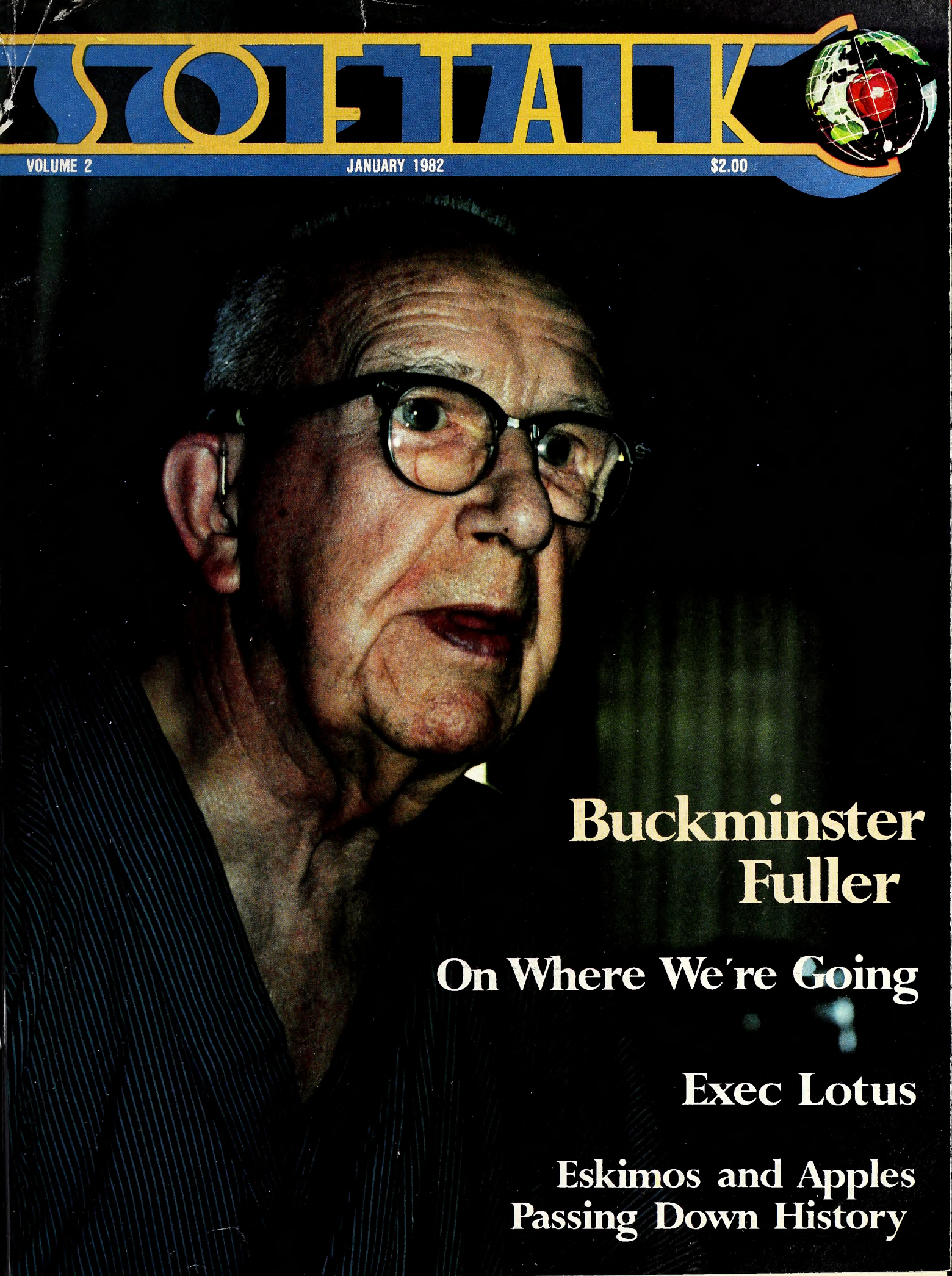
SOFTALK

VOLUME 2 JANUARY 1982 $2.00



Buckminster Fuller

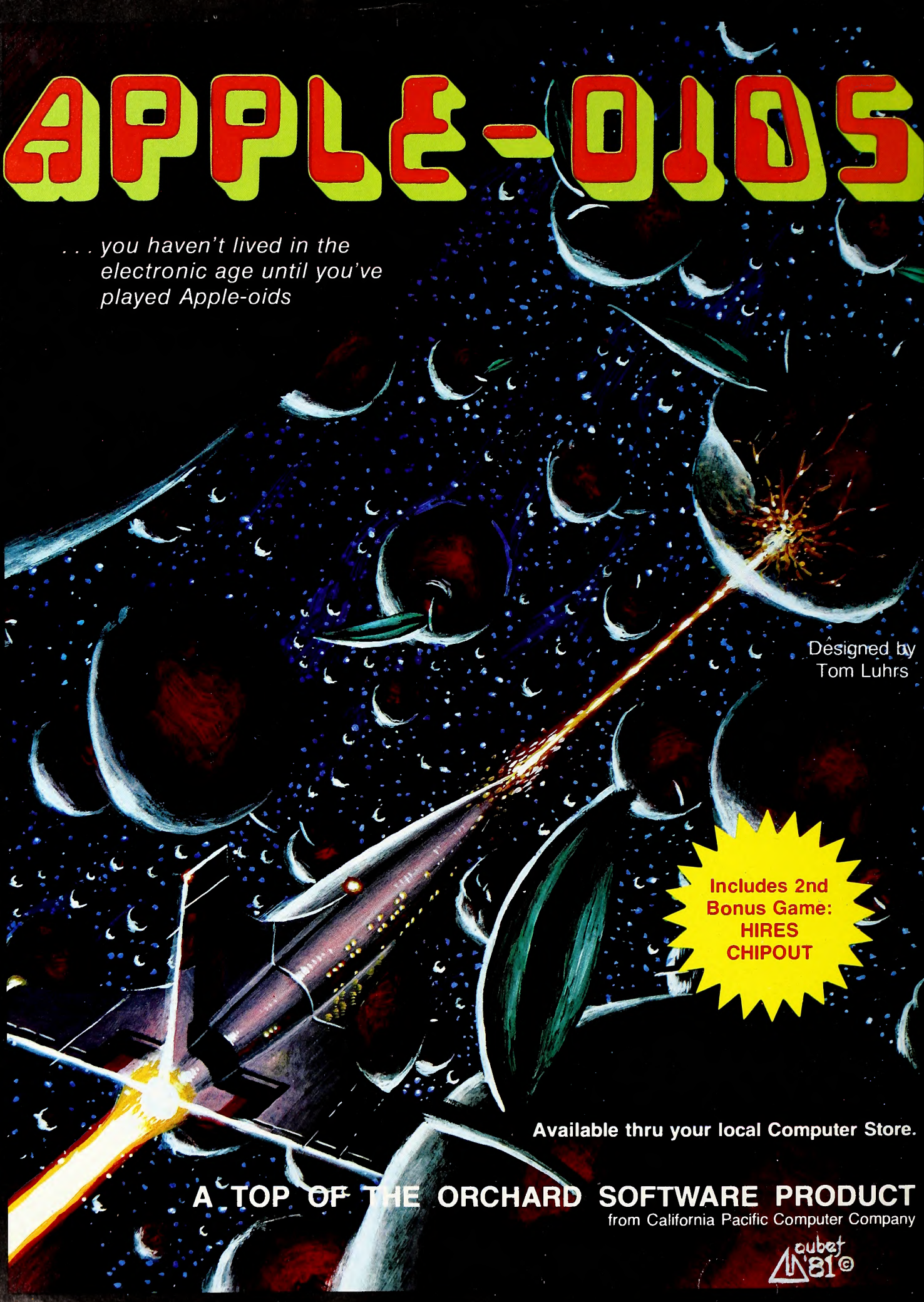
On Where We're Going

Exec Lotus

Eskimos and Apples Passing Down History

APPLE-OIDS

. . . you haven't lived in the electronic age until you've played Apple-oids



Designed by
Tom Luhrs

Includes 2nd Bonus Game: HIRES CHIPOUT

Available thru your local Computer Store.

A TOP OF THE ORCHARD SOFTWARE PRODUCT
from California Pacific Computer Company

aubet '81 ©

SOFTALK
JANUARY 1982

Chairman John Haller
Publisher Al Tommervik
Editor Margot Comstock Tommervik
Managing Editor Craig Stinson
Art Director Kurt A. Wahlner
General Manager Mary Sue Rennells
Editorial Associate Jean Varven
Special Assignments Melissa Milich
Copy Editor Andrew Christie
Art Assistant Donna Siebert
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
Basic Solutions Wm. V. R. Smith
Special Projects Greg Voss
Circulation Ron Rennells
Robert Mann
Systems Wm. V. R. Smith
Ken Williams
Advertising Coordination and Sales David Hunter
(213) 980-5074

Cover photo and design by Kurt Wahlner.
Composition by Photographics, Hollywood, California. Printing by Volkmuth Printers, Saint Cloud, Minnesota.
Apple and Applesoft are registered trademarks of Apple Computer Inc., Cupertino, California.
UCSD Pascal is a trademark of the University of California at San Diego.
VisiCalc is a trademark of Personal Software, Sunnyvale, California.
SoftCard is a trademark of Microsoft, Bellevue, Washington.

ISSN:0274-9629. *Softalk* is published monthly by Softalk Publishing Inc., 11021 Magnolia Boulevard, North Hollywood, CA 91601. Telephone, (213) 980-5074. Second-class postage paid at North Hollywood, CA, and additional mailing offices.

Postmaster: Send address changes to Softalk, 11021 Magnolia Boulevard, North Hollywood, CA 91601.

Free Subscription: Complimentary for one year to all owners of Apple computers in the USA. If you own an Apple, but you've never received *Softalk*, send your name, address, and Apple serial number with a request for subscription to Softalk Circulation, 11021 Magnolia Boulevard, North Hollywood, CA 91601. *Softalk* is totally independent of Apple Computer Inc.; sending your warranty card to Apple Computer will not inform *Softalk* of your existence.

Paid Subscription: After one free year, $24 per year without sponsor, $18 per year with sponsor. At the end of the free year, 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. For multiple subscriptions, deduct $6 per subscription. Special rates for schools and libraries, $8; multiple subscriptions for schools and libraries, $5 each.

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

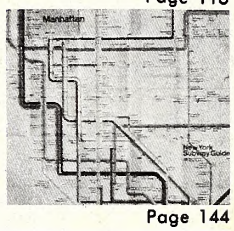
Problems? If you haven't received your *Softalk* by the fifteenth of the month, or if you have other problems with your subscription, Ron Rennells or Bob Mann can help out. Call (213) 980-5099.

Moving? Send new address and old to Softalk Circulation, 11021 Magnolia Boulevard, North Hollywood, CA 91601; telephone, (213) 980-5099.

CONTENTS


Page 24


Page 116

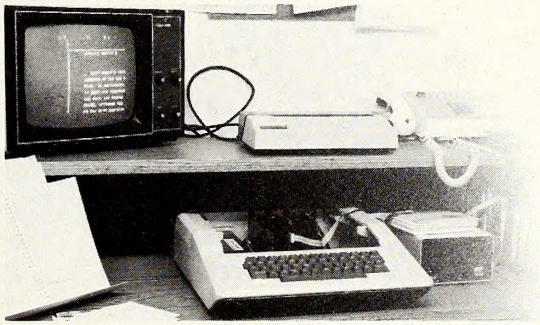


Page 144

FEATURES

DEPARTMENTS

PREVIEWS



Forword into February ... The Thinking Person's Teleconference: Communitree ... Exec Muse Software ... How To Use The Poscol Editor as a Word Processor ... and more.

ADVERTISERS INDEX

Apple JUMBLE

Philip Suh, of San Francisco, won September's Contest Contest. Apple Jumble is his winning entry. Here is his contest.

A group of ten oddly named Apple enthusiasts decided to get together to form an Apple users group. By a strange coincidence, the letters of each individual's name, when correctly rearranged, produced the name of an Apple-related product, either hardware, software, or firmware.

Determine the particular product represented by each name and then use the letters that fall in the squares to form the topic for discussion at the first meeting of this newly formed Apple users group.

Only the topic conceived by Philip Suh will be acceptable as the answer to the last part. It is possible, with so many letters, that other conceivable subjects could be formed. However, the correct answer is not at all awkward and is thoroughly relevant. So if the letters you get happen to form "computed the montauk islands," look further.

Deadline for entries is February 15, 1982. Prize is $100 worth of your choice of goods produced by Softalk advertisers from your local computer store. In case of ties, the Apple random generator will choose the winner. ◨

Sam Bredt
Sue Vander Rip
Rollo C Nethert
Dori Gafner
Hector Plagbits
Dingim Cawow
Toto Dikols
Commie Dorm
Niles Pety
Nudles Vobo II

The topic for discussion at the first meeting was:

_____ _____ _____ _____

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

Topic: _____ _____ _____ _____

Apple Jumble. Mail to Softalk Jumble, 11021 Magnolia Boulevard, North Hollywood, CA 91601.

Name _____

Address _____

City/state/zip _____

Choice of prize _____

Dealer _____

Your autograph _____

Note: There are two product-oriented guest contests, sponsored by the product makers, on pages 6 and 7. Both offer generous prizes!

Can YOU Survive?

BUG attack

The bugs are on the loose, and they're hungry.
Defend against ants' antics, malicious millipedes, and the menacing medfly invasion.
Bug Attack will keep the whole family captive with colorful creatures and superb sound effects.
Expert de-buggers who defeat the bugs in the three gardens beware, as they must face a five-fly formation with a deadly flight pattern!
Defeat the Bug Attack at your local computer store.
Written by James L. Nitchals

Brought to you by:

Bug Attack requires Apple II, 48k. (13 or 16 sector).
*Apple is a Trademark of Apple Computer, Inc.

Exclusively distributed by

SOFTSEL
8295 So. La Cienega Blvd., Inglewood, CA 90301
(800) 421-5770 In Calif. (800) 262-4242

CAVALIER COMPUTER™
P.O. BOX 2032 — DEL MAR, CA 92014 — (714) 755-8143

CONTEST WINNERS!

The Oracle '81—Grand Prize Winner. Jim Ganz, who led the Oracle '81 race for a disk drive, scored maximally on the final contest to secure first place in the year-long contest. Ganz, from West Hartford, Connecticut, achieved a cumulative score of +42, a considerable achievement because three of the six contest parts scored minus points against a perfect zero. Ganz's winning entry was his only entry in the contest.

Here's how he racked 'em up: Part 1 called for the combined winning margins of the 1981 Rose Bowl and Super Bowl winners. Ganz predicted 30, for a score of −4. Part 2 was the first of three chances for plus points: predicting the winners of the best actor, best actress, and best film Oscars. Ganz got them all for +30 points and a two-part total of +26. His prediction of 155 mph average speed for the winner of the Indy 500 was 16 off the mark; now he had +10 points total. Next came the prediction of Apple stock's price on July 30; Ganz's guess was $8 on the bullish side, for a diminishing total of +2.

The crazy baseball season that almost wasn't caused concern that there might never be an answer to part 5—naming the contenders and winner of the World Series. Ganz went with the odds on the teams that would play and picked up 20 points for the effort. Like all but six other people who predicted a Yankee/Dodger confrontation, Ganz put his money on the Yankees. Still, his total was now a respectable—and contest leading—+22. Finally, Ganz had the foresight to predict the company that would frequent the *Softalk* Top Thirty most often in 1981 for 20 more points and the grand total of 42.

Ganz will pick up his Disk II at A.M. Computer, Southington, Connecticut.

Runnersup in the Oracle '81 were Charles S. Lewis (Richmond, VA), +23⅝; Michael B. Preston (Culver City, CA), +21⅝; Paul Shanberg (Moraga, CA) +21⅜; Gary Kim (Seattle, WA) and Daniel Tobias (Poughkeepsie, NY), tied with +9⅜; Cliff Josephy (Brookville, NY), +9; Linda Weintraub (New Hyde Park, NY), +6⅜; Paul Shanberg again, +4⅜; and Gary Kim again, ⅝.

Oracle '81—The Hidden Race. While Jim Ganz was enjoying a good lead on the overall Oracle, several companies suddenly found themselves vying to be the one everyone should have picked for Oracle part 6. Going into December, Apple Computer itself was the only substantial contender against the leader, and contend it did. The two appearances on the Top Thirty by which Apple trailed could have been a piece of pie to the company that averages four to five appearances each month. But the less regular but very popular "small" software publisher hung on anxiously as Apple closed the gap to one. But that was it, that was the end of the rally. And On-Line Systems emerged as the company most frequently appearing in *Softalk*'s Top Thirty during 1981. On-Line's product's appearances totaled 43. After Apple's 42 came Sirius Software with 37, Personal Software with 28, Broderbund Software with 25, and Microsoft with 22.

Oracle '81—The Winner, Part 6. Many people send multiple entries to *Softalk* contests; ironically, the random generator has called the number of a single-entry contestant almost every time. The winner of part 6 entered forty-three times for the Oracle. Did the random number generator change its ways? No. Winner Ethan Starr chose On-Line Systems as the answer to part 6 on only one of those entries, so only one was entered in the tie-breaking.

Starr, who lives in Amherst, Massachusetts, will choose a prize from the Retail Computer Center in Northampton, Massachusetts.

Nasty Puns 'n Anagrams Explained. P. Cucka (Scotch Plains, NJ) was one of only seventeen correct solutions to the crossword puzzle. Many entries came very close, with only one word or even one letter off (*core* for *code*, for example). Every entry wins applause, though, for super effort; the clues were nasty.

Here's the answer, with the clues explained.

Across

2. Applesoft. *Giving* (synonym for soft) *fruit* (genus of *apple*) returned (meaning read it backward).
6. Printer. *Terrapin* gets no A's (remove a from *terrapin* and anagram to get *printer*) when *machine* (genus definition) copy is hard (bonus clue: printers produce "hard copy").
9. Basic. *B*achelor of *A*rts (take common abbreviation, BA), as it is (translation of Latin *sic*). This form of clue is called a charade.
10. And. *Plus* (definition) *genetic part* (DNA) backward (reverse DNA).
11. ROM. Less than *more* (drop e from more to get "mor") back (reverse *mor* to get *rom*) only to read (functional clue: read only memory).
12. Teams. Members of the AFL (American Football League—members are football teams, thus definition) gather to feast on many *meats* (anagram).
13. Chip. Screwball *pitch*, no tea (drop t from pitch, anagram—from screwball), with part of tollhouse cookie (definition; part of tollhouse cookie is a chocolate chip).
15. Load. "Hello, *Ad*eline" (hidden

GOTO 114

Explosive hi-res action from Brøderbund Software!

RED ALERT by Olaf Lubeck

Your civilization is under attack by the stinging space meanies and vicious thud putters. A protective shield slows their assault, but without quick counteraction your defenses will crumble one by one. Requires 48K Apple II Integer or Plus and will boot on either DOS 3.2 or 3.3. $29.95

Space Warrior by Marc Goodman

On the far outposts of the galaxy the Space Warrior waits — protecting the Empire from the attacks of the dreaded ram ships. Requires 32K Apple II Integer or Plus and will boot on either DOS 3.2 or 3.3. $24.95

Each game is fully guaranteed. If it ever fails to boot, for any reason whatever, return the disk to Brøderbund Software, Inc. for a free replacement. If the disk has been physically damaged, you will be charged the cost of a replacement disk.

Ask your dealer or send check or money order to:

Brøderbund Software, Inc. • 2 Vista Wood Way San Rafael, CA 94901 (415) 456-6424

The Great Trick Shot Tournament

When Innovative Design Software introduced *Pool 1.5* to the Applemarket last year, the smooth animation of the complicated physics involved in simulating pool caused such a stir that few ever questioned what happened to *Pool 1.0* or whether there would be a *Pool 2.0*.

There's still no answer to those questions. Instead, IDSI followed *Pool 1.5* with *Trick Shot*, a package that includes several other variations on pool as well as the ability to set up and test your own trick shots.

Trick shots are the ultimate test of a pool player's ability to handle his cue ball; and control of the cue ball will determine a pool player's fate in any contest. So trick shots have always served the player as a demonstration of his ability to exercise exacting control of the cue ball while sinking one or more object balls.

Can he put enough draw on the cue ball to pull it the length of the table and sink object balls in opposite pockets? Can he sink six balls with one shot without taking any ball except the cue ball off a rail?

Trick shots are the ballet of pool, combining the highest artistry with the greatest skill. *Trick Shot* has that aura within the programming community. Among the innovations implemented by authors Don Hoffman, Howard de St. Germain, and Dave Morock are pixel-level accuracy in the placement of each ball on the table.

Moving one ball one pixel on the screen seems insignificant; but the actual differences in results from such minute moves are the true indication of the importance of such accuracy.

Now the authors of the program want to determine if providing that level of accuracy was worth the effort, and they're willing to pay handsomely to find out.

The Great Trick Shot Tournament should elicit the best efforts of computer owners across the country, if for no other reason than that the first prize is $1,000. It also gives everyone a chance to pit his imagination and skill against those of fellow computer owners.

The object of the tournament is to consummate the most artistic, the most imaginative, the most innovative, and the most difficult pool shot that can be conceived.

In addition to the $1,000 first prize, there will be second and third prizes at the national level as well as a prize to the best entry from each state. Second prize is worth $500, third prize is $250 and each state winner will earn $10.

The *Trick Shot* disk has several examples of fancy shots on it to give contestants an idea of what kind of shots might qualify for prize money.

The tournament rules:

1. Design one or more shots that you believe represent the epitome of trick shot artistry. It is not necessary that the shots be makable on a real pool table so long as you can execute them with the *Trick Shot* program.

2. Save your best shots to disk and mail them to Softalk IDSI Trick Shot Tournament, 11021 Magnolia Boulevard, North Hollywood, CA 91601. Limit the number of shots submitted to three per disk. No disks will be returned.

3. There are no limits as to the speed, number of balls, or kind of English applied to the cue ball. However, only shots submitted that use friction settings 2 or 3 will be considered for prizes.

4. Judges will consider imagination, finesse, degree of difficulty, and the innovative aspects of the shots. Decisions of the judges are not subject to appeal.

5. Overall prizes of $1,000 for first place, $500 for second place, and $250 for third place will be awarded. The best entry from each state will receive $10. The District of Columbia, all U.S. territories, and all foreign countries will be considered as one state for judging purposes.

6. Entries must be postmarked not later than February 28, 1982. All entries must include the following information: name, street address, city, state, zip code, and telephone number of the entrant and the name of the dealer from whom the *Trick Shot* program was originally purchased.

Winners will be announced in the May issue of *Softalk*. ◨

Enthusiasm of computer owners for nationwide contests has grown tremendously in the last year, fueled partially by the regular contests appearing in *Softalk* and partially by independent contests sponsored by software publishers and arcade manufacturers.

Softalk, with rare exceptions, has kept its contests from being product oriented so as to avoid the suspicion of favoritism for one advertiser or product over another.

However, the number of innovative software programs conducive to national contests continues to grow, making possible this new section of *Softalk* wherein we announce contests sponsored by other companies but administered by *Softalk*.

