wizard and the Princess*, an oldie but goodie.

There were no real shockers in the Strategy 5. *Rendezvous* closed on leader *Castle Wolfenstein*. Former entry *Hi-Res Computer Golf* regained the list, pushing *Sargon II* to fifth. Strategic Simulations's line of Rapid Fire games hovered just outside the limit.

The inclusion of *Ali Baba and the Forty Thieves* in the Fantasy 5 list rectifies a *Softalk* error. Had it been included in this category in the prior two months, it would have been third in July and fourth in August. As it is, the Quality Software entry scores fourth in a category still dominated by *Wizardry* and *Knight of Diamonds*.

*Home Accountant* continues to dominate the Home 10 listing. Other than personal finance packages and communications software, this list tends to reflect seasonal requirements; for example, tax packages fare well during the early part of the year. Now is no different. With high school students going back to school and preparing for their college boards, Micro Lab's *SAT English I* package came from nowhere to edge into the list.

The other change of note was a resurgence of strength by Microsoft's *Typing Tutor*. For a time, it looked like *MasterType* would blow *Typing Tutor* out of the Apple market, but the pioneer typing teacher is making a comeback.

## Strategy 5

| This Month | Last Month | |
|---|---|---|
| 1. | 1. | **Castle Wolfenstein**, Silas Warner, Muse |
| 2. | 2. | **Rendezvous**, Wes Huntress, Edu-Ware Services |
| 3. | 3. | **Flight Simulator**, Bruce Artwick, SubLogic |
| 4. | — | **Hi-Res Computer Golf**, Stuart Aronoff, Avant-Garde |
| 5. | 4. | **Sargon II**, Dan and Kathe Spracklen, Hayden |

## Adventure 5

| This Month | Last Month | |
|---|---|---|
| 1. | 1. | **Escape from Rungistan**, Bob Blauschild, Sirius Software |
| 2. | 2. | **Hi-Res Adventure #2: The Wizard and the Princess**, Ken and Roberta Williams, Sierra On-Line |
| | 5. | **Zork I**, Infocom |
| | — | **Kabul Spy**, Tim Wilson, Sirius Software |
| 5. | 3. | **Deadline**, Infocom |
| | 4. | **Zork II**, Infocom |

## Fantasy 5

| This Month | Last Month | |
|---|---|---|
| 1. | 1. | **Wizardry**, Andrew Greenberg and Robert Woodhead, Sir-tech |
| 2. | 2. | **Knight of Diamonds**, Andrew Greenberg and Robert Woodhead, Sir-tech |
| 3. | 3. | **Ultima**, Lord British, California Pacific |
| 4. | — | **Ali Baba and the Forty Thieves**, Stuart Smith, Quality Software |
| 5. | 4. | **Apventure to Atlantis**, Bob Clardy, Synergistic Software |

# COMPUTER GRAPHICS


for the Artist

## Special Effects

by David Lubar and Mark Pelczarski

Paint on your computer in over 100 colors with 96 different brushes! Magnify mode for precision touch-ups; mirror image; color reverses; move any part of picture anywhere; packing routine for increased disk storage of any standard graphics screen. Can be used by anyone. Paddle/joystick $39.95. Apple Graphics Tablet $69.95.

for the Designer

## THE COMPLETE GRAPHICS SYSTEM II

by Mark Pelczarski

Everything needed for computer-aided design. Easily draw and design in two or three dimensions, mix text and graphics freely, define shapes and create typefaces with unique character generator. Great for presentations, videotape displays, storing designs on disks, graphics for other programs, or just for experimentation. Paddle/joystick $69.95, Apple Graphics Tablet $119.95. 50 additional fonts, $19.95.

for the Programmer

## The Graphics Magician

by Mark Pelczarski, David Lubar, and Chris Jochumson

Add fast, smooth animation and hundreds of pictures to your programs. Create animated shapes and draw paths; have up to 32 independently controlled figures on screen. Use a special palette of over 100 colors to create pictures and objects in highly compressed format. Easy to use, and includes a special programming tutorial. Has saved months in development time for several games already on the market. Paddle/joystick $59.95, Apple Graphics Tablet $69.95.

**penguin software** 830 4th Avenue, Geneva, Illinois 60134 (312) 232-1984

All Penguin applications products are now on unprotected disks for your convenience. Apple II is a trademark of Apple Computer, Inc.