In this issue, we have two such contests, one for the arcade addict and one for the pool enthusiast. The prizes are significant and the contests should provide you with hours of fun.

Are Chet and David Sirius?

Sirius Software wowed 'em in Peoria as well as Silicon Gulch when they published *Epoch*, an arcade game by Larry Miller with some of the smoothest animation and best three-dimensional effects yet seen on the Apple.

Miller's next, *Hadron*, features the same high quality graphics in a space chase to destroy an enemy base.

Along the way, the player sees many diverse objects in space, not the least of which is a space ship carrying two chatty characters by the names of Chet and David.

When the game ends, a musical theme heralds your success—or merely your finish.

The characters Chet and David and the musical theme are the basis for two separate contests sponsored by Sirius.

One contest asks you, in 250 words or less, to re-create the conversation between Chet and David. What are they saying? Judging will be based on originality, wit, and appropriateness to the situation the duo find themselves in.

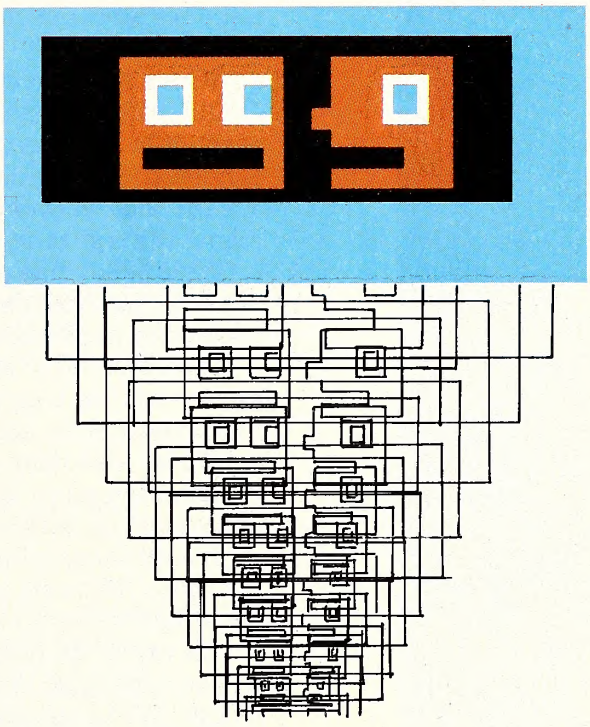
First prize is $250 for the most creative conversation. Second prize is $100. Eight honorable mention prizes of one product of the entrant's choice from the Sirius catalog will also be awarded.

The second contest requires that you complete the game and hear the musical theme. If you can identify that theme correctly, you're in the running for a $250 first prize. All correct entries will be thrown into Apple's random number generator to determine the winner.

Drawings will also be held for a $100 second prize and eight honorable mention prizes of one product of the entrant's choice from the Sirius catalog.

You must submit a separate entry for each contest. The entry must include your name, street address, city, state, zip code, telephone number, and place of purchase of *Hadron*, and must reach *Softalk* by February 15, 1982.

Address the entries to Softalk Sirius Conversation or Softalk Sirius Music, 11021 Magnolia Boulevard, North Hollywood, CA 91601. ◨



SOFTALK CHANGES ITS

Christmas 1981 had a pall over it for the folks at *Softalk* because they knew the text of the following message that would serve as our New Year's greeting to you, our readers:

Effective immediately, *Softalk* will no longer be provided free to all owners of Apple computers.

Instead, all Apple owners will be given a one-year complimentary subscription, after which they will be given the option of dropping the magazine or converting their complimentary subscription to a paid one.

The decision to make the change in policy was an easy one. Economics dictated that *Softalk* either find alternate sources of revenue or cease publishing.

Living with that policy change has been somewhat more difficult. Because the fact is that we at *Softalk* have genuinely enjoyed giving each of you the best magazine we could produce each month. In a way, seeing the first issues off the press made each month seem a little imbued with the Christmas spirit.

Our fondest wish through these seventeen issues has been that some of that enjoyment and spirit has seeped into the magazine and that you've been able to share it.

Softalk never has pretended to be a typical computer magazine. In our very first issue we stressed that the emphasis here would be on journalistic style rather than technical data.

We've made every attempt to keep that pledge. No other computer magazine, in or out of the Apple world, makes an attempt to cover the issuance of each new product in its chosen field of coverage. But between Marketalk News and Marketalk Reviews, *Softalk* does exactly that each month, even when the products seem so arcane that we wonder why anyone would have ever bothered to dream them up, let alone attempt to make them commercially available.

That policy has paid off in the satisfaction we get when the telephone starts ringing with inquiries for more information on such products. Some of them we still don't pretend to understand—we're still coming to grips with the depth of technical expertise and breadth of applications interest in the Apple market ourselves—but we've been able to realize true pleasure in seeing a frustrated user get together with a nascent manufacturer to accomplish something never before done with an Apple computer.

We've also been able to bring you applications articles designed to stretch your imagination as to what you can do with an Apple. Is it a game machine? A word processor? A number-cruncher? A keeper of organized data?

It's all these, of course, but the amazing thing about the Apple is that it's become far more than anyone, Steve Jobs, Steve Wozniak, Mike Markkula, or John Couch, could have ever imagined.

An esprit de corps has surrounded the Apple that almost approaches religious fervor. And what Apple owners won't attempt to do with that machine defies imagination. We've attempted to bring you some of those stories.

We've also tried to adapt ourselves to the mold you've perceived as fitting for a computer magazine.

In that regard, we've constantly monitored our mail and added columns that seemed to be in demand.

The only noticeable difference in the short run between the *Softalk* you received this month and the *Softalk* you'll receive in February is that the February magazine will be given a protective wrapping to spare it the ravages of the U.S. Postal Service.

The wrapping, in whatever form it may take now or in the future, could not be afforded when *Softalk* was shipping for free.

In addition, the likelihood is that you'll see more feature coverage from a wider geographical spectrum than ever before, contests with bigger prizes, and more articles such as the ones on compilers and Fortran that are long lead-time items.

But generally, what you'll be getting for your money is exactly what you previously got for free—*Softalk* magazine as you see it now.

Implementation of the paid subscription policy is as follows:

1. All subscriptions are considered to have started with the October 1980 issue. September was mailed courtesy of Apple Computer Inc. and will not be counted against the one-year free subscription.

2. Dating from the first month in which our records show that a subscriber has received continuous service, twelve issues will be mailed on a complimentary basis.

3. Subsequent to the twelfth complimentary issue, a notice of renewal will be mailed, advising of the changeover to paid subscription. All subscribers who do not choose to renew will be promptly removed from the active circulation rolls. Under no circumstances will any Apple owner ever be billed for issues not ordered. The notice of renewal is not an invoice, but a reminder of a change in status.

4. Beginning with the February issue, the cover price of *Softalk* will be $2.50, or $30 per year. The one-year sub-

DISTRIBUTION POLICY

We began with a tutorial in machine language by Roger Wagner. Since then we've added columns on *VisiCalc,* SoftCard, Business Basic, Applesoft, investments and finance, business applications, Basic subroutines, Pascal, a tutorial for beginners, and, as of this month, a column on peripherals.

Our columnists have been selected on the basis of their knowledge and their ability to communicate information in a literate manner.

We've derived pleasure from your cards and letters, which often have described how some tidbit in *Softalk* helped solve a problem you had been wrestling with.

We've also derived pleasure in the last seventeen months from seeing the blossoming of the Apple market. The list of companies and products that did not exist that short time ago is lengthy; contemplate that Broderbund, On-Line Systems, and Sirius had released no product and *DB Master* was still on the drawing boards when we started planning *Softalk* in May 1980. *VisiCalc* had just been released.

The people who bring you the programs and peripherals that drive your Apple have been living their own Horatio Alger stories, and we've attempted to keep you abreast of them and how they did it. It's been said before in the pages of this magazine, but, by and large, some of the most genuinely fine human beings on the planet function in the Apple marketplace.

But most of all, we've enjoyed making our efforts complimentary to each of you. It is indeed more blessed to give than to receive, and we at *Softalk* have been truly blessed during these past seventeen months with some of the finest hours any journalist could hope for.

Below you'll find what we hope is a complete statement of the new policy. Because we've never done this before, there may be bugs in the program. If you detect one, please inform us.

Also below you'll find a list of everyone who contributes, either editorially or through such heart-breaking and mundane efforts as preparing our sixty-thousand-name mailing list for printing each month. Regardless of whether you ratify our past efforts by choosing to now pay for *Softalk*, please be assured that we've all received great pleasure from providing it for you.

Al Tommervik, publisher
Margot Comstock Tommervik, editor
Kurt A. Wahlner, art director
Craig Stinson, managing editor
David Hunter, ad coordinator
Mary Sue Rennells, general manager
Melissa Milich, special assignments
Jean Varven, editorial associate
Andrew Christie, copy editor
Donna Siebert, art assistant
Ron Rennells, circulation
Bob Mann, circulation
Greg Voss, special projects
Roger Wagner, columnist
Jim Merritt, columnist
Jeffrey Mazur, columnist
William V. R. Smith, columnist
Peter Olivieri, columnist
Ken Landis, columnist
Taylor Pohlman, columnist
Greg Tibbetts, columnist
Doug Carlston, columnist
Ken Williams, systems

---

scription rate will be $24, with the following exceptions:

a. Apple owners who receive *Softalk* compliments of their local retail store or a software publisher will be granted the courtesy rate of $18 per year. *Softalk* will regularly publish a list of those retail outlets and software publishers who have joined this program for the information of subscribers.

b. Apple owners who desire multiple subscriptions may deduct $6 per subscription for each subscription after the first.

c. Public libraries, nonprofit institutions, and schools of all kinds, public and private, elementary, secondary, and higher education, receive the special rate of $8 per year for the first subscription and $5 per year for each additional subscription.

5. *Softalk* will not recognize as a break in continuous service the failure of the U.S. Postal Service to deliver any issue that *Softalk's* circulation list shows was mailed. Subscribers should inquire about replacement copies of non-delivered issues approximately on the 15th of the month of issue.

6. No break in service, for whatever reason, subsequent to the receipt of twelve free issues will entitle an Apple owner to additional complimentary issues.

7. *Softalk* welcomes written inquiries as to the applicability of the subscription rates to any individual. Such inquiries must state clearly what question is being raised and postulate the answer from the subscriber's viewpoint.

8. Any paid subscriber may terminate his subscription at any time for any reason and request a full refund of the unused portion of the subscription rate.

9. Under no circumstances should anyone presently receiving *Softalk* remit money until such time as they receive their notice for renewal.

OPEN DISCUSSION

The Address Controversy

Please send me the address of Shakti Systems, manufacturer of the Pegasus DBMS for Apple Pascal, as mentioned in Marketalk of your August 1981 issue.

I, like Mr. Peter Wagner (letter, September issue), would like you to give the addresses of the manufacturers listed in Marketalk and other columns in *Softalk*. I understand your wish that readers patronize their local Apple dealer and sincerely wish that I could. I am sure that there are many other *Softalk* readers who either do not have a local Apple dealer or whose dealer has no knowledge of recent products for the Apple. We, and the manufacturers, your advertisers, would benefit by being able to contact one another.

Gerald Perkins, APO San Francisco, CA

You'll find addresses accompanying all items in Marketalk beginning in this issue of Softalk.

We continue to encourage our readers to purchase through their dealers whenever possible. However, with the increase in our readership, our capacity to deal individually with requests for addresses from readers such as Gerald Perkins, who is with the armed services overseas, has been sorely tried.

With the exception of a few mail-order retailers who specialize in service for people such as Perkins, no one can give you the customer support and service that your retailer can. Manufacturers and publishers are seldom set up to deal with service problems; some will not respond to direct orders but will send you to your dealer or recommend another source for their wares.

Because we don't always have access to phone numbers, those will not accompany addresses in Softalk. *Therefore, if you wish to phone a company, please call information.*

A Kick When You're Down

While your magazine is a greatly needed interface between consumer and industry, it is interesting to note that the consumer still seems to get short-circuited.

I read in a recent issue of your magazine that Broderbund Software was offering an upgraded joy-stick version of *Snoggle* for those who had purchased the keyboard version. I anxiously attempted to take advantage of this offer. However, upon contacting Broderbund, I was informed that I was a couple of days late; they had sold all *Snoggle* material to Atari and cannot deal with this product. They suggested that I contact "my" Atari dealer and continue to "enjoy" the version that I have.

I'm certain "my" Atari dealer will love to take care of me and my Apple. Maybe he will offer me an upgraded version if I agree to take receipt of an Atari 800. Sounds like a deal to me.

As most probably know, it is not all fun and games the way it is written in your magazine. I just wish to make you and perhaps some readers aware of this type of unfortunate "big business" dealing that will sour the appletite of many personal computerists. If things like this continue, it will be easy justification for anyone to bootleg, rob, or steal such software. "Aye—Mate, pirate or perish" might be the only answer.

Jerry Roberts, Melbourne, FL

The letter you received from Broderbund stated, ". . . as of October 21, all Snoggle *material became the property of Atari. . . ." You have jumped to the conclusion that Broderbund sold* Snoggle *to Atari. They did not. Threatened lawsuits from Atari and injunctions from that company and its parent, Warner Communications, based on the similarity of* Snoggle *to the arcade game,* Pac-Man, *were more than a small, family-owned company such as Broderbund could fight. Therefore, Atari took* Snoggle.

The apology in your letter from Broderbund is sincere.

The question is, indeed, one of piracy, or, at least, of copyright violation. Programmers such as Jun Wada did not see the translation of a program from one medium—the arcade—to another—the Apple home computer—to be a violation. Neither, at first, did the arcade gamemakers. Recently, the worm has turned, and Atari has been demanding royalties on and/or removal from the market of any products that resemble their own. In many cases, their objection is open to question, but most software companies don't have the resources to stand up to the multimegabusinesses behind Atari. Ken Williams of On-Line Systems, with slightly *more resources than others and with a great deal of courage, is doing so.* Softalk *will keep you informed of the results. (See Tradetalk, page 44.)*

The controversy will be covered in depth in the January issue of Softline, Softalk*'s sister publication.*

Even if Broderbund's actions had been everything you interpreted them as, it would not constitute the least excuse for you to engage in piracy. Boycott—doing without a program—is an appropriate response to business practices you don't like; theft is not.

When DOS Boss Hits Again, Cassidy Will Be There

As a software author for the Apple, I toil in relative obscurity. My friends always ask me what do I do all day? Why do I sit at home, instead of working, like a normal person? I think they think I watch soap operas all day.

You can imagine my elation when I opened my November issue of *Softalk* and saw that one of my programs, *DOS Boss*, had made it to the top thirty! And not only that, but it was number three on the Hobby 10, right behind *DOS Toolkit*!

Despair followed on the heels of joy. I soon noticed that my name, Jack Cassidy, was not included in the entry. How could this have happened? My name is right next to the author's, Bert Kersey's, name everywhere on the package and in the program. Is there some foul plot to frustrate my childhood ambition for recognition in the microcomputer industry???

So now, I go around to my friends, and I show them my copy of *Softalk* in which I have penciled in my name next to Bert's. And I say, "That's me there. That's what I do all day. I've made it to the big time now." And they all smile and say they're happy for me, but as soon as I turn my back I can hear them laughing and making jokes among themselves.

So *Softalk*, my question to you is this: Would it be possible for you to put my name in as coauthor of *DOS Boss*, or should I just pencil my name into the *VisiCalc* entry, and go for the really big time?

Jack Cassidy, San Diego, CA

This Blackjack System Works

We were all delighted at the fine review

you gave *The World's Greatest Blackjack Program.* I enjoy your magazine immensely, but it's even more fun to read about something we've done.

My only concern was in your use of "allegedly" in describing the power of the Hi-Opt I card-counting system. I know this fits because you haven't seen any of the academic material, but the Hi-Opt I is generally considered the most powerful, simple system available.

Julian Braun, an IBM employee and widely regarded as the final authority in the performance analysis of gaming systems, established the merits of the Hi-Opt I in his paper, "The Development and Analysis of Winning Strategies for the Casino Game of Blackjack." He rated the Hi-Opt I as having an advantage of between 0.8 percent and 2.8 percent *over* the house for single deck Las Vegas Strip rules. While exact performance depends upon many factors, there is no question in anyone's mind of the Hi-Opt being able to beat the casino.

I know several high rollers who quickly justified the entire Apple system solely for its ability to teach them how to win at blackjack using *The World's Greatest Blackjack Program.*

It's a minor point, and this letter is too long, but I don't think it would hurt to consider an item in the Letters section "clearing" the slur on the Hi-Opt I system.

Carl Cooper, Ph.D., Ballwin, MO

An Open Letter to Rudy Lauer

Thanks for your thoughtful response to my challenge. I appreciate the concern expressed in your letter to *Softalk* regarding my claims for *Absolute Security.*

I realize that my challenge cipher does not rigorously prove the insolubility of my system; however, I view the *Softalk* challenge as an effective way to expose the merits of *Absolute Security* to the public. It seems to be a popular belief everywhere I turn that *any* code is solvable, given enough time and computing resources. In order to sell my product, I am faced with the difficult task of overcoming that belief.

Yes, *Absolute Security is* a true "One-Time-Pad" system. Its strength depends entirely on specific keys created by each user. I realize that computers are only capable of generating "pseudo-random" sequences: therefore, *Absolute Security* has *two* sources of keys. One, based on a twelve-character "seed" input from the keyboard, is a pseudo-random generator. It takes the twelve-character seed as the basis for a 72-bit generator (after a Knuth algorithm). In my documentation for the program, I have been very careful to point out that using this pseudo-random generator does *not* provide absolute security—rather, it is included as a convenience for higher volume, lower security applications.

The other source of keys uses a hardware function readily available on every Apple—its cassette interface. By placing a cassette recorder in monitor mode and connecting it to the Apple (i.e., utilizing it as an audio amplifier), it is possible to compile highly random files from ambient sounds (e.g., speech, room noises, dogs barking, traffic sounds, rain, etc.). This effectively handles the difficulty of creating large quantities of unique keys by making key production almost automatic.

My documentation cautions users that they *must* use keys *one time only* and then destroy them. I have also been careful to point out the need for *knowing* that keys have not been compromised in transit between sites.

Since I view this system as having its major application in the area of communication (rather than storage protection), I don't see key loss as a major problem. If a key is lost or stolen, simply delete all copies and don't use it for encoding data. It is far better to lose a key in transit (since keys have no integral value) than to lose a real document containing valuable data.

I recognize that physical security at both ends of the communication path is vital. This is true of any secure communication system, however, and *not* just of *Absolute Security*. Physical security is much more straightforward than securing a communication link, and I view it

as a separate problem best addressed by each individual user of *any* coding system.

In summary: I stand by my claims for *Absolute Security*. Additionally, I feel that, in conjunction with the modern hardware of the Apple computer and the media of magnetic disks, *Absolute Security* overcomes many of the traditional objections to the use of "One-Time-Pads" by simplifying and automating many functions. This leaves the *major* benefit of the One-Time-Pad: its *mathematically provable* insolubility.
Dann McCreary, San Diego, CA

A Hex on the Corner

To Craig Stinson: I enjoy reading your Beginners' Corner very much. I don't think you wrote this article in the November 1981 issue. It doesn't sound like you.

I also found a very bad mistake for you to have made in writing an article for beginners. In the third column, second paragraph from the top, starting with the words "Counting in the binary system," you use the binary number 1011 for the decimal equivalent of thirteen instead of the binary number 1101.

I also found in the second to the last paragraph that you say that the binary number 1100 is the decimal number eleven. This is not true. It is equal to twelve. The hexadecimal is C6.
Daniel W. Zarwell, Milwaukee, WI

The Digital Woodman

I do woodworking as a hobby and am often forced into making decisions on cutting lumber that I am sure are not optimal. I buy unfinished lumber and am able to resaw and plane the material to varying thicknesses, lengths, and widths. My problem is, on the one hand, having a list of required boards of finished dimension and, on the other hand, having a pile of unfinished lumber of varying dimensions that can be cut and finished into almost unlimited dimensional possibilities.

The ideal situation, of course, is to be able to utilize the minimum amount of lumber with essentially no waste and have the largest possible unfinished boards left over. Is anyone aware of a program that I can utilize with my Apple II that can help me make the proper decisions on what unfinished lumber to use in order to produce a list of finished boards of known dimensions?

On another note, I have read at least two advertisements for program-writing programs that sell for about $600 each. Is there any plan to report on these in a future issue?
Lance Goddard, Belmont, CA

Answering Their Own Question

I had you put a letter in *Softalk* about entry addresses in Applesoft; meanwhile we have been asking around about such a list. Enclosed is such a list that I typed from a friend. These have not *all* been tested but the ones we've used have worked. (Some are 2 to 4 off but can be found—see article 1.) He derived this list by using the token table in the reference manual plus making educated guesses from looking at the machine language. Some entry addresses need to be seen to load registers. I also was told of two articles that deal with this subject: 1. "Notes on Hi-Res Graphics Routines and Applesoft," by C. R. Mesztenyl, *Apple Orchard* (Spring 1981). 2. "Applesoft Internal Entry Points," by J. Crossley, *Apple Orchard* (March-April 1980). I have not seen the latter. The former is very helpful.

We'll be grateful for any other information we get through our letter in the magazine. Any corrections to the list, too—and additions.

Entry Addresses for Applesoft Keywords

TOKENS:			ADDRESS	
HEX	DEC	KEYWORD	HEX	DEC
80	128	END	D870	55408
81	129	FOR	D765	55142
82	130	NEXT	DCF9	56569
83	131	DATA	D995	55701
84	132	INPUT	DBB2	56242
85	133	DEL	F331	62257
86	134	DIM	DFD9	57305
87	135	READ	DBE2	56217
88	136	GR	F3F9	62457
89	137	TEXT	F399	62361
8A	138	PR#	F1E5	61925
8B	139	IN#	F1DE	61918
8C	140	CALL	F1D5	61909
8D	141	PLOT	F225	61989
8E	142	HLIN	F232	62002
8F	143	VLIN	F241	62018
90	144	HGR2	F3D8	62424
91	145	HGR	F3E2	62434
92	146	HCOLOR=	F6E9	63209
93	147	HPLOT	F6FE	63230
94	148	DRAW	F769	63337
95	149	XDRAW	F76F	63343
96	150	HTAB	F7E7	63463
97	151	HOME	FC58	64600
98	152	ROT=	F721	63265
99	153	SCALE=	F727	63271
9A	154	SHLOAD	F775	63349
9B	155	TRACE	F26D	62061
9C	156	NOTRACE	F26F	62063
9D	157	NORMAL	F273	62067
9E	158	INVERSE	F277	62071
9F	159	FLASH	F280	62080
A0	160	COLOR=	F24F	62031
A1	161	POP	D96B	55659
A2	162	VTAB	F256	62038
A3	163	HIMEM:	F286	62086
A4	164	LOMEM:	F2A6	62118
A5	165	ONERR	F2CB	62155
A6	166	RESUME	F318	62232
A7	167	RECALL	F3BC	62396
A8	168	STORE	F39F	62367
A9	169	SPEED=	F262	62050
AA	170	LET	EA46	55878
AB	171	GOTO	D93E	55614
AC	172	RUN	D912	55570
AD	173	IF	D9C9	55753
AE	174	RESTORE	D849	55369
AF	175	&	03F5	01013
B0	176	GOSUB	D921	55585
B1	177	RETURN	D96B	55659
B2	178	REM	D9DC	55772
B3	179	STOP	D86E	55406
B4	180	ON	D9EC	55788

TOKENS: HEX	DEC	KEYWORD	ADDRESS HEX	DEC
B5	181	WAIT	E784	59268
B6	182	LOAD	D8C9	55497
B7	183	SAVE	D880	55472
B8	184	DEF	E313	58131
B9	185	POKE	E77B	59259
BA	186	PRINT	DAD5	56021
BB	187	CONT	D896	55446
BC	188	LIST	D6A5	56389
BD	189	CLEAR	D63A	54906
BE	190	GET	DBA0	56224
BF	191	NEW	D649	54857
C0	192	TAB(	EB91	60305
C1	193	TO	EC24	60452
C2	194	FN	EBB0	60336
C3	195	SPC(	000B	00011
C4	196	THEN	E2DF	58079
C5	197	AT	D413	54291
C6	198	NOT	DFCE	57294
C7	199	STEP	E200	58112
C8	200	+	EE8E	61870
C9	201	−	EFAF	61359
CA	202	*	E942	59714
CB	203	/	EF0A	61194
CC	204	^	EFEB	61419
CD	205	AND	EFF2	61426
CE	206	OR	F03B	61499
CF	207	>	F09F	61599
D0	208	=	E765	59237
D1	209	<	E6D7	59095
D2	210	SGN	E3C6	58310
D3	211	INT	E708	59144
D4	212	ABS	E6E6	59110
D5	213	USR	E646	58950
D6	214	FRE	E65B	58971
D7	215	SCRN(	E687	59015
D8	216	PDL	E692	59026
D9	217	POS	C07A	49274
DA	218	SQR	79E8	31208
DB	219	RND	E7AA	59306
DC	220	LOG	817C	33148
DD	221	EXP	7BEA	31722
DE	222	COS	EA69	60009
DF	223	SIN	967E	38526
E0	224	TAN	50EF	20719
E1	225	ATN	DF55	57173
E2	226	PEEK	4E47	20039
E3	227	LEN	7FE0	32736
E4	228	STR$	EED0	61136
E5	229	VAL	9780	38784
E6	230	ASC	64DF	25823
E7	231	CHR$	DF65	57189
E8	232	LEFT$	4E46	20038
E9	233	RIGHT$	46C5	18117
EA	234	MID$	D250	53840
		X-COLOR	F6EC	63212
		x low-x	F457	62551
		y low-y	F605	62981

Carol Colter, Whittier, CA

Medley of Thoughts from Nowhere

I live in Barstow, California, where there are no computer magazines for sale, never mind computers. Plus, I have my Apple at work in my office at Fort Irwin which is thirty-seven miles north of the middle of nowhere. So, I need a little help, please.

Can I access my MX-80 through *Apple Writer?* You stated *Magic Window* can do it through control-B. Does *Apple Writer* have a similar device? If not, which other word processors can control the MX-80 print styles from text embedded commands?

My Apple was malfunctioning the other day by intermittently blanking out the screen with reverse and full green video. It turns out some of my ICs had almost worked loose. After they were pushed down tight, everything worked normally.

On-Line Systems promptly and politely helped me with a problem I had with one of their adventures (I admit it; I was stumped over the combined notes in the *Wiz*). All things not being equal, I recommend that the water level in that infamous hot tub be lowered about eight inches. And put me in it!

When can we talk to our Apples? The revolution will truly start when we, the masters, can talk to our machines, the slaves. They now respond to buttons and levers. I want a three hundred dollar card that will let me talk to the Apple with a vocabulary of twenty-five words with the capability of constructing an infinite number of words based upon the first twenty-five. Is that too much to ask?

The hardest part of living out here is not being able to see the appetite-whetting programs in operation. Thirty to a hundred dollars is too much to pay for a "looksee." But then, when I go into a computer store, what do I see? A bunch of dead machines with no one around that can adequately bring them to life. Imagine a car showroom where the salesman can't start or drive the car! That's the situation in many Los Angeles computer stores—no action.

In a software store recently, I had a hell of a time talking the manager into running *Superscribe II* for me. I wanted to see the seventy-column format on the screen without the eighty-column card installed. After much cajoling, he opened the package up, booted up, and then gave up. He couldn't put the format on the screen and he wouldn't let me try. He wants me to pay $130 for a program I can't see in action? (I'm sure it wasn't the program's fault.) As we say: the buyer-seller interface has some bugs.

Don't you love the ads on TV by the Big Boys? They are selling computers like selling a restaurant because the waitresses are pretty. The sizzle of computers needs to be sold in those TV ads. The sizzle is software. How shall I put it? "Sell sizzling software!"

J. Barry Smith, Barstow, CA

International Apples

Mr. Gregory Enos asked for names of Apple computer users' groups in foreign countries (October 1981, page 9). While I am unaware of any official group here in Rio, I do know many of the Apple owners. Perhaps if he were to contact me for whatever help he wants, a "users' group" would appear.

John D. Trotter, Rio de Janeiro, Brazil

Analyzing Diet Analysis

I was somewhat hesitant to comment on one of the programs reviewed in Marketalk Reviews in your November 1981 issue. I did feel obligated, however,

to warn your other readers that the *Diet Analysis* program written by Javed Aslam and marketed by Apple as part of their SDS program has design deficiencies and defects which render it of limited value to the serious user.

I purchased *Diet Analysis* for my wife on the basis of the description published in Apple's SDS catalog. It sounded like a great way for her to keep up with her intake of carbohydrates and all the other things she loves so well. The first problem encountered was that the program would crash when an attempt was made to activate one of the functions from the menu. Since the disk was locked to prevent the user from copying the program and the various tricks had been used to reset the *reset* vector and prevent listing the program, the only recourse was to turn off the computer. In addition to this fatal flaw, it turned out that, contrary to indications in the SDS catalog, it was not possible to add foods to the data base to tailor it to the user's likes. Finally there was no provision for printing any of the results. Since the program had to be booted to run it, it was not possible to install a screen print routine and, for reasons stated above, it was not possible to add printing options to the code. Fortunately, the local store took the program back.

Once again, I regret having to throw cold water on Dr. Aslam's efforts, particularly since the program otherwise has a lot of merit. I am disappointed, however, that Apple did not try a little harder to ensure that the program would run as the author intended, and that their catalog was misleading. Perhaps I would have kept the program had I been permitted to modify the code to do what I wanted.
Mike Kramer, Kingwood, TX

Wants To Copyright Supergame
I've written an adventure game to market through a software firm, and, as it is the first such program I've ever written, I would like to know the procedure for copyrighting it. I've done my level best in cheatproofing the code, but modifying DOS 3.3 is still beyond my expertise. I hope to have the program hit the stands before the end of next year.

If public response is sufficient, I will develop an idea into a second adventure game and have that marketed. Perhaps, someday, I can resign my present job as a chemistry lab technician, but I keep such musing under strict control.

Some of my cheatproofing efforts involved the use of programs like the Extended Memory Dump by Curt Deegan, which was simply thrown away by being published in *Apple Orchard*. I've seen some blame good marketable software thrown away by release via a magazine. A software magazine is for short utilities, patches, news on computer developments, ads on software and hardware reviews, and seminars on computer uses (Like Roger Wagner's Assembly Lines), but not to publish marketable games like Suitor's *Fast Life* or Thomas Mimlitch's *Spelunker*, which involved a lot of ingenious work and were yet thrown away via magazine release. All they'll get is payment for the article and for republication in a "Best Of" book, and they lose hundreds of dollars.

My game is far too long to publish in a magazine, and involves untypable file names, exit disable, auto-run mode, and starting at boot-up. Besides, typing it in from a printed page would reveal all playing secrets and spoil the fun.
Paul R. Wilson, Bergenfield, NJ

Case of the Heavy Manual Cover
Just a simple question. I know that I must clear my Apple screen before I *Load* an overlapping program.

But when I *save* an updated program (using the same old title) do I have to first delete the old program?

Half of my "experts" say, "absolutely." The other half say, "the Apple will automatically push the old program into a different sector. There will be no overlap problems."

I can't find this in any of my eight manuals, though I confess I've been too lazy to thoroughly research them.
Charles M. Larson, Whittier, CA

Saving an updated program with exactly *the same name as the old program will write over the old program. The old program will be effectively deleted automatically. You can never have two programs with precisely the same name on the same disk. Names need differ only slightly, however, to retain both programs. Naming the revise of MyProgram MyProgram 2 will render two separate files.*

Reading your manual (in this case, Applesoft Tutorial, *page 60) is always a more certain way and usually a faster way of finding an accurate answer than polling the populace. It also takes less effort.*

More to Games Than Fun
I appreciated "Psychology and the Apple" (November 1981). As a psychologist and behavior therapist I have used my Apple II to enhance treatment of youngsters in much the same way as Dr. Wanderer.

Perhaps, by showing the article to parents, it may be easier to explain how "playing games" on the Apple can be therapeutic for their children.
Harold I. Mathis, Ph.D., Southfield, MI

A Telling Time for Clocks
We would like to amplify several of Richard Kaapke's comments about the California Computer Systems Model 7424 Calendar-Clock Module. We purchased two of these clocks. The manual with the first contained clock-setting and reading programs that had to be debugged. We agree

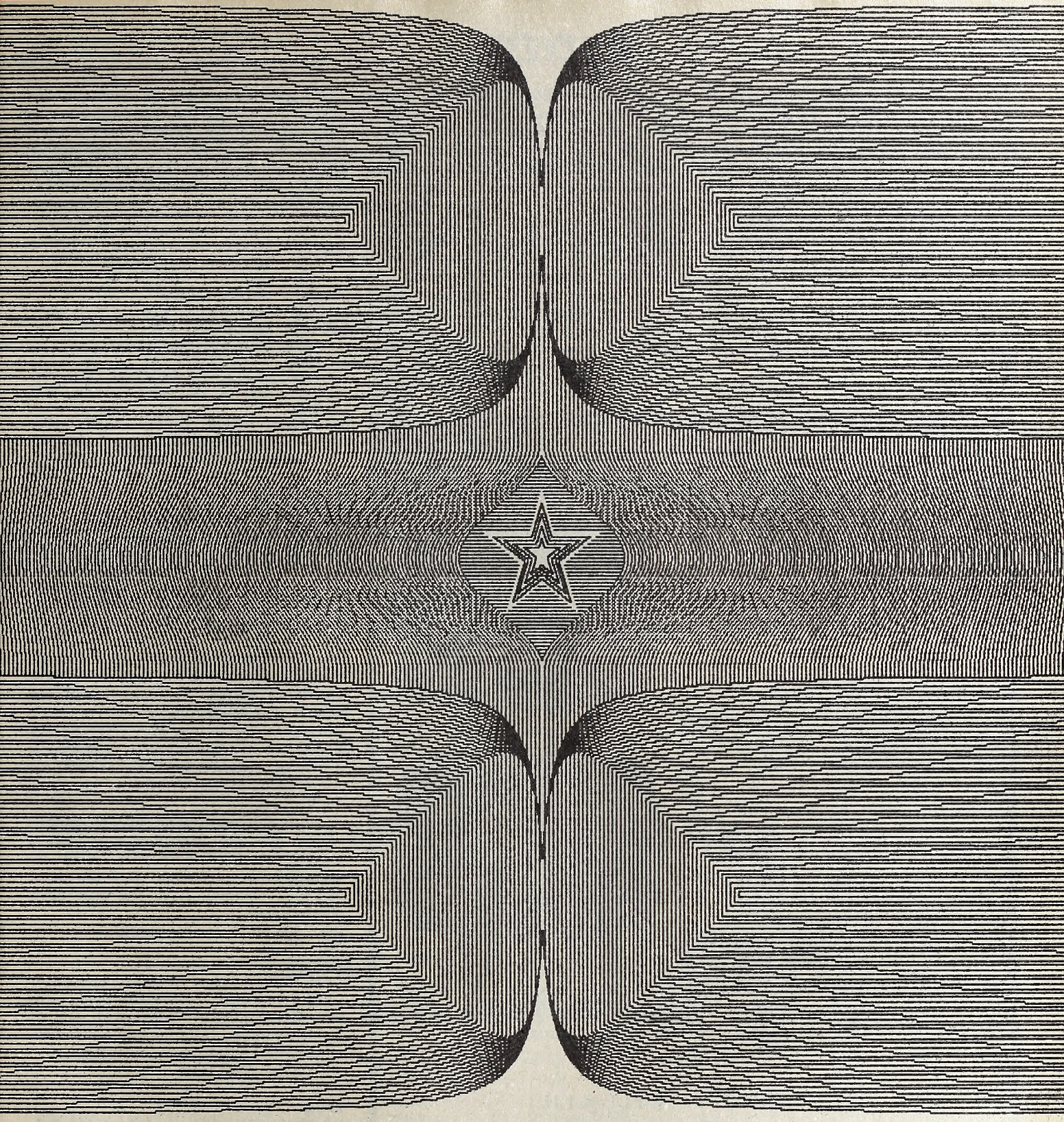


IMAGE IN IMAGINE

These 1.2 Million Pixels have been brought to you by ULTRA-RES TM using an Apple II+ and an Epson MX-100.

ULTRA-RES TM is a Trademark of Data Transforms Inc. 906 E. Fifth Ave. Denver, CO 80218 (303) 722-8774

with Mr. Kaapke that a product like the 7424 should be accompanied by a disk of tested programs, not a manual full of defective ones. The manual with the second 7424 that we purchased at a later date wasn't much of an improvement. Even "seasoned computer owners" will find it disorganized and practically impossible to follow. Our electronics technician had to examine both the circuit board and the schematic diagram even to install the batteries (not supplied). Finally, the major disadvantage of the CCS 7424 is that to change the mode of operation one has to open the Apple and move jumpers on the clock board.

Although our clocks now function properly, we would not recommend them primarily because of the poor documentation. However, for anyone who already owns this clock we would like to offer some help. Mr. Kaapke stated that in the "screen display routine . . . there is no simple way for a program to read the time off the screen. . . ." The program we offer *peeks* the memory addresses of the text screen positions to which the CCS clock writes the time. Because the screen display routine operates via interrupts, any Basic program can run simultaneously and make use of the time data so long as the Basic program does not use the same text screen positions. To use this program without modification, the appropriate jumpers should be installed for the screen display mode and for 1 Hz or slower interrupts. We found that 1 Hz interrupts only make screen scrolling a bit slower than usual, whereas 1.024 kHz interrupts interfere with disk operations.

```
 1  REM  TIME CATCHER IS DESIGNED
 2  REM  TO READ THE TIME FROM
 3  REM  SCREEN MEMORY FOR CCS
 4  REM  TIME CARD OPERATING IN
 5  REM  INTERRUPT MODE
 6  REM          BY
 7  REM  TERRY MIKITEN PHD
 8  REM  NEUROSCIENCE SOFTWARE
 9  REM          PROJECT
10  REM  UT HEALTH SCIENCE CTR
11  REM  SAN ANTONIO TEXAS
12  REM
13  HOME : VTAB 12
14  INPUT "WHAT SLOT IS THE CLOCK IN? ";SL
15  CALL 49360+ (256 * SL): REM TURN DISPLAY ON
16  FOR I = 1063 TO 1052 STEP- 1:Z$ = CHR$
    (PEEK(1)) + Z$
17  PRINT Z$
18  REM
19  REM  TURN OFF DISPLAY WITH:
20  REM  POKE49395+(256*SLOT #)
```

A. P. Shepherd and T. M. Mikiten, Associate Professors, University of Texas, San Antonio, TX

Master Plan for Modem

I have just finished reading my copy of October's *Softalk* and was completely taken with the article on "The GameMaster." That unique service is the best argument I've heard yet for investing in a modem. It is articles like that that keep me reading *Softalk* from cover to cover each month.

I would very much like to contact Harlow Stevens, Bob Kniskern, and/or Paul Martin and learn more about the GameMaster philosophy and the possibility of bringing it to the east coast where I live.

Paul Mershon, Irvington, NJ

Poser for Symposiumites

I am an avid fan of Greg Tibbetts's SoftCard Symposium. I have an Apple II, two disk IIs, SoftCard, Videx card, and an Apple Silentype printer—and a question.

As you can see from the directory, I have a file BH3.BAS.

```
A>B:
B>DIR
B: MBASIC  COM : BH2       BAS : CLEAR  BAS
 : HLINE    BAS
B: BH2-P   BAS : CURSOR    BAS : VLINE  BAS
 : SUBROUTE BAS
B: BH0     BAS : BH3       BAS
```

Without the inconvenience of going into MBasic (change *print* to *lprint*), can you tell me how I can directly from the CP/M directory run the file BH3 over Monitor and/or printer?

Hans J. Baerwolf, Inglewood, CA

More on the Amazing Epson

In John Butler's letter to the editor (September 1981), the question came up, how do you get Epson's printer code (escape-

E) to work on the *Magic Window*? I suspect that this problem may arise with other printers and other word processors. The key is simple: use a *capital E*, represented in *Magic Window* by an inverse letter. The key sequence for the example (assuming you are not in caps lock) is:

CTRL-B	ESC	ESC	E	cantinue typing....
a ctrl	this	capital		
char	is the	letter		
follows	char	fallaws		

This shift key also explains why the characters @ [/] < _ come out as ') !) $ on the printout; the first bunch were shifted and appear on the screen in inverse.
Tim Desmond, Harbor City, CA

To Michael Gibson, Roy Trahan, and Martin Tiersky: **THANKS!** While I don't have a *Magic Window* with my Apple II, I do have the *Apple Pie* Text Editor. Up to now, I've not found a way to send an escape sequence to the printer. My letters to Programma brought back a rather cryptic reply of *" . . . type the standard series followed by ESC."* I was unable to decipher this code until I saw the letters of the above gentlemen in reply to someone else's problem. The same fix (entering an escape code sequence as control-escape) works with my *Pie*. The keypresses are: Shift-control-M, then escape, then the number or letter for the code (the letters *must* be uppercase).

By the way, with the new *Graftrax-80* option on my MX-80, I can now mix compressed, emphasized, double-print, and expanded print on the same line, mixing them at will, and cancelling any at any time (with the exception of the expanded characters; they still turn off at the end of the line).

Not only that, but I now have an italics mode that can add extra emphasis to any printed matter. Then add the new ability to backspace, and underlining becomes a reality at last.

There are still more advantages to the Graftrax-80 that I'll not go into now, but suffice it to say that this is not *just* a set of PROMs to produce hard-copy graphics (which it also does quite well—thank you, Epson).

Finally, thanks for a great magazine, and I look forward to your newest venture, *Softline*. It should prove interesting to the adventurer in all of us.
Steve Nelson, Arlington, TX

The Oft-Overlooked Integer Variable

Your September 1981 Basic Solution for inputting cursor control commands in a clear uniform manner (and also other commands as required by the program) uses excessive code to accomplish this task, mainly because the subroutine does not use the two most powerful programming tools available in Basic, the *array* and the *for-next* loop. Also, since real numbers are used, rather than integers, the subroutine is executing slower than it would if integers were used.

The first step in perfecting this routine, then, is to change all the real number variables to integer variables. Thus line 110 would be changed to read CM%=1 : GOSUB 1000 and lines 1010 and 1020 would use C% rather than C to store the input from the keyboard.

Secondly, the character codes that will be considered valid are rearranged and read into an integer array during program initialization in the following manner:

Character Codes	Return Codes
27 Escape	1
08 left arraw	2
21 right arraw	3
13 return	4
47 slash	5
— string	6

```
10 DATA 27,8,21,13,47
20 FOR Z= 1 TO 5 : READ CC%(Z) : NEXT
```

The codes are now in array CC%. Entry one contains a 27, entry two an 8, etc. The subroutine is modified by replacing the lines 1030 through 1080 with the following:

```
1030 FOR RV = 1 TO 5 : IF C% = CC%(RV) THEN
     1090
1040 NEXT : IF NOT CM% THEN INPUT ""; I$ :
     RETURN
```

The subroutine will now read array CC% and if it finds a match to the value in C% then a GOTO 1090 is executed where I$ is set to the character represented by the ASCII value in C%. When the return is

executed, the array entry number is set in the variable RV and the ON RV GOTO instruction processes normally.

Now, all this work will save approximately 45.73 microseconds (this figure is accurate to +/− 10 microseconds), but more importantly, program storage is reduced. No matter how many different codes you want the program to check, the same instructions will do the job. The only exception is if more than ten character codes are used, then a *dim* statement is required to dimension array CC%.

Jack Van Zandt, Palmdale, CA

Remedy for Absent-Minded Renumberer

The *Renumber* utility program supplied with Apple DOS 3.2 and 3.3 has an insidious bug—after you use *Renumber*, your program may still appear to run perfectly, so you may not even notice that your program's operations have been altered! *Renumber* will correctly change all line number references to agree with the new line numbers. Unfortunately, *Renumber* may also alter any number in an arithmetic expression that (a) follows an asterisk (the multiply operator), and (b) has the same value as a pre-*Renumber* line number.

I obtained the corrections for the DOS 3.2 version from the Apple Hotline in May of 1980. I just discovered that the problem still exists in the DOS 3.3 version, and I am still seeing letters in various magazines from perplexed Apple users.

The fixes for the DOS 3.2 and DOS 3.3 versions of the program are similar—they involve swapping two data values in the program, as follows:

	DOS 3.2	DOS 3.3
From Basic:	POKE 4815, 172 : POKE 4816, 171	POKE 4789, 172 : POKE 4790, 171
or		
From Monitor:	* 12CF: AC AB	* 12B5: AC AB

To correct the *Renumber* program permanently, you must a) load *Renumber*. b) Do the two pokes for your version of DOS. c) Save *Renumber*.

All Apple owners should take note of these fixes—even if you don't use *Renumber*, you'll be able to help out the next guy, who may not have read about this problem! For your future reference, Apple dealers have a loose-leaf notebook that answers commonly asked questions—including "What's wrong with *Renumber*?"—you just have to know to ask!

P.S. The latest issue of Apple Orchard indicates that the two locations to be POKE'd for RAM Applesoft *Renumber* are 14342 and 14343.

Robert C. Leedom, Glenwood, MD

Renovating the Dumps

Art Christopher's Screen Dump Program (November 1981, page 15) works very well. I have made a few changes that should be interesting.

First, because of the intertwined pattern in which the screen is stored, he used three *for-next* loops. Since the loops are values that are exactly forty apart, a single loop can be nested within another.

Second, the *gosub* was needed to avoid repeating the code for printing one screen line within each of the three loops. Since there is now one nested loop, the code can be put in the middle of the loop and the *gosub* is eliminated.