There was a significant shakeup in positions in the Hobby 10 list, although Beagle Bros's *DOS Boss* retained the lead. *Zoom Grafix* jumped to second and *DOS Tool Kit* jumped to third.

For the past two months, Apple Computer's aggressive marketing of a Family Plan package that included an *Apple Writer II* with the purchase of a 48K machine has been depressing the market for word proc-

# Home 10

| This Month | Last Month | |
|---|---|---|
| 1. | 1. | **Home Accountant,** Bob Schoenburg, Larry Grodin, and Steve Pollack, Continental Software |
| 2. | 2. | **MasterType,** Bruce Zweig, Lightning Software |
| 3. | 3. | **Transend,** Tim Dygert and Bob Kniskern, SSM |
| 4. | 7. | **Typing Tutor,** Image Producers, Microsoft |
| 5. | 6. | **Data Capture 4.0,** David Hughes and George McClelland, Southeastern Software |
| 6. | 9. | **Apple Logo,** Apple Computer |
| 7. | 8. | **VisiTerm,** Tom Keith, VisiCorp |
| 8. | 4. | **Personal Finance Manager,** Jeffrey Gold, Apple Computer |
| 9. | — | **SAT English I,** Eileen Shapiro, Micro Lab |
| 10. | 5. | **ASCII Express,** Bill Blue, Southwestern Data Systems |

# Hobby 10

| This Month | Last Month | |
|---|---|---|
| 1. | 1. | **DOS Boss,** Bert Kersey and Jack Cassidy, Beagle Bros |
| 2. | 7. | **Zoom Grafix,** Dav Holle, Phoenix Software |
| 3. | 6. | **DOS Tool Kit,** Apple Computer |
| 4. | — | **Locksmith 4.1,** Omega Microware |
| 5. | 3. | **Bag of Tricks,** Don Worth and Pieter Lechner, Quality Software |
| 6. | 9. | **Apple Mechanic,** Bert Kersey, Beagle Bros |
| 7. | 5. | **Utility City,** Bert Kersey, Beagle Bros |
| 8. | 4. | **Graphics Magician,** Mark Pelczarski, Penguin Software |
| 9. | — | **Watson,** Bill Sefton, Omega Microware |
| | — | **Alpha Plot,** Bert Kersey and Jack Cassidy, Beagle Bros |

# Business 10

| This Month | Last Month | |
|---|---|---|
| 1. | 1. | **VisiCalc,** Software Arts/Dan Bricklin and Robert Frankston, VisiCorp |
| 2. | 2. | **Personal Filing System,** John Page and D. D. Roberts, Software Publishing Corporation |
| 3. | 3. | **DB Master,** Alpine Software/Stanley Crane and Jerry Macon; and Barney Stone, Stoneware |
| 4. | 7. | **VisiTrend/VisiPlot,** Micro Finance Systems/Mitch Kapor, VisiCorp |
| | 8. | **VisiFile,** Creative Computer Applications/Colin Jameson and Ben Herman, VisiCorp |
| 6. | 4. | **PFS: Report,** John Page, Software Publishing Corporation |
| 7. | 6. | **BPI General Ledger,** John Moss and Ken Debower, Apple Computer |
| 8. | 5. | **PFS: Graph,** Bessie Chin and Stephen Hill, Software Publishing Corporation |
| | — | **BPI Accounts Receivable,** John Moss and Ken Debower, Apple Computer |
| 10. | — | **VersaForm,** Joe Landau, Applied Software Technology |

# Softalk Presents The Bestsellers

essors. It seems now that the emphasis on word processing has rebounded to the benefit of everyone.

Even discounting *Apple Writer II* giveaways, the package took over the lead from *Screen Writer II*. But *Word Handler, Super-Text 40-56-70,* and *PIE Writer* all appeared stronger in September. Chief losers appeared to be *Screen Writer II* and *WordStar.*

There was a tremendous increase in the number of business packages being sold by dealers, belying the theory that the Apple cannot compete with IBM or other personal computers for the business market. The *BPI Accounts Receivable* package regained the list, while *VersaForm* attained the list for the first time.

*BPI* appears to have a clear lead as the accounting software of choice, but *Great Plains*, *Accounting Plus*, and newcomer *State-of-the-Art* appear neck-and-neck for runner-up position.

*VersaForm* also performed well in the Apple III market, where it jumped from tenth to sixth. *Apple III Business Graphics* made a three-position jump from sixth to third.