Third, because the *gosub* and the branch around it are eliminated, the entire routine can be entered on one line number. All three of these changes add to the efficiency of the routine.

A fourth improvement that, for clarity, I did not choose to include is to eliminate the variable name after each *next*.

```
    1  REM EPSON SCREEN DUMP
    2  REM BY MARK LAVETTER
    3  REM 11-5-81
    4  :
    5  REM BASED ON
    6  REM SCREEN DUMP PROGRAM
    7  REM BY ART CHRISTOPHER
    8  REM LETTER TO
    9  REM SOFTALK 11/81 P. 15
   10  :
63999  PR# 1:
       PRINT CHR$ (9) "80N":
       FOR H = 0 TO 80 STEP 40:
         FOR I = H + 1024 TO H + 1920 STEP
             128:
           FOR J = I TO I + 39:
           A = PEEK (J):
           PRINT CHR$ (A);:
           NEXT J:
       PRINT CHR$ (13);:
         NEXT I:
         NEXT H:
         PR# 0
```

The routine just listed should work for most printers. For printers that can handle 132 character lines (Epson MX-80 with condensed print, or Epson MX-100, for example) the following routine will print the screen dump three-up. This takes advantage of the interleaved organization of the lines in memory so that the first eight lines on the screen print on the left, the middle eight lines in the middle, and the last eight lines on the right.

```
    1  REM EPSON SCREEN DUMP 3-UP
    2  REM BY MARK LAVETTER
    3  REM 11-5-81
    4  :
    5  REM BASED ON
    6  REM SCREEN DUMP PROGRAM
    7  REM BY ART CHRISTOPHER
    8  REM LETTER TO
    9  REM SOFTALK 11/81 P. 15
   10  :
   11  REM PRINT WIDTH = 132
   12  REM EPSON MX-80 AND MX-100
   13  REM CONDENSED PRINT
   14  :
   15  REM FOR MX-100 NORMAL PRINT
   16  REM REMOVE PRINT CHR$(15)
   17  :
63999  PR# 1:
       PRINT CHR$ (9)"132N":
       PRINT CHR$ (15):
       FOR I = 1024 TO 1920 STEP 128:
         FOR K = I TO I + 80 STEP 40:
           FOR J = K TO K + 39:
           A = PEEK (J):
           PRINT CHR$ (A);:
           NEXT J:
         PRINT " ";:
         NEXT K:
       PRINT CHR$ (13);:
       NEXT I:
       PR# 0
```

As with Art's routine, the addition of :RETURN at the end makes this a subroutine which can be used from within other programs.
Mark Lavetter, Southfield, MI

I wish to express my thanks for a most informative magazine. I am a new adventurer to the world of computing and subscribe to various publications but yours is by far my favorite.

I have a question regarding Art Christopher's Screen Dump Program. Art stated that it could be called with a control-P; would you please clarify this on just how it could be written into a program?
Dr. Joseph D. Scalzo, Scottsdale, AZ

Mr. Christopher's screen dump program is intended to be used as a subroutine. It can be called from any input *or* get *statement in the main program. Assuming you want to use control-P to trigger the screen dump, the calling statement might look like this: If A$=CHR$(16) THEN GOSUB 33000. Naturally A$ in this statement could as easily have been X$ or any other string variable. The use of control-P is also arbitrary, although the use of a control character has the merit of not echoing the trigger character on the screen following an* input *statement.*

Toward Easy Conversion

I have been reading *Softalk* for the past few years now and I must say that it has come a long way in that time. I have found that among the most helpful articles in *Softalk* are Basic Solution and Assembly Lines. One difficulty some of us have is converting hex numbers to decimal and decimal to hex. To that end, I submit a program, which I think will help my fellow Apple users.

Here is how it works:

```
 10  REM **THIS PROGRAM IS DESIGNED TO
     CONVERT HEX ADDRESSES TO DECIMAL OR
     DECIMAL TO HEX .
 20  DIM A%(4,1),D$(4,1)
 30  CLEAR
 40  PRINT "ENTER INPUT NUMBER FORMAT 'H' =
     HEX, 'D' = DECIMAL?": GET O$
 50  IF O$ = "D" THEN GOTO 220
 60  IF ASC (O$) = 13 THEN PRINT "BYE": END
 70  IF O$ < > "H" THEN PRINT "INVALID OPTION
     ENTER 'H' OR 'D'": GOTO 40
 80  PRINT : INPUT "ENTER HEX NUMBER TO BE
     CONVERTED? ";A$
 90  L = LEN (A$): IF L = 0 OR L > 4 THEN PRINT
     "INVALID ENTRY, REENTER": GOTO 30
100  FOR J = 1 TO L:N = L - (J - 1)
110  FOR I = 0 TO 15
120  READ B$
130  IF B$ = MID$ (A$,N,1) THEN A%(J,1) = I:
     GOTO 170
140  NEXT I
150  PRINT "AN ILLEGAL CHARACTER WAS
     DETECTED IN THE INPUT STRING. IT IS -" MID$
     (A$,J,1): RESTORE : GOTO 30
160  GOTO 30
170  RESTORE : NEXT J
180  D = A%(1,1) + (16 * (A%(2,1) ) ) + (256 *
     (A%(3,1) ) ) + ( (16*16*16) * (A%(4,1) ))
190  PRINT "THE DECIMAL EQUIVALENT OF"A$" ' IS
     "D
200  PRINT
210  GOTO 30
220  PRINT : INPUT "ENTER DEC NUMBER TO BE
     CONVERTED? ";R
230  IF R = 0 OR R > 65535 THEN PRINT "INVALID
     DECIMAL ENTRY!": GOTO 220
240  C = R
250  A%(1,1) = R / 4096
260  R = R - (4096 * (A%(1,1) ) )
270  A%(2,1) = R / 256
280   R = R - (256 * (A%(2,1) ) )
290  A%(3,1) = R / 16
300  A%(4,1) = R - (16 * (A%(3,1) ) )
310  FOR J = 1 TO 4
320  FOR I = 0 TO 15
330  READ B$
340  IF I = A%(J,1) THEN D$(J,1) = B$: GOTO 370
350  NEXT I
360  PRINT "DECIMAL NUMBER OUT OF RANGE FOR
     4 HEX BYTES": GOTO 30
370  RESTORE : NEXT J
380  E$ = D$(1,1) + D$(2,1) + D$(3,1) + D$(4,1)
390  PRINT : PRINT "THEN HEX EQUIVALENT OF
     '"C"' IS "E$
400  PRINT
410  GOTO 30
420  DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
430  REM ***** THIS PROGRAM DONATED BY
     RICHARD A. PATTON, 362 HASTINGS BLVD.,
     BROOMALL, PA 19008
```

Lines 40 to 70 are used to select and validate the option "H" or "D."

Line 80 requests the hex address as a string variable.

Line 90 validates the input.

Line 100 establishes the for-next loop used to check each input character, the N variable is created to invert the J sequence since the *mid$* step in line 130 reads characters from left to right (left justification for characters).

The for-next loop of lines 110 to 140 is used to check each input character against the data statement in line 420. Then use the value of I (which corresponds to the match of "B$" to input character J) to load the array A%,J.

Line 150 detects an illegal character.

Line 170 resets the data pointer to entry #1.

Line 180 is the heart of the hex-dec conversion; it raises each variable in the array A% to the appropriate power of sixteen and then sums the results into variable D.

Line 190 prints the results.

Line 220 is used when dec to hex conversion is requested. The input is stored as a real number since Apple will not accept an integer number larger than 32767 and $FFFF is 65535 in decimal.

Line 230 validates the input.

Line 240 saves the input in variable C for later display.

Lines 250 to 300 convert the input "R" into an equivalent integer variable A%. Each A% variable is loaded with the appropriate number of times that the power of 16 (4096, 256, 16, 1) will divide into R, the input. Each time a division occurs, the product of sixteen raised to the appropriate power and the integer value A%

is subtracted from the input variable R. Thus the remainder is passed on to the next lower power of sixteen for division.

Line 310 starts the for-next loop that will load the output variables.

Lines 320 and 350 match the integer loaded in A%,J against I; when a match is found the corresponding B$ variable represents the hex equivalent of the integer A%,J. The appropriate B$ variable is then loaded into the string array D$,J.

Line 360 is printed when no hex match can be found.

Line 370 restores the data pointer as above.

Line 380 is used to concatenate all of the D$ variables into the output variable E$.

Line 390 prints the result of the conversion.

Line 420 is the data statement that represents all the possible hex designations.

I hope you will accept this as my thanks for the good work you folks are doing.
Richard A. Patton, Broomall, PA

From Each According to His Ability, to Each According to His Whim
Let's End The War

Right now the Apple users are in a state of confusion.

Looking at all the new software that is on the market, with the boom in home computers, and seeing the prices on programs makes us wonder what's going on.

We have to make a decision between saving our hard earned cash and paying the ridiculous high prices for these programs or being illegal and getting black market programs at a more affordable price; neither is a good choice.

I have noticed in recent ads in other computer magazines that certain companies have more or less declared war on those that are making these copies.

Granted software companies should make a profit on the work and effort of what they have done. But how much of a profit and for how long? That's what it gets down to.

Most companies need at the most 5 percent profit to put back into the company. The prices we see have to be more than that, figuring they can't be paying over $2 for the diskette, and that leaves the remainder for the printed matter, which can't be more than $5 (basing this on the price of similar material on the market), which comes to a total of $7. Most software listed in *Softalk* sells from $26.95 on up.

If these manufacturers would bring the price down, these so-called pirates couldn't afford to buy the copy programs, take the time to copy, and still figure that they are coming out ahead.

I am not for copying copyrighted programs because these companies have earned the rights to them and it is against the law. However if these same companies would wake up to the fact that they could all but eliminate these pirates by cutting their prices (in some cases by more than half), it would make a simple compromise.
Steven Straughn, Omaha, NE

A Better Way To Trace Roots

I want to add my congratulations on the content of *Softalk,* which covers a wide range of interests and abilities in the personal computing field. My own interest is in genealogy and you might want to consider an article on the available programs and the various uses that are made of them. While I can't claim that genealogy is the leading use for small computers, it certainly has great potential.

The average person interested in genealogy is not a computer or programming expert *but* is vitally interested in storing and sorting data. Since cost is a factor, I have found a great reluctance to invest in genealogy programs that are advertised, since many "horror stories" are circulated, and most computer stores do not have sample disks to demonstrate to the neophyte what a particular program can do.

Since I started some time ago, I am using *File Cabinet* on a 48K, 13 sector, 3.2 system, and many others also use either *File Cabinet* or *Data Factory* as a method of storing the data. I have heavily modified my *File Cabinet* but still it is not a genealogy program and of course moving data from disk to disk is still not a simple matter, though I have programs to help a little. I am not in the "expert programmer" category and need all the help I can get. Some of us do exchange ideas of course, but I have not found any real national source of ideas or programs . . . a mention in *Softalk* of the problem might uncover a solution.

I modified the Sklar program for finding dates (October 1981) and can now figure dates forward or backward, compute birth dates from the common death info found in genealogy, etc. I appreciate his efforts and yours in publishing the basic program.

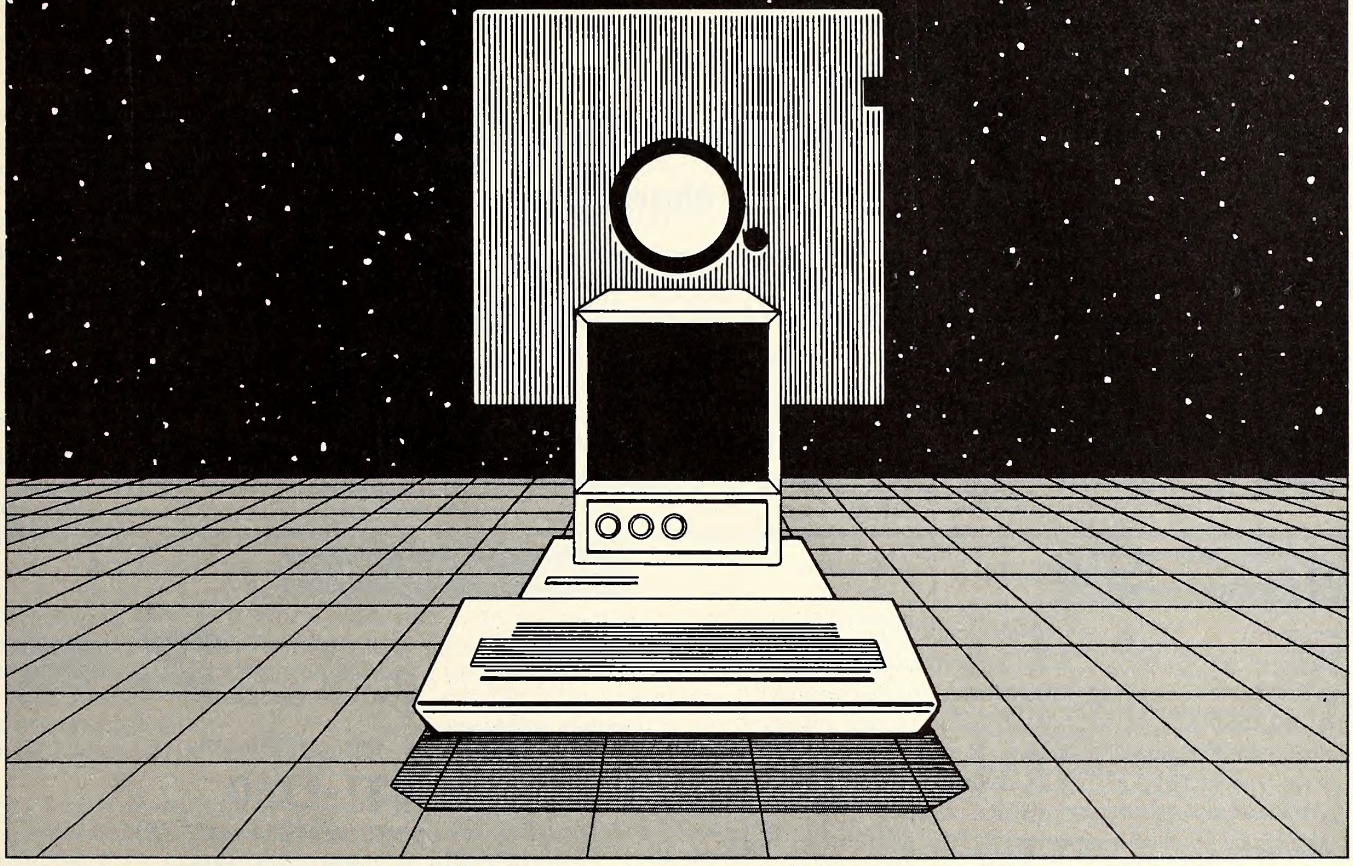
I would like to use a time-share system to store and sort my files . . . more than four thousand names and associated data . . . and tried to use the Source but that was totally unsatisfactory and I have not yet found any other that seemed reasonable.

Any genealogy help that is forthcoming will be appreciated by many people.
Jim Grinnell, Chicago, IL

Seeking a Good Connection

Since the first issue I have been extremely impressed by your magazine and credit it for making the entry into the world of the Apple Computer much easier. Your different tutorials have been especially valuable. I have come up with a problem, however, that I hope one of the

Professional proofreading on the APPLE II Computer is now a reality



THE APPLE SPELLER fills the void that has consistently kept the large variety of excellent word processing packages for the Apple II Computer from approaching the power of a dedicated word processor. Finally, the first professional quality spelling verification program is available for the Apple II. The Apple Speller will certainly be the standard against which all other similar programs are compared.

The Apple Speller interfaces to the most popular Apple word processors including Applewriter, Apple Pie, Superscribe II, and Magic Window just to name a few. In fact, the Apple Speller can analyze the output of any editor that writes a standard Apple binary or text file to a diskette. In addition to this flexibility, the performance of the Apple Speller will astound the microcomputer world.

The Apple Speller is supplied with a 30,000 + word dictionary on a single 5¼" diskette with additional space to easily add another 8,000 words to suit your individual needs. The Apple Speller has built-in utilities to maintain the dictionary diskette. You can readily add words, delete words, and create an unlimited number of modified and/or new dictionaries for specific applications.

The Apple Speller is unbelievably fast. The first pass reads your document and collects all the words it contains at a rate of 5,000 words per minute. Next, the words are compared to the dictionary for spelling errors at the incredible speed of 50,000 words per minute. Finally, all misspelled words are marked as such in your document with a rate of 1,000 words per minute. This translates to proofreading a 10 page document in 1 minute if there are no spelling mistakes and 2 minutes, 15 seconds for an unlimited number of spelling errors.

Numerous options are provided throughout the program to enable you to completely control all activities of the Apple Speller. These include the ability to ignore both control codes and formatting commands, an alphabetical listing of either misspelled words or all the words in your document along with usage frequencies, multiple options for the action taken with each misspelled word, and much, much more! A verification mode is provided to allow you to examine and dispense with misspelled words while viewing them in the actual context in which they appeared in your file.

The Apple Speller requires an Apple II/Apple II + equipped with 48K, Dos 3.3, and two disk drives.

The Apple Speller is being introduced at the incredible price of **$75.00!**

SENSIBLE SOFTWARE
6619 Perham Drive Dept. M
West Bloomfield, Michigan 48033
(313) 399-8877
Visa and Mastercard Welcome
Please add $1.25 postage and handling per diskette

*Apple T.M. of Apple Computer Inc.

Take Charge. . .

...of your growing collection of disk-based software.

Introducing:

Disk Library®

The Software Management System

Disk Library is an elegant, user-oriented system for creating and maintaining a thorough, cross-referenced Index of all your disk-based programs and data files. It provides for Automatic entry into your Library file of the full catalog of any Apple* diskette. Disks formatted under other operating systems (such as Pascal and CP/M*) are easily entered from the keyboard. Written entirely in machine code, Disk Library's operation is both smooth and swift.

EASY TO OPERATE:

- Menu-driven; • User-definable prompt defaults
- Single keystroke operation • Full featured **Editing**
- Super fast **Sorts** by any field (1200 items sorted in 4 seconds!)
- Works with all disks created under DOS 3.1, 3.2 and 3.3
- User definable **Program Types** (e.g., Business, Game, Utility) of up to 15 characters each can be assigned to each program entry with single keystrokes or via block actions
- On-screen and printed **Summaries,** by File Type (Integer, Applesoft, Binary, Text) and by Program Type (e.g., Accounting, Graphics, Music)
- **Block Actions** (global editing/deleting)
- Instant **Searches...**by full or partial string (find any item in 1/3 sec.!)
- New Files can be **Appended** to existing records, in memory or on disk
- **Unique Feature:** User can redefine the Disk Volume Number displayed by the DOS Catalog Command
- A Unique Volume Identifier and Disk Title can be Assigned to each disk entry in your library file.
- Printed **Reports** are attractively formatted for easy readability

EASY TO LEARN:

A 75 PAGE, PROFESSIONALLY PREPARED USER'S GUIDE IS PROVIDED: INCLUDING:

- Introductory Tutorial, will have you using Disk Library in 10 minutes
- Advanced Tutorial, enables you to master **Disk Library's** many advanced features
- Reference Section, provides quick answers for experienced users
- Applications Section, gives you many ideas for maintaining your library
- Index, enables you to find whatever you need

EASY TO ORDER:

Use your Visa or MasterCard ANYDAY, ANYTIME

VISA MasterCard

TOLL FREE: (For orders only)

1-800-331-1750

Ask for operator 948, Dept. 100

(In Okla.): 1-800-722-3600

Ask for operator 948

Or send check or money order to: MODULAR MEDIA

Florida residents add 4% sales tax

PRICE: $39.95

(Special Introductory offer until Feb. 28, 1982)

$49.95 (Regular Price).

WHAT YOU NEED:

- A 48K Apple II or II+ with DOS 3.3
- A desire to get organized!

MODULAR MEDIA

11060 Paradela St., Miami, FL 33156 (305) 661-7310

Developers/Publishers of Innovative Software

* Apple, Apple II and Apple II+ are registered trademarks of Apple Computer, Inc. CP/M is a registered trademark of Digital Research, Inc.

other readers has a solution to. The problem relates to the use, or, in this case, the nonuse of an Epson printer with Apple Pascal 1.1 with a nonApple interface card. I have been unable to get the printer to print using any of the procedures outlined in the Pascal manuals. I have been using an Epson parallel card while a friend has been using an AIO card, both with little success. Any assistance will be appreciated.
Jim Burke, Iowa City, IA

What Is Obscene, after All?
Softalk is a useful tool for those interested in programming. However, I feel that some of the ads are completely inappropriate for such a magazine. I am referring to the advertisements for erotic software and pornographic games. I find it disgusting and do *not* want to see anything of the sort in *Softalk*. In addition to agreeing with Ms. Smith's objection (in the November *Softalk*) that one such ad was degrading to women, I think such ads are harmful to the industry. What does one say to the parents who see topless women in their child's magazine? I do not wish to preach from the pulpit, but a prostitution simulation game is a moral outrage. What people do is their business, *as long as it is not intruding upon the freedom of others*. If others wish to engage in such activities, then they have made that choice. But being assaulted by this filth merely because one turns a page is unacceptable.
Michael Daugherty, Fort Collins, CO

The sexual mores of a society are always under debate. That very statement implies that there are two sides to the issue, neither side more morally correct than the other, both only seeking for some higher truth. In our society, the Supreme Court has defined pornography as those portrayals of human sexual activity that have no socially redeeming value; Softalk *has seen some programs like that and has rejected ads for them. The programs for which we have accepted ads in this context are those that we believe to be legal under the guidelines of this country.*

In general, it's been a pitiably small minority that have raised their voices to object to these ads. In fact, more than three times as many persons have called with support for carrying those kinds of ads as have protested. But even more cogent is the point that nothing, not even the pure science of computer programming, exists external to the society within which it functions. And that society will intrude and force its own reality upon its subgroups.

Thank you for printing the objections to cheesecake ads (November). I would like to add my agreement to the opinions of Gillett and Smith, and thank them for raising the issue. *Softalk* doesn't need this sort of material and advertisers need to know it. (Sex is wonderful, but if I want to look at nude people I'll subscribe to *Playboy*.)

I would add the ad for *Hi-Res Secrets* (November, page 39) to the list of undesirables. Please ask Don Fudge if he really wants his name on such cheap sleeze.

Please continue your high-density coverage of the Apple. You have been a real help.
John L. Zimmer, M.D., Kalamazoo, MI

Softalk Gaffes—All in One Spot
Basic Solutions? Good idea. But they are not solutions at all when they are full of errors. Take December for example (this is the first one I've tried to implement). You'd do us (and yourself) a favor by printing actual computer listings (that work) instead of a typeset version of same. If a "solution" must be typeset for publication, try keying the program from the proof and make sure it works.
David Fellman, Rochester, MN

Bugs in the Forest
After reading the November 1981 article entitled "Apples Among the Acorns," I felt compelled to write you concerning a few inaccurate statements.