The big news brings us full circle to where we started—VisiCorp. *VisiCalc—Advanced Version* jumped into tenth place in its first full month of distribution, indicating that Apple III users are intent on getting the most out of their powerful desktop tool.

That's good news for VisiCorp, even if *1-2-3* is not.

Apple-franchised retail stores representing approximately 6.7 percent of the sales of Apples and Apple-related products volunteered to participate in the poll.

Respondents were contacted early in October to ascertain their sales leaders for the month of September.

The only criterion for inclusion on the list was number of sales made—such other criteria as quality of product, profitability to the computer retailer, and personal preference of the individual respondents were not considered.

Respondents in October represented every geographical area of the continental United States.

Results of the responses were tabulated using a formula that resulted in the index number to the left of the program name in the Top Thirty listing. The index number is an arbitrary measure of relative strength of the programs listed. Index numbers are correlative only for the month in which they are printed; readers cannot assume that an index rating of 50 in one month represents equivalent sales to an index number of 50 in another month.

Probability of statistical error is plus-or-minus 4.7 percent, which translates roughly into the theoretical possibility of a change of 3.94 points, plus or minus, in any index number.



## The Top Thirty

| This Month | Last Month | Index | |
|---|---|---|---|
| 1. | 2. | 171.82 | **VisiCalc,** Software Arts/Dan Bricklin and Robert Frankston, VisiCorp |
| 2. | 1. | 146.32 | **Choplifter,** Dan Gorlin, Broderbund Software |
| 3. | 3. | 99.15 | **Personal Filing System,** John Page and D. D. Roberts, Software Publishing Corporation |
| 4. | 6. | 67.86 | **Apple Writer II,** Paul Lutus, Apple Computer |
| 5. | 4. | 64.49 | **Wizardry,** Andrew Greenberg and Robert Woodhead, Sir-tech |
| 6. | 5. | 56.31 | **Screen Writer II,** David Kidwell, Sierra On-Line |
| 7. | 7. | 50.54 | **Home Accountant,** Bob Schoenburg, Larry Grodin, and Steve Pollack, Continental Software |
| 8. | 27. | 45.24 | **The Arcade Machine,** Chris Jochumson and Doug Carlston, Broderbund Software |
| 9. | 27. | 43.32 | **Word Handler,** Leonard Elekman, Silicon Valley Systems |
| 10. | 8. | 35.62 | **Snack Attack,** Dan Illowsky, DataMost |
| 11. | 10. | 34.65 | **Knight of Diamonds,** Andrew Greenberg and Robert Woodhead, Sir-tech |
| 12. | 9. | 28.88 | **DB Master,** Alpine Software/Stanley Crane and Jerry Macon; and Barney Stone, Stoneware |
| | 26. | 28.88 | **Serpentine,** David Snider, Broderbund Software |
| 14. | 25. | 25.51 | **VisiTrend/VisiPlot,** Micro Finance Systems/Mitch Kapor, VisiCorp |
| | — | 25.51 | **VisiFile,** Creative Computer Applications/Colin Jameson and Ben Herman, VisiCorp |
| 16. | 13. | 24.55 | **Cannonball Blitz,** Olaf Lubeck, Sierra On-Line |
| 17. | 15. | 24.07 | **PFS: Report,** John Page, Software Publishing Corporation |
| 18. | 20. | 23.58 | **BPI General Ledger,** John Moss and Ken Debower, Apple Computer |
| 19. | 11. | 23.10 | **Star Blazer,** Tony Suzuki, Broderbund Software |
| | — | 23.10 | **Super-Text 40-56-70,** Ed Zaron, Muse |
| | 12. | 23.10 | **MasterType,** Bruce Zweig, Lightning Software |
| 22. | — | 22.62 | **Magic Window,** Bill Depew and Gary Shannon, Artsci |
| 23. | 17. | 19.73 | **Transend,** Tim Dygert and Bob Kniskern, SSM |
| 24. | — | 19.25 | **BPI Accounts Receivable,** John Moss and Ken Debower, Apple Computer |
| | 19. | 19.25 | **PFS: Graph,** Bessie Chin and Stephen Hill, Software Publishing Corporation |
| 26. | — | 18.77 | **A2-PB1 Pinball: Night Mission,** Bruce Artwick, SubLogic |
| 27. | — | 18.29 | **Seafox,** Ed Hobbs, Broderbund Software |
| 28. | 16. | 17.33 | **Castle Wolfenstein,** Silas Warner, Muse |
| 29. | 27. | 16.85 | **DOS Boss,** Bert Kersey and Jack Cassidy, Beagle Bros |
| 30. | — | 16.36 | **Zoom Grafix,** Dav Holle, Phoenix Software |

# Sirius™ All Star Games

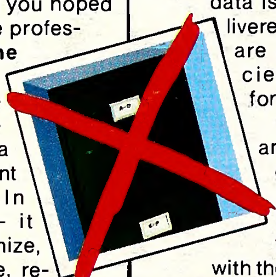

# SUPER FUN!

# The reason you bought a computer in the first place.

**The Agony...** You expected your new computer to perform miracles — to bring order out of chaos. You looked for it to organize and manage your business information. You looked forward to the end of errors, the end of frustration . . . and the saving of time, effort and money. Afterall, that's the reason you invested in a computer in the first place. Yet, there it sits. Nothing.

# The General Manager

**...and the ecstasy.**
Well, your computer *can* perform all the miracles you hoped for. It needs just one professional addition. **The General Manager.**

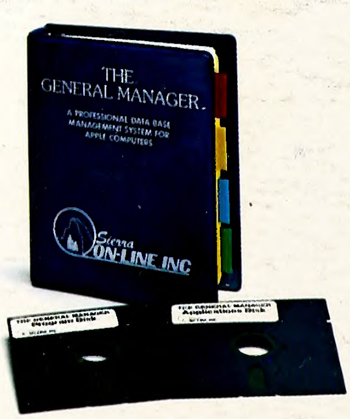


The General Manager is what the computer industry calls a data base management program (DBM). In everyday words — it allows you to organize, store, file, find, save, retreive, interrelate, control and print out all or selected parts of your information. The result: your information, or data, is managed totally, completely, automatically.

**Ordinary**
The ordinary DBM system expects your business to conform to its program design. So you must change your records, your forms, your way of having information cross-referenced, saved and . . . well, you almost end up with a different business! Certainly a more frustrating one.

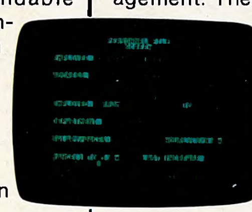
**Extraordinary**
The General Manager on the other hand is extraordinary in the DBM field, because it makes no such demands on you. Instead, it lets you make demands on it! The General Manager was designed so that your business routines can be kept as individualized as you want . . . so your data is managed and delivered in the ways which are most useful, efficient and effective for you.

It works so easily and so well because of its "hierarchial" structure. This sensible "family tree" type of design starts with the main subject, then branches out to related information. You enter data on "Blank Forms" which you may construct to your exact needs. The data may be updated, deleted or modified to your heart's content. To know The General Manager will be to love it!




**Power & Price**
Nothing near the price of the General Manager (by hundreds of dollars) gives you all the power, features and benefits it does! At $229.95, The General Manager is the absolute value in its field.

Consider this: it supports 1 to 4 floppy disk drives (even hard-disk systems). It includes utility programs which others charge hundreds extra for. Upper and lower case characters in the data base are provided without need for additional costly hardware. If someone goofs, the "error message" is displayed in understandable English. There is an on-screen "Help" function available any time. It creates Applesoft usable files for your program needs. And many especially useful printing commands are built-in for greater flexibility. When you consider all these advantages, and more, we think your business sense will agree, there's no contest at any price.



**The fantasy...**
Almost everyone claims user friendly documentation. The fact remains much of it is convoluted, complicated and defies understanding. You can't afford that! — for a program without excellent documentation is frustrating and basically useless.

**...and reality!**
At Sierra On-Line we've spent the time and the effort to create superb documentation. It is so good that you can have The General Manager up and running after the first two chapters! And after you're thoroughly at home with it, you can move on to the other chapters as you have need for the many additional functions and capabilities.

**The reason you bought...**
. . . a computer in the first place was, we know, twofold: for word processing (our Screen Writer program is the leader) . . . but mainly for information management. The General Manager is your powerful answer . . . the truly outstanding value in DBM's, bar none, at only $229.95.

For further information and ordering, see your computer dealer or contact us:

**(800)344-7448**

Sierra ON-LINE INC

SIERRA ON-LINE SYSTEMS, INC.
Business products division.
36575 Mudge Ranch Road
Coarsegold, CA 93614

*The General Manager, version 2.0 requires 48K Apple II or II+, 1 or more drives, DOS 3.3. Direct orders add $3.00 shipping/handling.