In the section entitled Apple's Instant Recall Handles Logistics, the very first paragraph speaks of a program feature that deploys helicopters to incidents by providing a heading (in degrees) and a distance (in miles). This aspect is available *only* to our air tankers based at Fox Field Airport, in Lancaster, California. The distance and heading is calculated by the Apple from the Palmdale VOR (Variable Omni Range), to the center of the section (a 640-acre area), where the fire is located.

In paragraph six of the same section, Matching Specific Fires to Methods that Worked Historically, you incorrectly mention that the main information center (for another computer we have access to) is in Atlanta, Georgia. This should be corrected to read Riverside, California. I might mention that this computer in Riverside, labeled the PRIME, ties together a multitude of cooperating fire agencies in a system called FIRESCOPE (FIrefighting REsources of Southern California Organized for Potential Emergencies).

Finally, in the section Margin for Error, second paragraph, the Marble Cone Fire occurred in 1977 and burned 177,000 acres, not, as the article stated, 1979 and 130,000 acres burned.

On the positive side, I would like to take this opportunity to express our appreciation for the article and your interest in our program. The response from the Forest readers has been good and we will continue to look forward to reading your magazine each month.
Michael Bergdahl, Angeles National Forest Dispatcher, Pasadena, CA ◨

Apple president Mike Markkula predicted in the December Softalk *that the personal computer would equal the printing press in its impact on society. Should Markkula prove prescient in that regard, then it's fitting that magazines that treat the use of the personal computer also address the larger issues.*

This series of Newspeak articles will attempt to air the issues confronting our civilization. Because of that orientation, it will contain no inside information on peeks and pokes, bits and bytes, or RAMs and ROMs.

The issues are far less concrete and to that extent far more discomforting than the technical aspects of how to make your Apple do your personal bidding. But it's not realistic in this era of such rapid change to use technology as a shield from the real world. The real world will impinge on the ethics, morality, and pragmatics of pure science and technology, even as our growing technological expertise impinges on the real world.

Because this is true, the interaction with the civilization at large of those members of civilization who use such advanced tools as the personal computer is as much a proper subject for a magazine like Softalk *as delivering tutorials on specific languages or specific applications programs.*

Not all the articles in this series will be as wide-ranging as this one. In his eighty-six years, R. Buckminster Fuller has developed a thoughtful, coherent, logically complete cosmology for the universe and the planet he affectionately calls Spaceship Earth. He limits his consideration of infrastructures to those whose policies, if continued, represent the forces of opposition to the changes he feels are necessary.

We did not so much interview Buckminster Fuller as we did attend a personal lecture. Fuller's hearing impairment makes the intercouse usually implied by the word interview *difficult.*

Another consideration is that Fuller is a man feeling deadline pressure. He feels human civilization is fast approaching a crisis point and his interest is in disseminating the philosophy he feels the human race need adopt to salvage itself. With such

UTOPIA OR A VISIT WITH

A NEWSPEAK FEATURE

cosmic issues under consideration, directing Fuller's attention to more mundane topics seemed inappropriate.

The pursuit of Truth, however, is never inappropriate. If Markkula is correct, it's mandatory for Softalk*'s readers to give thoughtful consideration to the alternate futures available. Perhaps few will find Truth or an appealing alternative future in the words of R. Buckminster Fuller. But all will be better equipped to evaluate other alternatives and other philosophies once they've grasped Fuller's views.*

> *Whether it is to be Utopia or Oblivion will be a touch-and-go relay race . . . Humanity is in "final exam"*
>
> —Buckminster Fuller, *Critical Path*

You may remember the Right Reverend Thomas Malthus. It was his postulation that the earth's food resources grow arithmetically while its population increases geometrically, implying inevitable food shortages and worldwide starvation.

That awful realization was what caused David Hume to dub economics the "dismal science."

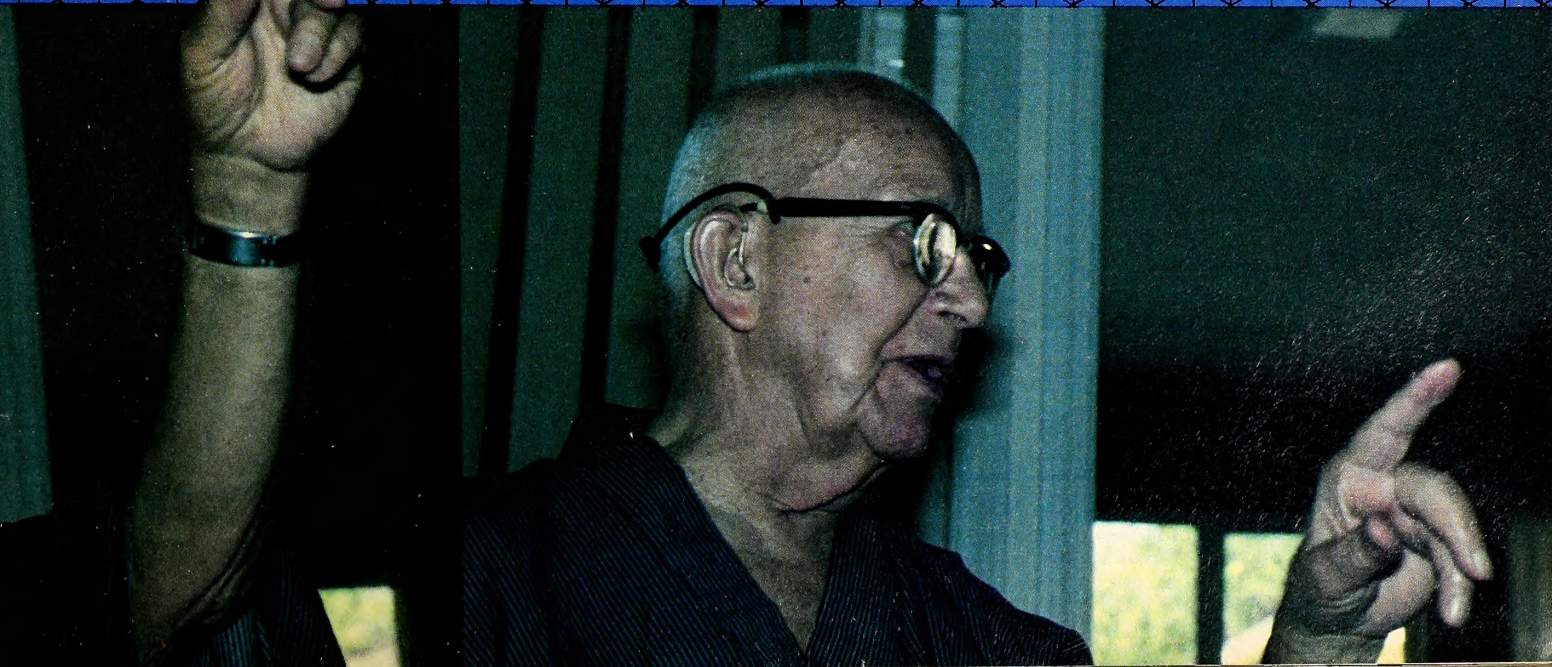
For a short time in this century, Malthus's theory became the object of scorn and derision. Technology would overcome all such seemingly determinate mathematics.

When it later became apparent that the population was not only acting in the manner observed by Malthus, but it was also rapidly converting fertile agricultural lands to urban home sites, his theory again gained a degree of credibility. It seemed not even technology could overthrow the inexorable fact of scarcity.

Advocates of Scarcity. But that may not be the fact. Richard Buckminster Fuller, mechanic, architect, and philosopher, believes that the entrenched interests in most advanced countries embrace the philosophy of scarcity and use it as the raison d'etre for activities that, in his opinion, hardly qualify as being in mankind's best interests.

OBLIVION?
BUCKMINSTER FULLER

BY ALLAN TOMMERVIK



What's most frustrating about these activities, which Fuller tends to subsume under the label "political," is, he believes, that the fundamental underlying principle of scarcity is no longer operative.

About ten years ago, mankind turned the corner and technology became available that, properly applied, had the power to repeal the law of scarce resources. That the organized bodies of society have failed to do so is prima facie evidence of their vested interest in a status quo that imbues value to goods and services viewed as scarce. After all, the law of supply and demand only functions in an environment where there's no equilibrium between the elements.

Much of this new technology that makes it possible to dream of a higher standard of living for all men came from research into weaponry. Now Fuller believes it's time to break down our weaponry and build "livingry"—the modern equivalent of the Biblical injunction to beat swords into plowshares.

Biblical metaphors are not out of place when discussing Buckminster Fuller's ideas. He has a firm belief in a higher intelligence, which he is reluctant to call God and drag into the realm of religious controversy; and this belief and his overall view of the universe are a necessary departure point for understanding Fuller's philosophy.

The Ultimate Design. In conversation, Fuller alludes to the American Indian belief in a Great Spirit as being analogous to the higher intelligence he perceives. He derives what he believes to be an experiential proof of the existence of this being from the universal physical laws that find their expression in mathematics. These higher laws are distinct from that body of natural phenomena where the exception proves the rule in that there are no exceptions. They coexist and interact and augment each other at times, but never do they conflict.

An example of such a law, from the field of astronomy, is the interattraction of celestial bodies, which always varies inversely as the second power of the arithmetical distances intervening.

In essence, Fuller sees design, in contradistinction to randomness, in such laws and from that design deduces a galactic intellect greater than our own.

Design is important to a man of architectural skills such as Fuller, who cites various levels of design skill. A house can be designed with 500 drawings and requires building tolerances to one thirty-second of an inch. An automobile requires 5000 drawings and requires tolerances of one ten-thousandth of an inch, which is finer than the human eye can detect.

A Boeing 747 requires 50,000 drawings. Because air resistance increases as the second power of the speed of the craft, the 747, traveling at 650 miles per hour, must be able to withstand air pressure one hundred times greater than the velocity of a hurricane. For such a task, it requires better engineering design in each nut and bolt than in all the automotive industry.

Yet all this pales in comparison with the design Fuller sees in the universe.

Caretakers of the Mind. Another manifestation of this greater intellect's design is that the universe is self-regenerative. Everything in the universe is energy, either in the form of matter or in the form of radiation. No energy is ever lost, nor is any ever added, to the universe. Instead, the universe regenerates itself.

Fuller depicts mankind's role in this universe is as a husbandryman for the small corner of it that we inhabit. It is man's role to serve as a local information gatherer in this sector of the universe, detecting and solving local problems accommodating evolutionary change.

Humanity needs to produce technology to address those problems it finds. But, as the human becomes more adept at problem solving, it doesn't mean that all the problems are solved; instead, bigger problems manifest themselves and beg for solution.

Fuller believes mankind was nominated as caretaker for this section of the universe because of the presence in humans of mind as well as brain.

Brain, which humans have in common with other animals, is that part of the intellect that deals with the senses. The brain processes data on what we see, hear, smell, and feel. To that extent, the brain is limited to experiential data.

On the other hand, mind comprehends data beyond the senses. It's mind that allows man to recognize principles and universal laws that control the data the brain recognizes. It's mind that makes the great leap from the concrete to the abstract. It's mind that allowed astronomers from Copernicus to Galileo to make valid assumptions about our universe even though they lacked the means to objectively verify their hypotheses.

Fuller emphasizes the greater importance that mind plays in the understanding of reality in this day and age.

When he was born in 1895, reality was measured essentially by the senses. What you could see, hear, smell, and feel constituted reality. But in that same year, Guglielmo Marconi invented the wireless. Harnessed radio waves were a phenomenon that could not be sensed objectively; only the results of their existence could be known.

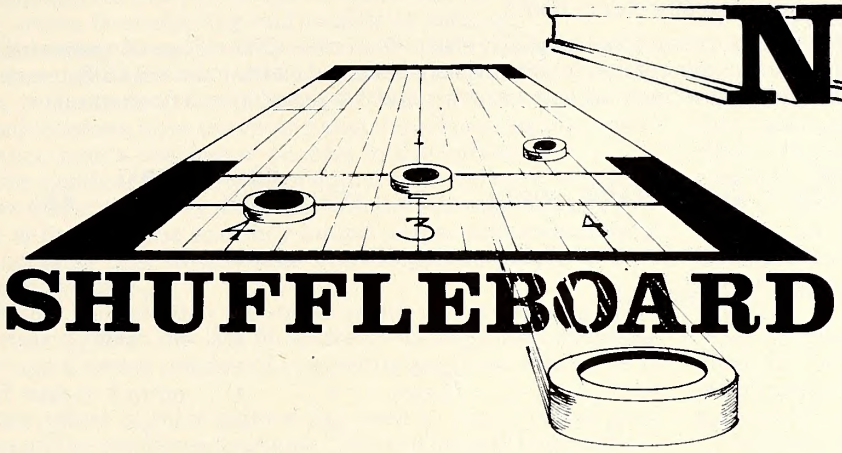
Such subsequent discoveries as the electron and the development of metallurgy continued the trend toward changing reality from a visible one to an invisible one until today 99.99 percent of what constitutes our reality is not directly contactable by our senses.

Man Must Think To Live. Mankind is also eminently suited to its role as caretaker of this end of the universe because of its lack of specialization.

Fuller points to the unique advantages enjoyed by all other species—birds with wings, animals with speed, strength, cunning, or daring, fish with protective coloring—and concludes that while these special advantages help to preserve each species, they also tend to channel species development in directions where their advantages can be best exploited.

On the other hand, man has no unique advantage. In fact,

Innovative Design Software, Inc.
ANNOUNCES
SHUFFLEBOARD
for your APPLE II.™
only $29.95

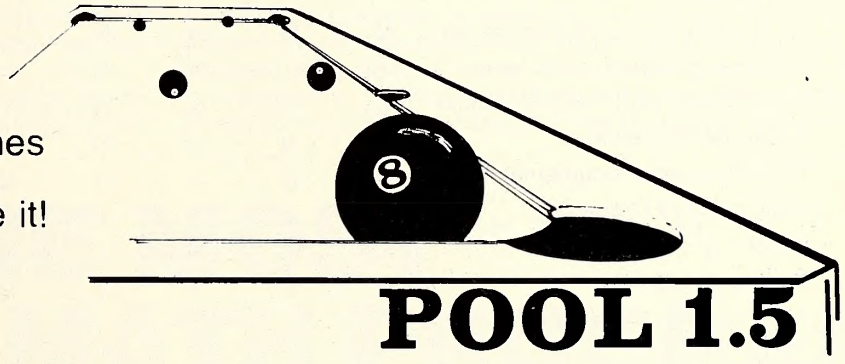


- *Real time HIRES Color Graphics*
- *Play Against your APPLE or another opponent*
- *An accurate and challenging game following in the tradition of Pool 1.5*

— Another first from IDSI

POOL 1.5 features

- Realistic, life-like motion
- HIRES Color Graphics
- Choice of 4 popular pool Games
- You've Got to see it to believe it!
- Only $34.95



"IT (Pool 1.5) is so realistic, you begin filling in the details of the pool hall yourself"
— Softalk Magazine

Both of these excellent games require a 48K APPLE II/Plus and a DISK II. Pool 1.5 requires a set of game paddles or Joystick. Order directly from IDSI or see your nearest APPLE dealer.

IDSI

P.O. BOX 1658
Las Cruces N.M. 88004
(505) 522-7373

We accept
Visa, MasterCard,
Check or Money Order.

• Apple II/Plus is a Trademark of Apple Computer Inc. Pool 1.5 is a trademark of IDSI

Lazer isn't afraid to compare!

(because we have the best lower case system available.)

Despite the fact that we were one of the first manufacturers to produce lower case equipment for the Apple II, Lazer MicroSystems products are still the state-of-the-art. Beside the obvious price/performance advantage we have over the competition, our products are expandable. Lazer is constantly introducting new products including our Lower Case + Plus II, Character Set + Plus (that adds 2 additional character sets to the Lower Case + Plus), and our new "Double Vision + Plus" for owners of Computer STOP's Double Vision 80-column board.

Lazer's products are compatable with more word processors than anybody else's. Our Lower Case + Plus is compatable with Easywriter and unmodified Apple writers. None of our competitors below can make that claim. In fact, BASIS' board isn't even compatable with Programma's PIE! The following chart lists Lazer's superiority over the competing units.

(LC+ = Lower Case + Plus. LC+II = Lower Case + Plus II; KB+ = Keyboard + Plus)

Feature	Paymar LCA-1	Paymar LCA-2	VIDEX	BASIS	VISTA	LC+	LC+II	KB+/ LC+II	KB+/ LC+	KB+
True ASCII upper/lower case display	Y	Y	Y	Y	N	Y	Y	Y	Y	N
Inverse Lower Case	N	N	rev 7 only	N	—	Y	N	N	Y	—
Font Size	5 x 7	5 x 7	5 x 8	5 x 8	—	5x7, 7x8	5 x 7	5 x 7	5x7, 7x8	—
# of on-board character sets	1	1	1	1	—	up to 4 (2 std)	1	1	up to 4	—
Pseudo-descenders	Y	Y	N	N	—	Y	Y	Y	Y	—
True descenders	N	N	Y	Y	—	optional	N	N	optional	—
Optional fonts avail. (ROM, disk)	N	N	N	Y	—	Y	N	N	Y	—
2716-compatible character generator compatable with fonts created by HIRES character generators	N	N	N	N	—	Y	N	N	Y	—
On-board graphics character set	N	N	N	N	—	Y	N	N	Y	—
Software provided on diskette	$5 extra		N	N	—	Y	Y	Y	Y	Y
Single board works with all Apples	N	N	N	N	Y	Y	N	N	Y	Y
Expandable System	N	N	N	N	N	Y	Y	Y	Y	Y
Extensive user Documentation	N	N	Y	N	N	Y	Y	Y	Y	Y
High quality PC board	N	—	Y	Y	Y	Y	—	Y	Y	Y
Reset key disable	N	N	Y	Y	N	N	N	Y	Y	Y
Shift key mod	N	N	Y	Y	N	N	N	Y	Y	Y
All 128 characters available from keyboard	—	—	N	N	—	—	—	Y	Y	Y
Type ahead buffer	N	N	N	N	Y	N	N	Y	Y	Y
# of characters in buffer	—	—	—	—	40	—	—	64	64	64
Ability to clear or turn off buffer	—	—	—	—	N	—	—	Y	Y	Y
PRICE	59.95	49.95	129.95	125.00	49.95	64.95	29.95	129.90	164.90	99.95

Don't settle for anything less than the Lower Case + Plus, Lower Case + Plus II, or Keyboard + Plus. If your dealer doesn't stock our products give us a call, we'll try to connect you with a dealer in your area.

Before you buy a lower case system for the Apple II, call or write for our free booklet "Keyboard Enhances/Buffers and Lower Case Adapters: From The Inside Out." This booklet explains the advantages and disadvantages of using a lower case adapter in your Apple II.

Lazer products are carried by reputable dealers all across the world. If your dealer cannot provide you with a demonstration of our equipment and tries to sell you an inferior lower case adapter, give us a call, we'll give you the location of a dealer that can show you our equipment. Remember, an intelligent purchase cannot be made if you do not compare before buying.

Lazer's Products are recommended for use with DB Master by Stoneware.

State of the art performance, software compatability, and exceptional value make Lazer's products the best there are!

Lower Case + Plus	$64.95
Keyboard + Plus	$99.95
Lower Case + Plus II	$29.95
Character Set + Plus	$24.95
Double Vision + Plus	$39.95
Anix 1.0	$49.95
Lazer Pascal	$39.95

Visicalc Users! Now you can have lower case on your Visicalc Screen, *only* from Lazer.

Lazer MICRO SYSTEMS INC.
1791-G Capital
Corona, CA 91720
(714) 735-1041

Visicalc - TM of Personal Software
DB Master - TM of Stoneware
Apple II - TM of Apple Computer, Inc.

he comes to the world naked, helpless, and ignorant. This seemingly hapless state may be mankind's greatest strength in that it forces learning by trial and error.

The phrase "naked, helpless, and ignorant" rolls off Fuller's tongue often in any lengthy discourse. The understanding of man's origins is central to the theses he develops. Man's state at birth and his lack of special advantages forces him to maximize those attributes, such as mind, that he has. Development of the mind is what makes man suitable as a husbandryman for this part of the universe. Man is helpless by design—mind is everything and muscle is nothing.

Thwarting Responsibility. From Fuller's vantage point, however, mankind has failed to seize the responsibility thrust upon it. Instead, it has politicized itself, with the result that vested interests have prevented adoption of policies that would enhance man's chances of success in the universe.

The political activities of mankind have deleterious effects in two directions: they interfere with the self-regenerative process of the universe and they inhibit proper understanding of the universe that would cause knowledgeable people to seek political change.

Fuller believes that the major political blocs sprung up as defenses against the law of scarcity. As organized societies, they had a better chance of competing for those resources in short supply.

One effect of their actions has been to hoard rather than husband the resources available. This is a natural outgrowth of the view that such resources are limited. But hoarding inhibits the regenerative process of the universe by withholding from that process raw materials.

That's a serious enough effect to raise questions about the viability of multiple political bodies on Spaceship Earth. Equally as dangerous, however, is the fact that the political entities recognized that they had a vested interest in maintaining the myth of scarcity, even in the face of new technology that promised abundance.

The response of the governments has been to tie the people to them in divers ways as well as educating them all to become specialists.

The Main Problem: Dependence and Specialization. Fuller views water mains and sewer lines as devices designed specifically to make the individual dependent on the state. During his last visit to China, he pledged that he would not return until he had perfected the means to emancipate the Chinese from the chains of government-owned sanitation devices.

He now predicts that he's a year away from delivering a dry package toilet that will convert waste either to fertilizer or to gas and a means of personal cleansing using pressurized air instead of water.

But more dangerous in the long run than man's dependence on his government for services is man's ignorance caused by the means by which he's educated. Fuller believes that public education is designed to make each person a specialist. In the context of Fuller's view that mankind is the species best suited to husband this corner of the universe, making man a specialist is especially abhorrent.

Because the power structure fears high intelligence, it turns that intelligence in an individual to specialized functions where only a part of the picture can be seen instead of the entire picture.

Buckminster Fuller sees that as truly limiting. Nothing about the behavior of any part of a structure will, in his view, permit you to understand the functioning of the whole. There is nothing about one atom that predicts another atom.

Missing the Forest—and the Universe. There is nothing about an atom that predicts amino acids. Likewise, there is nothing about amino acids that predicts protoplasm. And nothing in protoplasm will permit one to deduce camels or palm trees.

Knowing atoms will not tell you that camels are smelly, cranky animals that bite. Yet those are germane facts.

When man becomes a specialist, he loses sight of the larger picture. In the case of mankind, specializing prevents a clear understanding of the universe, which is the most critical whole to comprehend. Fuller uses himself as an example, saying that he had to throw off the shackles of specialization to enable him-

self to grasp the verities of the universe.

So Fuller wages verbal war on the power structures embodied in political bodies as conducting themselves contrary to the best interest of the universe.

But what are the great technological breakthroughs that Fuller believes have repealed the fundamental law of scarcity? They are the get-more-for-less technologies that started arriving at the end of the sixties.

The Apple and the Dome. Technology has made it feasible to provide more basic needs with fewer pounds of materials, using fewer ergs of energy, and using fewer units of time. The personal computer is one example of the more-for-less technology. A loaded Apple II Plus approaches the computing power of the IBM 360 mainframes that were prevalent in the late sixties. Yet the Apple patently employs fewer materials in construction, requires smaller inputs of power to operate, and is constructed and maintained with far fewer units of manpower.

Construction technology has taken the same route. Fuller's own patented geodesic domes provide shelter that is stronger and more reliable than steel and brick at fractions of the cost and weight. The domes are energy efficient and can be constructed in relatively short time periods as compared to conventional structures.

Fuller's dome structures exist throughout the world, but they've been especially adopted by people in harsh environments such as the South Pole. What was required there were structures that could withstand 180 mile per hour winds and snow loads of as much as three hundred pounds per square foot. Structures using conventional building techniques cannot handle snow loads of more than sixty pounds per square foot, but Fuller's domes have been providing shelter for scientists at the South Pole for years.

The magnitude of the energy savings using more functional shapes such as domes and newer materials of higher tensile strength than steel and brick is brought home by considering the island of Manhattan, home of the skyscraper. To enclose Manhattan in a dome that would provide the same living and working space would cut the energy use eighty-four fold. Magnify that savings worldwide and you've made practical one of the planks of Fuller's platform—phase out the use of fossil fuels and atomic energy.

Fuller points out that fossil fuels are far more expensive than any price an individual can afford to pay if you consider the time and energy expended by nature to create the substance, rather than the monetary expense at the gas pump.

Fueling Nonsense and Aggression. What galls Fuller most about the use of fossil fuels is that they're used to operate reciprocating engines that are mounted in automobiles that carry people to jobs that are nonessential in terms of the universe. To him this is not just waste, but waste doubly damned.

Mankind's efforts should be expended in life support activities or in promoting the self-regenerative aspects of the universe. Other activities, such as an arms race that he views as futile, should be halted.

The trillions of dollars of materials, energy, and time that have gone into the buildup of weaponry are ghastly wastes in Fuller's view. The arms race reflects the mentality of scarcity and the need of competing power structures to defend their interests in the face of hostile competition from other power structures. Of course, if there no longer exists a situation of scarcity, then there no longer exists a need for power structures to compete for resources and therefore there no longer exists a need for defending those power structures with expensive armaments.

To allege that the arms race has its foundations in power structures competing for resources does not address another line of thought equally as likely. That body of reasoning holds that man, as other animals, is naturally aggressive and therefore societies need defense systems regardless of the relative abundance or lack thereof.

Fuller finds this hypothesis at odds with observation. He believes that aggression is a learned behavior caused by withholding life support for some period short of critical. The key life support elements are food, water, and air.

The universe has socialized air, making it freely available to all. In most areas, water supplies are adequate if not abundant. That leaves food as the life support element most likely to be withheld. This is especially true in the animal world, where periods of food deprivation are frequent. Fuller observes that the aggression quotient increases with the increased incidence of food deprivation.

Aggression is a learned behavior that stems from the aggressor having been punished by his environment through deprivation. In a future civilization of plenty, there need be no environmental deprivation for mankind.

Fuller believes the universe carries an inventory of relative abundance that can be harnessed to raise the standard of living of all persons to new heights. However, it's essential that mankind cease its activities that inhibit the self-regeneration of the universe.

Truth and the Computer. The individual's responsibility in this area is, in Fuller's words, to always tell the truth, the whole truth, and nothing but the truth. That may seem like little enough to ask, but it's vital in the overall scheme because, Fuller believes, the computer is the essential tool that can turn mankind away from its counterproductive activities to those that make more sense. But for the computer to do so, it needs truthful inputs.

Fuller places great faith in the computer as the eventual tool that will persuade the power structures to reorient their thinking away from a mindset of scarcity and toward a mindset of abundance.

He's fond of recounting the use of the computer by United Auto Worker president Walter Reuther to extract a record wage hike from General Motors. Reuther used the computer to perform complex calculations that concluded that General Motors would make higher profits by paying its workers more money.

This was in the late forties, when computers were not the ubiquitous tools they are now, and the negotiators for GM were convinced that Reuther had seeded his computer with bad information. They were shocked to find, upon running their own analyses, that his contention was absolutely correct.

What happened was that the computer was able to analyze data and detect correlations between data that escaped the visibility of man. In effect, the computer was able to prove through its logic circuits that what seemed quixotic was actually the correct and sensible course of action.

Fuller believes the computer can serve the same purpose in the present: ratifying his contention of abundance and causing the power structure to change its policies. But that makes it imperative that the computer be fed the unvarnished truth, rather than a concoction of misapprehensions and psuedolaws.

The requirement that the computer receive accurate data and be posed pertinent questions is the root of Fuller's counsel that individuals serve the cause of the universe best by truthfulness.

Fuller believes, among other things, that a computer fed good data would request that all humans be provided life support elements and that only those humans engaged in the production of life support elements should work. He believes the computer will conclude that if persons are not providing life support, they should be paid to stay home.

The energy and materials saved by eliminating occupations not directly involved with life support would go a long way toward providing the higher living standard Fuller believes is possible.

Such a policy would not necessarily lead to a life of idleness and mischief for that portion of the populace so unemployed. Fuller believes many would begin to seek out ways they could contribute to the welfare of the universe. He believes that in a world where work was not required for life support, working would become a desirable and honored calling.

Our Ultimate, Immediate Choice. Buckminster Fuller believes that mankind has the option to make it in the universe. But that option is no longer open-ended. For one thing, mankind must opt for change before one of the competing power structures decides to trigger the button that will start the next war. Fuller firmly believes that no one can win that war, which is another reason why it's so imperative to change policies now.

Another factor is that the power structures are impeding the process of self-regeneration of the universe. Even if no one pushes the button, such policies will eventually doom mankind.

What plagues Fuller is how to get the message to the four billion persons on Earth that they don't need to live in poverty. That there is abundance if only mankind will pursue appropriate policies.

To that end, Fuller crisscrosses the globe, spreading the message. He's been around the world forty-eight times and to this day, at age eighty-six, he speaks to at least one thousand persons every four days.

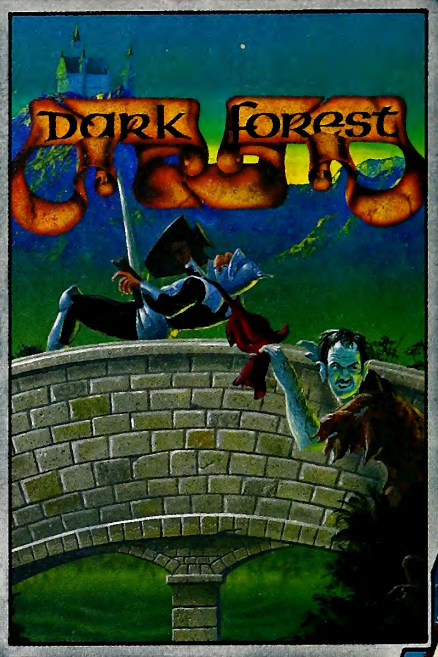
He's been an invited professor at more than six hundred colleges and has spoken on more than one thousand college campuses. For all that busy schedule, Fuller employs no public relations persons, uses no speakers bureau, has no agent, and does no promotion.

Unlike many prominent persons, he receives no remuneration from sitting on the board of directors of big companies. His only income is from speaking engagements and what monies are donated to advance his research.

But even though he's attempting to cut back on his speaking schedule, he carries a schedule more arduous than most younger persons as he strives to beat the cosmic deadline.

To Buckminster Fuller, the choice is between utopia and oblivion. He believes we have the option and the time to reach for utopia. And Fuller perseveres in forcing us to look to that option.

Bright New Stars



From Sirius Software

DARK FOREST: The age was dark, the forest was dark and the Gruds were everywhere . . . Three of your kingdom's most valued treasures are missing and you must comb the countryside to recover them. An adventurous game of strategy and conquest for up to six players.

BEER RUN: is a light-headed game of suspense. Can you catch the Artesians™ before the Guzzlers and Bouncers catch you? Enter the Sirius Building and find out!!!

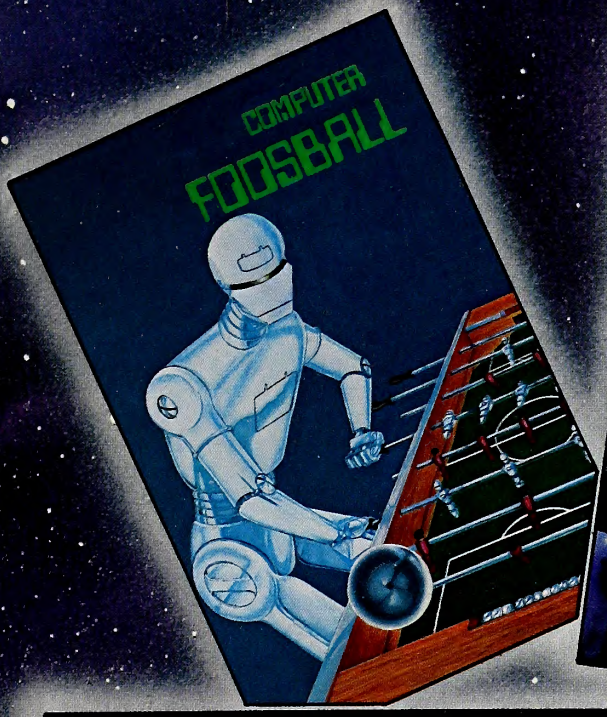
COMPUTER FOOSBALL: A fast action electronic version of this favorite table game. You and up to three friends can play this hires game using the new JOY-PORT.

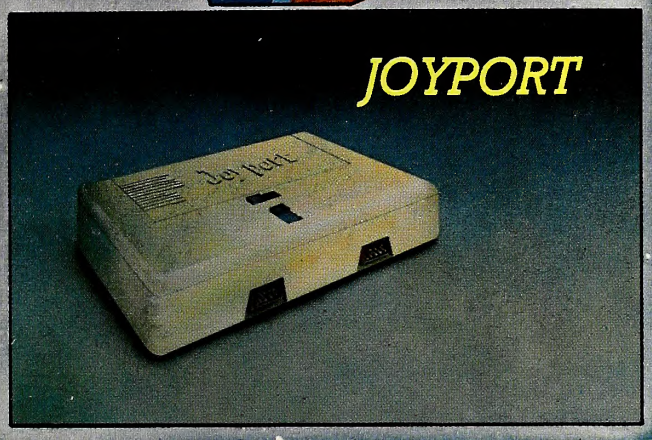
Coming Attractions . . .

AUDEX: Create sounds, shape them, edit them and play them back in your own programs. The only tools required are your Apple II keyboard, screen and an optional tape player.

BORG: Can you out run and out shoot the dragon's henchmen? Watch out for the wrath of Borg if you do!

LeGREEDY: So you always wanted to play the real estate game but couldn't afford to . . . Find out how much of LeGreedy is in you.





JOYPORT: Expand the Apple II game paddle port to handle up to four Apple type game paddles and two Atari joysticks. Four Apple game paddles can be read sequentially under software control. Comes in attractive impact resistant case.

HADRON: You are a fighter patrol in space. You are trying to follow an enemy drone ship back to its home base. To get there you must successfully negotiate a dense meteor field and duck fire from the base. Exciting 3-D play.

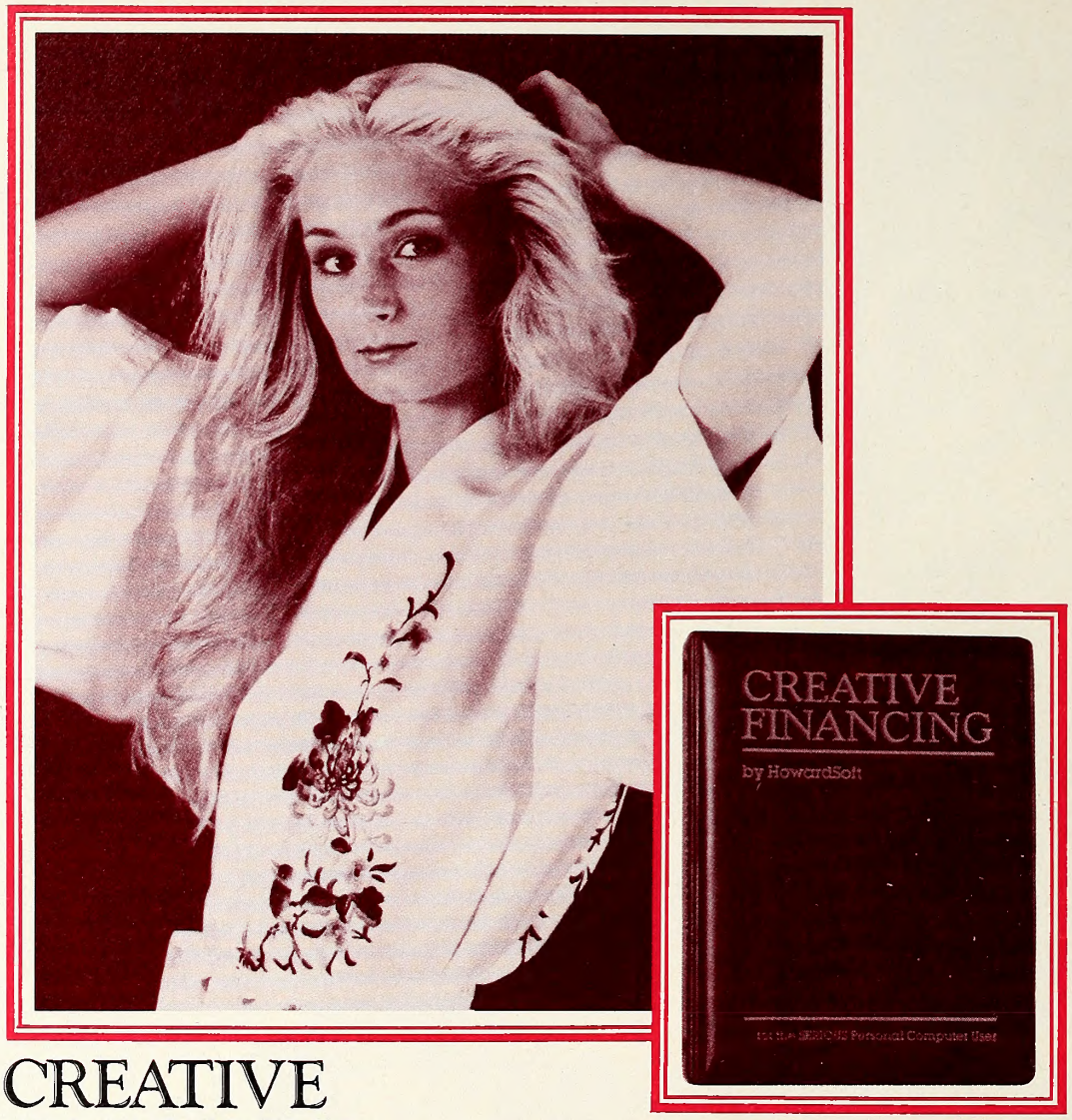
OUTPOST: Alone in a space outpost you've been attacked from all sides by enemy fighters. You must use your propulsion units and shields to ward off the attackers. A fast reflex action game.

Sirius Software, Inc.

10364 Rockingham Drive Sacramento, California 95827

All products are designed for use with Apple II computers.

WORTH WAITING FOR . . .



CREATIVE FINANCING by HowardSoft™

It's been a long time coming, but it's finally here—the powerful new financial tool from Howard Software—and it's totally UP-TO-DATE. It includes the latest in creative loan types and even reflects the tax consequences of the Economic Recovery Act of 1981, complete with the new ACRS depreciation tables for all future years. Whether you invest in bonds, stocks, real estate, or trust deeds, or you just want to evaluate annuities, mortgages, or consumer loans, you can trust this package to give you the real answers under the latest tax laws and changing economic conditions. Even more sophisticated than our respected Real Estate Analyzer, you can finally have professional analyses that give the answers to "what if" questions without requiring you to be a financial whiz.

PRACTICAL—generates professional reports complete with title page, printing only the table you request; includes present and future values, loan APR and IRR, monthly loan schedules, full tax impacts, equivalent annuities, annual depreciation schedule, and the useful performance measures ROI, IRR, and FMRR; great for objectively comparing dissimilar loans and investments.

FRIENDLY—not just an operating manual but a textbook on investment analysis; friendly recovery from user errors; menu driven and interactively self-prompting.

FLEXIBLE—numerous separate schedules with monthly or yearly inflating or staircase increases; tax rates, money rates, loans, depreciation, income and expense items all handled flexibly; numerous automatic features but manual entry as well.

AUTOMATIC—loan types include amortized, interest only, graduated payment, renegotiable rate, variable rate, reverse annuity, balloons, deferred payments, and more; depreciation types include straight line, sum of years' digits, declining balance (any percentage, with or without auto conversion to straight line), and ACRS.

NOW AT YOUR DEALER NEXT TO our **TAX PREPARER & REAL ESTATE ANALYZER** software. (Apple and CP/M versions available).

Howard Software Services
. . . for the SERIOUS Personal Computer User

8008 Girard Ave., Suite 310 | La Jolla, CA 92037 | (714) 454-5079

☐ **A Lot of Character.** A new typewriter and a computer are teaming up to help the Chinese learn to type. Typing is a skill many Chinese have never acquired because their language's thousands of symbols cannot fit on a keyboard. But a typewriter recently developed by Olympia in Wilhelmshaven, Germany, reproduces the characters by squirting ink through a computer-manipulated grid. The typewriter can produce fifteen signs per second from its computer bank. Now, if we could only translate that into words per minute, the Chinese could compete with us on an American typing test.

☐ **Lo-Tech Crime.** No matter how sophisticated technology becomes, there will always be a crook who uses the most unsophisticated techniques and gets away with it. A "shabby-looking box" collected some money recently from a southern California bank. According to police reports, customers who intended to stick their cash in the night deposit found an out-of-order sign on the slot. Instead the sign directed them to deposit their cash in an adjoining wooden box, which mysteriously disappeared by the next morning. Bank officials, who had no knowledge of either the out-of-order sign or the replacement deposit box, announced the robbery soon after customers started complaining that their night deposits hadn't been recorded on their monthly bank statements. The moral of this story: Computers aren't always to blame. Also, never trust a shabby box.

☐ **Just Doing What Comes Naturally.** In past issues of *Softalk* we've reported on many unusual applications for the Apple computer. Scientists, doctors, educators, and business executives have all found the Apple an invaluable tool. Although we haven't delved into the subject much, there are those who would use the Apple for less legitimate purposes.

One enterprising individual, Joseph Harvey, is a case in point. In August of last year, he and a couple of red-faced Apples were hauled off to jail for running a prostitution ring in Santa Ana, California. The thirty-six-year-old Harvey, an aerospace engineer had started a lucrative escort business that offered more than the usual services.

Using Apples and a five and a quarter inch floppy disk drive, Harvey kept track of all his customers, employees, and financial records. He also had a file called "pranks," which included known violent customers, nonpayers, and undercover agents and their pseudonyms. When a call came in it would be cross-refrenced against the data to determine if the caller was an undercover agent or some other undesirable.

Harvey also had a handy device built into his Apple—a "kill" button. If the need arose pressing "kill" would erase the disk in the drive in a matter of seconds. Sgt. Dan Felix of the Santa Ana police department says that the arresting officers entered Harvey's business fast enough to prevent the kill button from doing much damage. Felix also reports that Harvey was just about to expand from five and a quarter in inch floppies to an eight-inch floppy disk system.

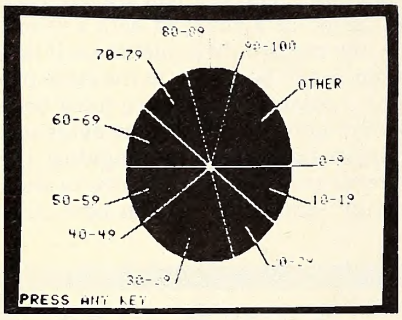
At this point, Harvey has not yet gone to trial. Since he's charged with a felony offense, the maximum punishment if he's convicted would be a stint in the state prison. The Apples and fostware have been impounded and will be used as evidence. No charges have been filed against the computers, and they have sworn to walk the straight and narrow from now on. ◨

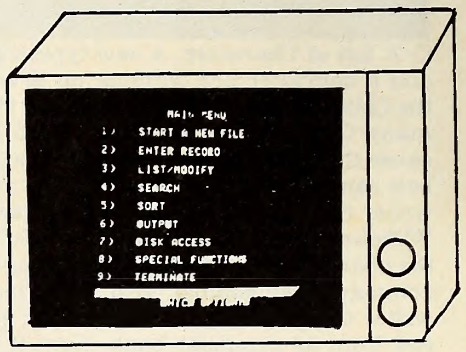
GET 120% VALUE FOR YOUR
PROGRAM PURCHASING DOLLAR
WITH

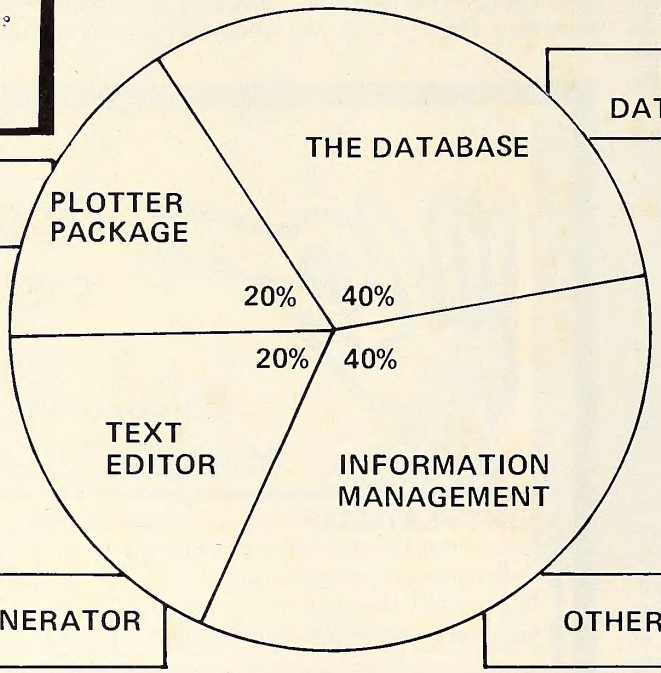
THE DATA REPORTER

MORE THAN JUST A DATABASE

Version 2 of the versatile Modifiable Database





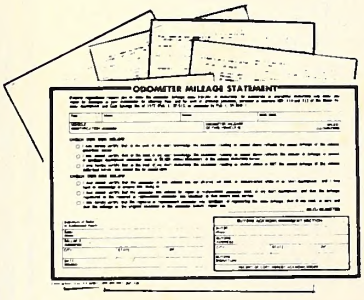


POWERFUL DATA MANAGER

- Versatile, user definable database can store data segmented by up to 35 fields.
- User oriented format is easy for the novice or professional to utilize. The use of menus, extensive prompting, single keystroke commands, and a universal escape capability allow anyone to store or retrieve information in seconds without errors.
- Machine language searches and sorts operate in a fraction of the time required by other programs.
- Searches or sorts, subtotals or totals may be performed on any field at any time, not just on those that are indexed or specified in advance.
- Search results may be displayed, printed, deleted, counted, totalled, edited, and/or saved to a new data file.

DATA PLOT & ANALYSIS

- Data may be plotted in a variety of formats such as scatter graphs, line graphs, bar charts, and pie charts.
- Ranges, minimums, maximums, means, standard deviations, correlation coefficients, etc. of any number of data files can be calculated



REPORT GENERATOR

- The sophisticated report generator allows you to format your data output in an infinite variety of ways.
- You can print form letters, columnar reports, lists, mailing labels, etc.
- Data, ratios or the results of calculations can be embedded anywhere in your letters or reports.
- The report generator gives your output the professional appearance that you require.

OTHER FEATURES

- **HARD DISK DRIVE COMPATIBILITY** with hard drive version, works with Corvus and other hard drives.
- Works with all Floppy drives with slot, drive and volume selection.
- You can append or merge up to a full disk of data files, or segment your data into separate files by a search key.
- Searches can contain up to 10 levels. You can search for a key word in any field, the absence of a keyword, or a number being within a specified range.
- Global editing of data may be performed.
- Arithmetic processing can be performed during record entry, edit, or output.
- Record entry, edit, or deletion (individual records or blocks) can be performed with no tedious delays waiting for disk accesses, index file updates, etc.
- Data may be stored on any number of floppy or hard disk drives.
- Data files can be reformatted at any time without reentering the data.
- With $5.00 Registration Fee receive one backup disk.
- The package requires an Apple II plus or Apple II with Applesoft firmware, 48K RAM, at least one disk drive, and DOS 3.3.

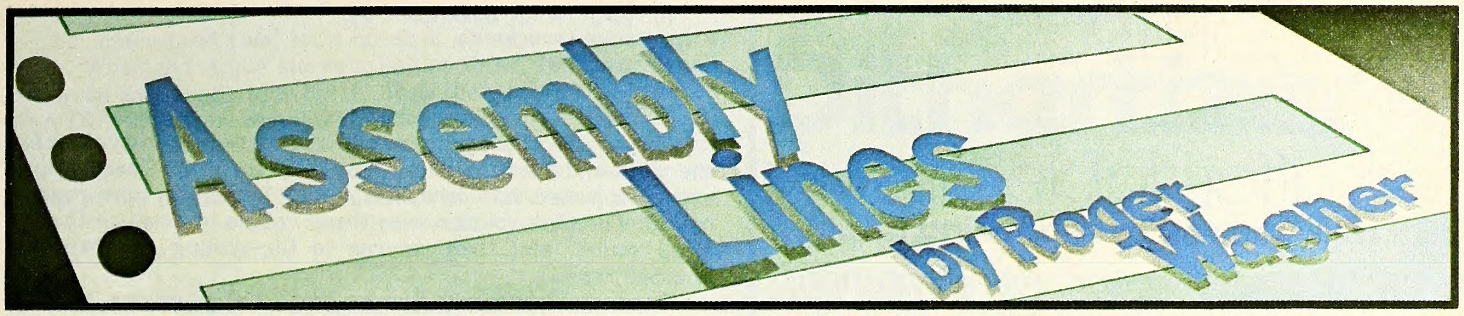
SYNERGISTIC SOFTWARE

ORDER YOURS TODAY!

Floppy Drive version **$220.00** — Hard Drive version **$220.00**

Available from your local dealer or send check or money order to **Synergistic Software**, 5221 120 Avenue S.E., Bellevue, Washington 98006 or phone **206-226-3216**.

Washington residents add 5.4% sales tax.
Apple is a trademark of Apple Computer, Inc.



Everyone's Guide to Assembly Language, Part 16

One useful application of machine language programming is in the enhancement of your existing Applesoft programs. Some people are inclined to write all their programs in machine language, but it may be more efficient on occasion to write "hybrids"—programs that are a combination of Applesoft and machine language. In this way, particular functions can be done by the operating system best suited to the particular task.

If you had to write a short program to store ten names, it would be best to do it in Applesoft:

```
10  FOR I = 1 TO 10
20  INPUT N$(I)
30  NEXT I
```

This is much simpler than the equivalent program in machine language. In cases where neither speed nor program size is a concern, Applesoft is a completely acceptable solution.

However, if you had to sort a thousand names, speed would become a concern, and it would be worth considering whether the job could best be done in machine language.

If you have ever done a *call* in one of your Basic programs, then you have already combined Applesoft with machine code. For example:

```
10  HOME
20  PRINT "THIS IS A TEST"
30  PRINT "THIS IS STILL A TEST"
40  GET A$
50  VTAB 1: HTAB 5: CALL-958
```

In this program, a line of text is printed on the screen. After you press a key, all text on the screen after the first word "THIS" is cleared.

Now although it might be possible to accomplish the same effect in Applesoft by printing many blank lines, it would not be as fast or as efficient in terms of code as the *call -958*.

In executing the above program, the Applesoft interpreter goes along carrying out your instructions until it reaches the *call* statement. At that point a JSR is done to the address indicated by the *call*. When the final RTS is encountered, control returns to the Basic program. In between, however, you can do anything you'd like!

*Call*ing routines is hardly complicated enough to warrant an entire article on the subject. The real questions are, how do you pass data back and forth between the two programs, and how can the problem of handling that data be made easier for the machine language program?

Simple Interfacing. The easiest way to pass data to a machine language routine is simply to *poke* the appropriate values into unused memory locations, and then retrieve them when you get to your machine language routine. To illustrate this, let's resurrect the tone routine from the May 1981 issue of *Softalk*.

To use this, assemble the code and place the final object code at $300. Then enter the accompanying Applesoft program.

```
 1  **********************
 2  * SOUND ROUTINE #3A *
 3  **********************
 4  *
 5  *
 6    OBJ $300
 7    ORG $300
 8  *
 9  PITCH EQU $06
10  DURATION EQU $07
11  SPKR EQU $C030
12  *
13  BEGIN LDX DURATION
14  LOOP LDY PITCH
15    LDA SPKR
16  DELAY DEY
17    BNE DELAY
18  DRTN DEX
19    BNE LOOP
20  EXIT RTS
```

From the Monitor, this will appear as:

```
*300L

0300-  A6 07     LDX   $07
0302-  A4 06     LDY   $06
0304-  AD 30 C0  LDA   $C030
0307-  88        DEY
0308-  D0 FD     BNE   $0307
030A-  CA        DEX
030B-  D0 F5     BNE   $0302
030D-  60        RTS
```

This Applesoft program is used to call it:

```
10  INPUT "PITCH, DURATION? ";P,D
20  POKE 6,P: POKE 7,D
30  CALL 768
40  PRINT
50  GOTO 10
```

The Applesoft program works by first requesting values for the pitch and duration of the tone from the user. These values are then *poke*d into locations 6 and 7 and the tone routine *call*ed. The tone routine uses these values to produce the desired sound, and then returns to the *call*ing program for another round.

This technique works fine for limited applications. Having to *poke* all the desired parameters into various corners of memory is not flexible, and strings are nearly impossible. There must be an alternative.

The Internal Structure of Applesoft. If you've been following this series for long, you've no doubt figured out by now that I'm a great believer in using routines already present in the Apple where possible, to accomplish a particular task. Since routines already exist in Applesoft for processing variables directly, why not use them?

To answer this, we must take a brief detour to outline how Applesoft actually "runs" a program.

Consider this simple program:

```
10  HOME: PRINT "HELLO"
20  END
```

After you've entered this into the computer, typing *list* should reproduce the listing given here. An interesting question arises: "How does the computer actually store, and then later execute, this program?"

To answer that, we'll have to go to the Monitor and examine the program data directly.

The first question to anwer is, exactly where in the computer is the program stored? This can be found by entering the Monitor and typing in: 67 68 AF B0 and pressing return.

The computer should respond with:

```
67-     01
68-     08
AF-     18
80-     08
```

The first pair of numbers is the pointer for the program beginning, bytes reversed of course. They indicate that the program starts at $801. The second pair is the program end pointer, and they show it ends at $818. Using this information let's examine the program data by typing in:

```
801L
```

You should get:

```
*801L
0801-  10 08     8PL   $0808
0803-  0A        ASL
0804-  00        8RK
0805-  97        ???
0806-  3A        ???
0807-  8A        TSX
0808-  22        ???
0809-  48        PHA
080A-  45 4C     EOR   #$4C
080C-  4C 4F 22  JMP   $224F
080F-  00        BRK
0810-  16 08     ASL   $08,X
0812-  14        ???
0813-  00        8RK
0814-  80        ???
0815-  00        8RK
0816-  00        8RK
0817-  00        BRK
0818-  F9 A2 00  S8C   $00A2,Y
0818-  86 FE     STX   $FE
```

This is obviously not directly executable code. Now type in:

```
801.818
```

This will give:

```
0801-  10 08 0A 00 97 3A BA
0808-  22 48 45 4C 4C 4F 22 00
0810-  16 08 14 00 80 00 00 00
0818-  8C
```

To understand this, let's break it down one section at a time. When the Apple stores a line of Basic, it encodes each keyword as a single byte *token*. Thus the world *print* is stored as a $BA. This does wonders for conserving space. In addition, there is some basic overhead associated with packaging the line, namely a byte to signify the end of the line, and a few bytes at the beginning of each line to hold information related to the length of the line, and also the line number itself.

To be more specific:

```
0801- [10 08] 0A 00 97 3A BA
0808- 22 48 45 4C 4C 4F 22 00
0810- [16 08] 14 00 80 00 [00 00]
0818- 8C
```

The first two bytes of every line of an Applesoft program are an *index* to the address of the beginning of the next line. At $801,802 we find the address $810 (bytes reversed). This is where line 20 starts. At $810 we find the address $816. This is where the next line would start if there were one. The double 00 at $816 tells Applesoft that this is the end of the Basic listing. It is important to realize that the 00 00 end of the Applesoft program usually, *but not always*, corresponds to the contents of $AF,B0. It is possible to hide machine language code between the end of the line data and the actual end as indicated by $AF,B0—but more on that later.

The next information within a line is the line number itself:

```
0801- 10 08 [0A 00] 97 3A BA
0808- 22 48 45 4C 4C 4F 22 00
0810- 16 08 [14 00] 80 00 00 00
0818- 8C
```

The 0A 00 is the two-byte form of the number ten, the line number of the first line of the Applesoft program. Likewise, the 14 00 is the data for the line number twenty. The bytes are again reversed. After these four bytes, we see the actual tokens for each line.

```
0801- 10 08 0A 00 [97 3A BA]
0808- [22 48 45 4C 4C 4F 22 00]
0810- 16 08 14 00 [80 00] 00 00
0818- 8C
```

All bytes with a value of $80 or greater are Applesoft keywords in token form. Bytes less than $80 represent normal ASCII data (letters of the alphabet, for example). Examining the data here we see a $97 followed by $3A. $97 is the token for *home*, and $3A the colon. Next, $BA is the token for *print*. This is followed by the quote ($22) and the text for HELLO (48 45 4C 4C 4F) and the closing quote ($22). Last of all, the 00 indicates the end of the line.

In line number twenty, the $80 is the token for *end*. As before, the line is terminated with a 00.

When a program is executed, the interpreter scans through the data. Each time it encounters a token, such as the *print* token, it looks up the value in a table to see what action should be taken. In the case of *print*, this would be to output the characters following the token, namely "HELLO".

This constant translation is the reason for the use of the term *interpreter* for Applesoft Basic.

Machine code on the other hand is directly executable by the 6502 microprocessor and hence is much faster, since no table lookups are required.

In Applesoft, a syntax error is generated whenever a series of tokens is encountered that is not consistent with what the interpreter expects to find.

Passing Variables. So, back to the point of all this. The key to passing variables to your own machine language routines is to work with Applesoft in terms of routines already present in the machine. One of the simplest methods was described in the October 1981 issue of *Softalk*, wherein a given variable is the very first one defined in your program (see the input routine). This is okay, but rather restrictive. A better way is to name the variable you're dealing with right in the *call* statement.

The important points here are two components of the Applesoft interpreter: TXTPTR and CHRGET (and related routines).

TXTPTR is the two-byte pointer ($B8, B9) that points to the

We'd like your apple® to meet The Executive Secretary™

The professional word processing system for the Apple™ computer

check this chart:

GENERAL Editing, printing, form letters, mail-merge, and electronic mail all in one package at one price?	✓	
User's manual designed for the user?	✓	
User-controlled configuration of printers, slots, drives, and 40/80 column editing?	✓	
THE MANUAL Complete index included?	✓	
Organized as a set of lessons?	✓	
Easel-bound for ease of use?	✓	
THE DOCUMENT PRINTER Integrates files from DB MASTER's Utility Pack™, The Data Factory™, Visicalc™, Information Master™, and most DIF™ files.	✓	
Accepts keyboard input at print time?	✓	
Supports all major printers, including Centronics 737 and IBM ET- series?	✓	
Has IF and related commands to allow conditional printing of information based on the contents of a database or on keyboard input?	✓	
Prints page headers of arbitrary complexity?	✓	
Prints page numbers wherever you want them?	✓	
Automatically generates alphabetical index for words you specify?	✓	
Supports file chaining and file nesting?	✓	
Has multi-level outline indenting?	✓	
Has left- and right-justified tab stops?	✓	
Gives full control of all margins, dynamic text reformatting, centering, and justification?	✓	
Supports Thunderclock™ and the CCS clockcard for automatic dating?	✓	

INTEGRATED CARD FILE FEATURES Allows multiple card files per disk?	✓	
Allows user to define size and content of 'cards' in each file?	✓	
Generates new subset card files based on search or sort criteria for an existing file?	✓	
Incorporates one/multiple line report printer for card files?	✓	
Allows totals and subtotals during report printing?	✓	
THE DOCUMENT EDITOR Keeps up with professional typing speeds?	✓	
User-defined phrase abbreviations?	✓	
40 or 80 character edit modes user-selectable?	✓	
Supports Smarterm™, Superterm™, Videoterm™ and Full View 80™.	✓	
Uses real shift key?	✓	
Supports file merge and unmerge?	✓	
Global search and replace?	✓	
Block operations: move, transfer, delete?	✓	
Character/word/line: insert/replace/delete?	✓	
Allows embedded commands to control special printer functions?	✓	
ELECTRONIC MAIL SYSTEM Menu driven?	✓	
Multiple document queuing?	✓	
Fully automatic with Hayes modem?	✓	

Published by: SOF/SYS, INC.

4306 Upton Avenue South
Minneapolis, Minnesota 55410
Phone (612) 929-7104

Distributed by:

Aurora Systems
2040 East Washington Ave.
Madison, WI 53704
(608) 249-5875

Eastern Software
17 Commerce Street
Baltimore, MD 21012
(301) 539-5022

Software Express
2615 Miller Avenue
Mountain View, CA 94041
(415) 949-1118

next token to be analyzed. CHRGET ($B1) is a very short routine that actually resides on the zero page and that reads a given token into the accumulator. In addition to occasionally being called directly, many other routines used CHRGET to process a string of data in an Applesoft program line.

Here then is the revised tone routine:

```
 1  **********************
 2  * SOUND ROUTINE #3B *
 3  **********************
 4  *
 5  *
 6    OBJ $300
 7    ORG $300
 8  *
 9  PITCH EQU $06
10  DURATION EQU $07
11  SPKR EQU $C030
12  *
13  COMBYTE EQU $E74C
14  *
15  ENTRY JSR COMBYTE
16    STX PITCH
17    JSR COMBYTE
18    STX DURATION
19  *
20  BEGIN LDX DURATION
21  LOOP LDY PITCH
22    LDA SPKR
23  DELAY DEY
24    BNE DELAY
25  DRTN DEX
26    BNE LOOP
27  EXIT RTS
```

This would list from the Monitor as:

```
*300L

0300-  20  4C  E7  JSR  $E74C
0303-  86  06      STX  $06
0305-  20  4C  E7  JSR  $E74C
0308-  86  07      STX  $07
030A-  A6  07      LDX  $07
030C-  A4  06      LDY  $06
030E-  AD  30  C0  LDA  $C030
0311-  88          DEY
0312-  D0  FD      BNE  $0311
0314-  CA          DEX
0315-  D0  F5      BNE  $030C
0317-  60          RTS
```

The Applesoft calling program would then be revised to read:

```
10  INPUT "PITCH DURATION? ";P,D
20  CALL 768,P,D
30  PRINT
40  GOTO 10
```

This is a much more elegant way of passing the values and also requires no miscellaneous memory locations as such (although for purposes of simplicity the tone routine itself still uses the same zero page locations.)

The secret to the new technique is the use of the routine COMBYTE ($E74C). This is an Applesoft routine which checks for a comma and then returns a value between $00 and $FF (0–255) in the X register.

It is normally used for evaluating *pokes*, *hcolor=*, and so forth, but does the job very nicely here. It also leaves TXTPTR pointing to the end of the line (or a colon if there was one) by using CHRGET to advance TXTPR appropriate to the number of characters following each comma. Note also that any legal expression—such as (X–5)/2—can be used to pass the data.

To verify the importance of managing TXTPTR, try putting a simple RTS ($60) at $300. Calling this you will get a SYNTAX ERROR, since upon return, Applesoft's TXTPTR will be on the first comma, and the phrase ",P,D" is not a legal Applesoft expression.

Now what about two-byte quantities? To do this, a number of other routines are used. For example, this routine will do the equivalent of a two-byte pointer *poke*. Suppose for instance you

wanted to store the bytes for the address $9600 at locations $1000, 1001. Normally in Applesoft you would do it like this:

```
.
.
.
50 POKE 4096,0: POKE 4097,150
.
.
.
```

Where 4096 and 4097 are the decimal equivalents of $1000 and $1001 and 0 and 150 are the low-order and high-order bytes for the address $9600 ($96 = 150, $00 = 0).

A more convenient approach might be like this:

```
.
.
.
50 CALL 768, 4096, 38400
.
.
.
```

or perhaps:

```
.
.
.
50 CALL 768, A, V
.
.
.
```

The routine for this would be:

```
 1  ***************************
 2  * POINTER SET UP ROUTINE *
 3  ***************************
 4  *
 5  *
 6     OBJ $300
 7     ORG $300
 8  *
 9  CHKCOM EQU $DEBE
10  FRMNUM EQU $DD67
11  GETADR EQU $E752
12  LINNUM EQU $50 ; ($50,51)
13  *
14  PTR EQU $3C
15  *
16  ENTRY JSR CHKCOM
17     JSR FRMNUM; EVAL FORMULA
18     JSR GETADR ; PUT FAC INTO LINNUM
19     LDA LINNUM
20     STA PTR
21     LDA LINNUM+1
22     STA PTR+1
23  *
24     JSR CHKCOM
25     JSR FRMNUM
26     JSR GETADR
27  *
28     LDY #$00
29     LDA LINNUM
30     STA (PTR),Y
31     INY
32     LDA LINNUM+1
33     STA (PTR),Y
34  *
35  DONE RTS
```

which will list from the Monitor as:

```
*300L

0300-  20  BE  DE  JSR  $DEBE
0303-  20  67  DD  JSR  $DD67
0306-  20  52  E7  JSR  $E752
0309-  A5  50      LDA  $50
030B-  85  3C      STA  $3C
030D-  A5  51      LDA  $51
030F-  85  3D      STA  $3D
0311-  20  BE  DE  JSR  $DEBE
0314-  20  67  DD  JSR  $DD67
0317-  20  52  E7  JSR  $E752
031A-  A0  00      LDY  #$00
031C-  A5  50      LDA  $50
031E-  91  3C      STA  ($3C),Y
0320-  C8          INY
0321-  A5  51      LDA  $51
0323-  91  3C      STA  ($3C),Y
0325-  60          RTS
```

The special items in this routine include CHKCOM, a syntax-checking routine that serves two purposes. First it verifies that a command follows the *call* address, and secondly it advances TXTPTR to point to the first byte of the expression immediately following the comma. If a comma is not found, a syntax error is generated.

FRMNUM is a routine that evaluates any expression and puts the real floating-point number result into Applesoft's *floating-point accumulator,* usually called FAC. This is a six-byte pseudo register ($97-9C) used to hold the floating-point representation of a number. It includes such nifties as the exponential magnitude of the number and the equivalent of the digits of the logarithm of the number stored.

At this stage you'd have to be something of a masochist to want to deal with the number in its current form, so the next step is used to convert it into a two-byte integer.

GETADR does this by putting the two-byte result into LINNUM, LINNUM+1 ($50,51).

Even if this is not exactly an in-depth explanation of all the most precise details of the operation, the bottom line is that the three JSRs (CHKCOM, FRMNUM, and GETADR) will always end up with the low-order and high-order bytes of whatever expression follows a comma in LINNUM and LINNUM+1.

These simple subroutines should be quite adequate for many applications. Next month, however, we'll look at string passing, some of the various other routines available, and how to pass data back to the *call*ing Applesoft program. ■

Are you ready to step into the world of

Wizardry™

The Ultimate Fantasy Experience

Wizardry—a revolutionary game for your APPLE II computer. Never before has a game done so much, so well, so fast! Groups of up to 6 adventurers explore a deep and mysterious maze in search of loot and glory. Brawny fighters, frail mages, nimble thieves, all must cooperate to survive. Not only must you battle hordes of monsters, but you must also solve the secret riddles hidden in the mazes. Starting from the safety of the castle, you must map the 3D maze as you move through it, swiftly running down the corridors and smashing through doors! Suddenly you encounter a group of monsters in their hideous lair! Leaping to the attack, swords swinging, your fighters wreak havoc amongst the monsters! Mages utter spells, causing destruction! Thieves skulk around in the corners, and priests attempt to bring the blessings of the gods upon your party! After the melee, there may be a chest to open, traps to evade, and loot to be divided!

A partial list of Wizardry features includes—A 10 level maze—8 character classes—5 races—20 stored on disk—3D maze display—complete castle—hundreds of monsters and magic items—monsters appear in mixed groups—50 castable spells, usable by players, magic items and even monsters—44 page illustrated manual and much, much more all for $49.95 (N.Y. residents add sales tax).

But don't take our word for it, *Wizardry* received reviews in the May issue of *Creative Computing*, the April issue of *Popular Mechanics*, page 38, and the August issue of *Softalk* magazine.

At Leading Computer Stores Everywhere

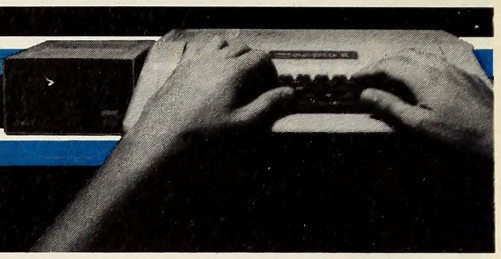
Sir-tech **SOFTWARE, INC.** Dept. F
6 Main Street / Ogdensburg, New York 13669 / (315) 393-6633

VISA

Apple is the registered trademark of APPLE COMPUTER, INC.

MasterCard

TRADETALK



☐ Round one in the legal brouhaha between **On-Line Systems** and **Atari** went to On-Line Systems. A federal judge in Fresno denied Atari's petition to prevent On-Line from shipping *Jawbreaker* on the grounds that it infringed on the audiovisual copyright held on *Pac-Man*.

The ruling only affected Atari's request for an injunction and did not go to the heart of the issue of infringement, a question presumably to be settled at a later date.

Atari's suit alleging infringement countered a prior On-Line suit against Atari alleging restraint of trade.

It was the second such defeat for Atari in a week. Earlier, in a Chicago federal court, **Magnavox** had been successful in preventing Atari from getting an injunction against *K.C. Munchkin*, Magnavox's eat-the-dots game for its Odyssey machine.

☐ **Edu-Ware Services** and its gaming branch **Interactive Fantasies** had the excitement of moving from their Canoga Park facility to new offices in Agoura, California, dampened considerably by thieves early in the dawn of December 7—once more a day of infamy. The break-in occurred when much of the company's equipment was conveniently packed and stacked for the move. Robbers simply grabbed the ready-to-ship cartons but proved to be computer illiterate when they tried for the still-set-up Apple belonging to Edu-Ware's chairman and director of research services, **Sherwin Steffin**. They couldn't manage to disconnect the computer but destroyed a controller card in the attempt.

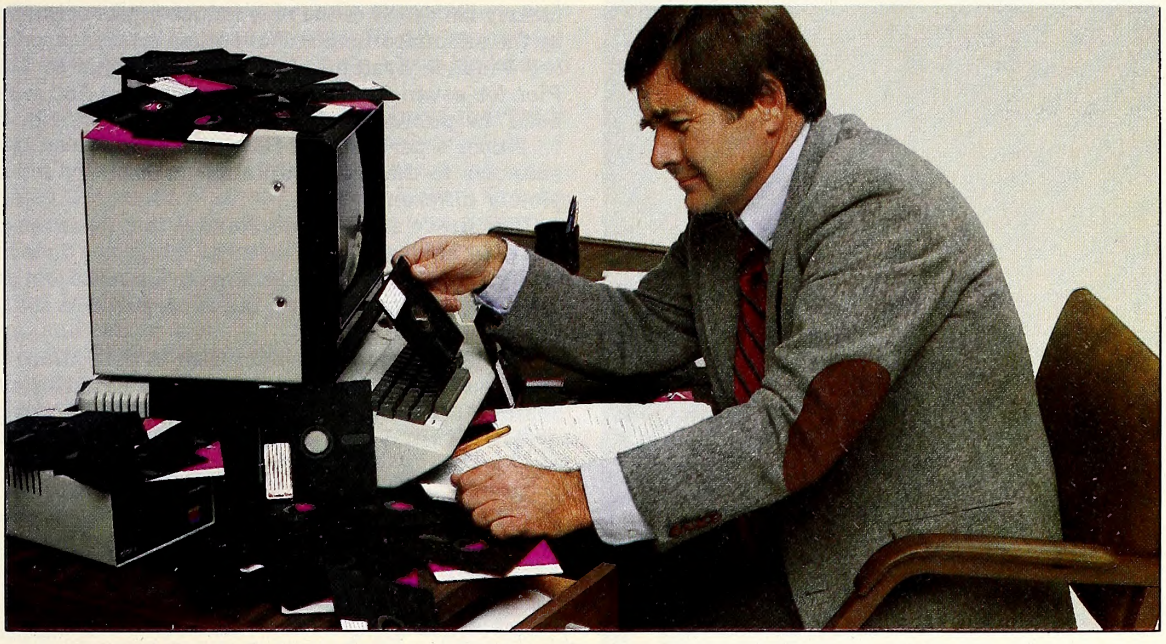
Among the equipment the scoundrels took was Edu-Ware president and premiere programmer **Dave Mullich**'s personal Apple. Other pieces included other Apples, monitors, a printer, a dictaphone, and part of the company's internal business software. Steffin estimated the loss as between fifteen and eighteen thousand dollars' worth; that the thieves confined their activities to the upstairs of the two-floor facility saved Edu-Ware from far more disastrous losses.

Investigating police warn that thefts of personal and business computers are skyrocketing. Today's thieves have entered the computer age, the officers said, and they know the market for word processors and computer systems.

If you deal with new or used equipment, watch for the following pieces stolen from Edu-Ware, here listed with serial numbers where available. If you should come across one, notify your local police or call Edu-Ware at (213) 706-0661.

1 Apple II, revision 1 motherboard
2 Apple II Pluses, one A2S2-185708, other unknown
1 black Apple (Bell & Howell Plus), A2S3-006294
6 Disk IIs, A2M3-00325, A2M3-23514,

GOTO 143



Starfire 5. The disk drive that won't play hide-and-seek.

What do we mean by that?

You chose the Apple II, and it's a good computer choice.

But Apple diskette drives play hide-and-seek with your valuable data.

And if you're not looking for a diskette, you're worrying about scratching one.

Besides, they're just too slow and hold too little.

That's why your Apple II needs a REAL business partner.

Like Starfire 5, the 5 Megabyte Winchester disk from Corona.

The one that holds 38 diskettes of data, available instantly, and can alphabetize a mailing list 5 to 10 times faster.

The one that seals your data off from danger and insures it with DataGuard automatic data protection.

Double-volume data files.

And best of all, Starfire 5 lets your files be twice as long, beating a crucial limitation of Apple's disk operating system, without abandoning it.

That's the kind of partner your Apple II needs to let it fulfill its computer promise.

At $2995, it's quite a deal.

So ask your dealer to show you Starfire 5 from Corona.

Or call us for the local dealer who can.

Because Starfire 5 is the BUSINESS partner for your Apple II.

corona data systems

213-998-0505
21541 Nordhoff St., Unit B, Chatsworth, CA 91311

Exec Lotus

A New Caper for Mitch Kapor

BY ALLAN TOMMERVIK

If you want to see a brave man cringe and a grown man near tears, just call Mitch Kapor an instant millionaire to his face.

He's actually not quite that sensitive to the charge, but he clearly isn't fond of the new industry perception of him caused by the announcement in November that his prior company, Micro Finance Systems, had sold the rights to *VisiTrend/VisiPlot* for seven-figure money to Personal Software, which had been the publisher of the program authored by Kapor.

Kapor's perception of the event, bolstered by all the facts unknown to the rest of an envious software industry, is completely different.

The first thing to understand is that no seven-figure money will ever change hands. As these things are done in the world of that size finance, the payments are spread out some to avoid the more obvious income tax repercussions that a lump sum payment would entail.

Another factor to reckon with is that Micro Finance Systems was a partnership of Kapor and Eric Rosenfeld. Clearly, some of the proceeds went to the lesser known partner.

The Skyrocket Was a Local. Finally, *instant* hardly de-

scribes the path that's taken Kapor from being a high school math prodigy to master of his own fate at Lotus Development Company.

What is Lotus Development Company? It's Kapor's new firm. Micro Finance Systems only had the one product, so it was dissolved upon completion of the sale. Kapor formed Lotus to keep his momentum going. But that's a tale for later in this yarn.

That Kapor's product was the first in the Apple market to draw such big figures has about it an aura of the incredible. After all, he didn't have the big system programming experience of Stanley Crane of *DB Master* fame or Robert Frankston of *VisiCalc* notoriety or Ken Williams of On-Line Systems.

Instead, Kapor's path led from being a disk jockey in Hartford, Connecticut, through teaching transcendental meditation and mental health counseling before he became a computer hobbyist and tinkerer.

It was a brash self-confidence that led him to becoming an Apple programmer. He had bought his Apple the previous day and was back in the store to look over software and get a question or two answered. At this time, he was unemployed, having just returned from the vacation that marked the end of his stint in mental health.

At the store counter was a man wearing a suit and buying an Apple. He was asking the usual novice questions, and Kapor, figuring he was an expert since he'd had his Apple all of one day, deigned to share his knowledge.

It turned out the buyer was in need of some custom programming. Kapor figured he was at least one day ahead, and if he could stay ahead he'd be all right. So he sold himself as an Apple programmer for $5 per hour.

Harvard's Loss. But it isn't as if Kapor embarked upon such an intellectual endeavor unequipped.

He had been a true math prodigy in high school on Long Island. As a senior, he discovered a new method of calculating square roots using recursive equations. Actually, his was one of several simultaneous discoveries of the technique.

His method was published, and he hastened to get copies of the paper to all the universities to which he had applied for admission as additional evidence of his worthiness. All the universities, that is, except Harvard, which is where his heart was. Kapor had been led to believe that entrance into Harvard was assured, so he failed to fire off this proof of his intellectual worthiness for matriculation.

The result was that of all the universities to which he applied, Harvard was the single entity that refused him admission. Kapor was stuck with Yale, although his heart has remained at Harvard, as evidenced by the location of his company in Cambridge, Massachusetts.

Kapor entered the mathematics program at Yale, but changed majors after a few weeks. It was 1967, a period when college students were more interested in protesting the war in Vietnam than in studying. Kapor joined the campus radio station and drifted away from mathematics into current events.

His eventual major was cybernetics.

From Yale, he became a disk jockey in Hartford and got interested in TM. Eventually he spent a year in Europe, studying the discipline.

After a brief stint in marketing research in Boston, he got into TM full-time until he became disillusioned, feeling that it not only wasn't solving the world's problems, it wasn't solving his problems either. In retrospect, he thinks he expected too much of it.

Subsequently, he went back to school and obtained the master of arts degree in counseling psychology. He became a mental health aide at a psychiatric hospital and soon learned "the best contribution I could make to the mental health field was to leave it."

It was then that he bought his Apple and talked his way into the job of contract programmer.

Getting Together. That first programming job was only fifteen hours a week, which left him plenty of time, but not much money. He realized that the best way to find other Apple owners who might require custom programming services was to join a user group.

Alas, there were none.

But in a copy of *Micro*, he saw an article written by Richard Suitor. He called Suitor, who did know of other Apple owners. They met and agreed to form a user group. The first meeting, in Kapor's living room, was attended by seven persons.

Eventually, Kapor found enough other clients who wanted custom programming to make it a real occupation. But in the meantime, he became a professional collector of early Apple software.

In those days, before the dawn of the professional programmer, programs were traded by the authors for other programs through user group channels. Kapor collected nearly sixty disks of public domain software.

During that period he met his future partner, Rosenfeld, who was then a graduate student at Massachusetts Institute of Technology. Rosenfeld was deeply involved in a graduate research project that needed hours of computer time to perform multiple regression operations, but he had run out of money to buy the time on the Institute's computer.

Tiny Troll—New Adventure? Kapor looked at Rosenfeld's requirements and agreed that the Apple could be made to perform the functions that Rosenfeld needed for his research. Programming in Basic and creating from scratch a tiny interpreter as he went, Kapor proceeded to create *Tiny Troll*, the first statistical analysis package sold for the Apple.

Tiny Troll went on to sell slightly more than one thousand copies even though it was never advertised or promoted. *Tiny Troll* was the basic program from which *VisiTrend/VisiPlot* eventuated, although the geographical connection is not as direct as one might assume, considering that Personal Software started in Cambridge during the same period Kapor was there.

Actually, Kapor did not come in contact with the Personal team until after Personal had moved to California. Kapor had met Robert Frankston, coauthor of *VisiCalc*. Frankston intro-

duced Kapor to Peter Jennings, one of Personal's founders.

Personal asked Kapor to add "a couple" of features to *Tiny Troll* and bring it out as a VisiProduct. The prototype finished, they offered Kapor a full-time position as new products manager and offered him stock options as an additional inducement to get him to move to California.

Kapor took the bait and was immediately thrown into the introduction of *CAA Data Management System*. That took so much of his time that he was unable to concentrate on sprucing up *Tiny Troll*. One reason for this was that "I was so inspired by *VisiCalc* that I kept adding elements to my own program to try to match the quality."

The Wrong Personality. He had joined Personal in March 1980, after the successful introduction of *VisiCalc,* so his stock options looked like money in the bank. But Kapor found he didn't like the California lifestyle and missed the Harvard Square environs. He also found out that he was too much the free spirit to work in the controlled environment of Personal, even when the company was relatively small.

So Kapor gave back his stock options in return for his release from his employment obligation and returned to Cambridge. There he concentrated on *VisiTrend/VisiPlot.* Because he wasn't working, he had to borrow money to keep going, eventually digging a $30,000 hole.

In addition, he was working under conditions that weren't exactly ideal. He didn't even have a printer to get hard copy of the 60K of code he packed into the programs. The program kept growing as he conceived further enhancements to *Tiny Troll.*

In sum, the couple of additions he contracted for with Personal in spring of 1980 took over a year to hit the market in greatly expanded form.

So when someone calls Kapor an instant millionaire, he has every reason to shudder.

The Opening of the Lotus. Regardless of whether he's earned the title of millionaire, instant or not, he's not resting on his laurels. Lotus now has four full-time programmers among its nine employees, situated in a twenty-five hundred square foot facility on Central Square in Cambridge, not far from the headquarters of Software Arts.

Kapor is readying for release his first product under the Lotus label, *Executive Briefing System,* which he categorizes as the ultimate in show and tell. *EBS* is a graphics package that will make it easy for businessmen to prepare colorful slide briefings for conferences directly on their Apple.

Although Kapor proved with *Tiny Troll* and *VisiTrend/VisiPlot* that he can program, he doesn't consider himself a computer person in the sense that he can converse knowledgeably about the speed in megahertz of a given chip. At Lotus, he confines himself to designing the projects on which the company works.

Kapor feels that there are surprisingly few excellent software packages available at present. He views software development as being in the Stone Age in terms of ordinary people using the products.

Perhaps one reasonf for this, he believes, is that it takes so long—perhaps more than a year—to develop a comprehensive piece of excellent business software. That kind of investment is so great that most software publishers opt for goals that include more frequent product introductions of software that is narrower in scope.

High Quality Takes Time. Kapor is great believer in the theory that you can't throw manpower at a software bottleneck. Instead, he prefers to approach the problems of software development more systematically.

In his view, good software development methodology is vi-

Apple II

Apple II

Quality Disk Software from SPECTRUM

PERSONAL FINANCE MASTER:

The premier personal and small business financial system. Covering all types of accounts including check registers, savings, money market, loan, credit card and other asset or liability accounts, the system has these features:

- Monthly Transaction Reports
- Budgets Income & Expense
- Reconciles to Bank Statements
- Prints Checks & Mailing Labels
- Automatic Year-End Rollover
- Prepares a Net Worth Report
- Searches for Transactions
- Handles Split Transactions
- User-Friendly Data Entry Forms
- Fast Machine Language Routines
- Extensive Error Trapping
- HI-RES Expense/Income Plots

For Apple II (48K) $74.95

COLOR CALENDAR: $29.95

Got a busy calendar? Organize it with Color Calendar. Whether it's birthdays, appointments, business meetings or a regular office schedule, this program is the perfect way to schedule your activities. The calendar display is a beautiful **HI-RES** color graphics calendar of the selected month with each scheduled day highlighted in color. Using the daily schedule, you can review any day of the month and schedule an event or activity in any one of 20 time slots from 8:00 A.M. to 5:30 P.M.

BUSINESS SOFTWARE:

Both Programs $249.95

A user-friendly yet comprehensive double-entry accounting system employing screen-oriented data input forms, extensive error-trapping, data validation and machine language routines for high speed operation. The series includes these two modules:

GENERAL LEDGER: A complete accounting system with these features:

- Up to 500 accounts and 500 transactions per month.
- Extensive check register management system (similar to our PFM).
- Prints checks and mailing labels.
- Produces these reports:
  - Transactions Journal
  - Account Ledgers
  - Income Statement
  - Balance Sheet
  - Account Listings

Requires Apple II, 48K RAM, 1 disk drive. $149.95

ACCOUNTS RECEIVABLE: A flexible system with these features:

- Up to 500 accounts and up to 500 invoices per diskette.
- Prints invoices and customer statements & address labels.
- Interfaces to General Ledger.
- Interactive screen-based invoice work sheet.
- Produces these reports:
  - Aged Receivables
  - Sales Analysis
  - Customer Listings
  - Invoice Search

Requires Apple II, 48K RAM, 2 disk drives $149.95

Both Programs $249.95

UNIVERSAL BUSINESS MACHINE:

An electronic spread sheet structured around a 100 row × 20 column table. User defines row and column names and equations forming a unique template. Table elements can be multiplied, added, subtracted, divided, summed, averaged and accumulated. Hundreds of unique templates can be created, used, stored and recalled for later use. Supplied with 8 standard templates ready for use covering these subjects.

- Cash Flow Analysis
- Proforma Profit & Loss
- Proforma Balance Sheet
- Real Estate Investment
- Sales Forecaster
- Source and Use of Funds
- Job Cost Estimator
- Inventory Analysis

Price (Apple II, 48K) $89.95

ELECTRONICS SERIES VOL I & II:

Entire Series $259.95

LOGIC SIMULATOR: SAVE TIME AND MONEY. Simulate your digital logic circuits before you build them. **CMOS, TTL**, or whatever, if it's digital logic, this program can handle it. The program is an interactive, menu driven, full-fledged logic simulator capable of simulating the bit-time response of a logic network to user-specified input patterns. It will handle up to 1000 gates, including **NANDS, NORS, INVERTERS, FLIP-FLOPS, SHIFT REGISTERS, COUNTERS** up to 20 user-defined random, or binary input patterns. Accepts network descriptions from keyboard or from **LOGIC DESIGNER** for simulation (48K) $159.95

LOGIC DESIGNER: interactive **HI-RES** graphics program for designing digital logic systems. Draw directly on the screen up to 10 different gate types, including **NAND, NOR, INVERTER, EX-OR, T-FLOP, JK-FLOP, D-FLOP, RS-FLOP, 4 BIT COUNTER** and **N-BIT SHIFT REGISTER.** User interconnects gates using line graphics commands. Network descriptions for **LOGIC SIMULATOR** generated simultaneously with the **CRT** diagram being drawn $159.95

MANUAL AND DEMO DISK: Instruction Manual and demo disk illustrating capabilities of both programs) $29.95

ELECTRONIC SERIES VOL III & IV:

Entire Series $259.95

CIRCUIT SIMULATOR: Tired of trial & error circuit design? Simulate & debug your designs before you build them! With **CIRCUIT SIMULATOR** you build a model of your circuit using **RESISTORS, CAPACITORS, INDUCTORS, TRANSISTORS, DIODES, VOLTAGE** and **CURRENT SOURCES** and simulate the waveform response to inputs such as **PULSES, SINUSOIDS, SAWTOOTHS**, etc...all fully programmable. The output is displayed as an **OSCILLOSCOPE-STYLE PLOT** of the selected waveforms or as a printed table of voltage vs time. Handles up to 50 nodes and 100 components. Requires 48 **RAM** $159.95

CIRCUIT DESIGNER: Interactive **HI-RES** graphics program for designing electronic circuits. Draw directly on the screen up to 10 different component types, including those referenced above. Component interconnect list for **CIRCUIT SIMULATOR** generated automatically, Requires 48K **RAM** $159.95

MATHEMATICS SERIES:

Entire Series $49.95

STATISTICAL ANALYSIS I: This menu driven program performs **LINEAR REGRESSION** analysis, determines the mean, standard deviation and plots the frequency distribution of user-supplied data sets. Printer, Disk, I/O routines. $19.95

NUMERICAL ANALYSIS: HI-RES 2-Dimensional plot of any function. Automatic scaling. At your option, the program will plot the function, plot the **INTEGRAL**, plot the **DERIVATIVE**, determine the **ROOTS, MAXIMA, MINIMA, INTEGRAL VALUE** $19.95

MATRIX: A general purpose, menu driven program for determining the **INVERSE** and **DETERMINANT** of any matrix, as well as the **SOLUTION** to any set of **SIMULTANEOUS LINEAR EQUATIONS** $19.95

3-D SURFACE PLOTTER: Explore the **ELEGANCE** and **BEAUTY** of **MATHEMATICS** by creating **HI-RES PLOTS** of 3-dimensional surfaces from any 3-variable equation. Disk save and recall routines for plots. Menu driven to vary surface parameters. Hidden line or transparent plotting $19.95

ACTION ADVENTURE GAMES:

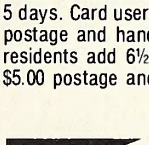
Entire Series $29.95

RED BARON: Can you outfly the **RED BARON?** This fast action game simulates a machine-gun **DOGFIGHT** between your **WORLD WAR I BI-PLANE** and the baron's. You can **LOOP, DIVE, BANK** or **CLIMB**-and so can the **BARON.** In **HI-RES** graphics plus sound $14.95

BATTLE OF MIDWAY: You are in command of the **U.S.S. HORNETS' DIVE-BOMBER** squadron. Your targets are the Aircraft carriers, Akagi, Soryu and Kaga. You must fly your way through **ZEROS** and **AA FIRE** to make your **DIVE-BOMB** run. In **HI-RES** graphics plus sound..... $14.95

SUB ATTACK: It's April 1943. The enemy convoy is headed for the **CONTROL SEA.** Your sub, the **MORAY,** has just sighted the **CARRIERS** and **BATTLESHIPS'.** Easy pickings. But watch out for the **DESTROYERS** - they're fast and deadly. In **HI-RES** graphics plus sound. $14.95

FREE CATALOG-All programs are supplied on disk and run on Apple II w/Disk & Applesoft **ROM** Card and require 32K **RAM** unless otherwise noted. Detailed instructions included. Orders shipped within 5 days. Card users include card number. Add $2.00 postage and handling with each order. California residents add 6½% sales tax. Foreign orders add $5.00 postage and handling per product.



SPECTRUM SOFTWARE
142 Carlow, P.O. Box 2084
Sunnyvale, CA 94087

master charge

VISA

FOR PHONE ORDERS:
(408) 738-4387
DEALER INQUIRIES INVITED.

tal to a successful programming endeavor. Kapor is fond of the filmmaking metaphor. As in making a movie, in software he believes you need structured groups with defined purposes. However, there's room for improvisation in the development of software, just as there is room for improvisation during filming.

One of Kapor's fears is that the personal computer user will become so inured to inadequate software that no demand for quality will ensue. Considering the potential of the personal computer, that could be the ultimate tragedy.

To combat the trend toward mediocre software, Lotus implemented stringent development techniques. Kapor designs the software and works in conjunction with his four in-house programmers to maintain high standards of implementation. With a programming crew that's comprised of big system programmers and hobbyists-turned-pro, Kapor has a versatile mix of experience from which to draw during development phases.

Visicalc was a major breakthrough in software and the microcomputer industry is awaiting the next such milestone. Until a new standard of excellence is set by a new package, Kapor believes the *VisiCalc* standard is the one to shoot for. A higher level of performance of better software tools for the end user, rather than what he terms "bit tiddling," should be the goal of all software producers.

Invisible Language Ahead. Lotus's long-range aspirations may include the development of an applications generator. Kapor hopes to take programming away from the programmers by designing a programming language that doesn't look like a programming language.

A watchword to Kapor is that it's up to the individuals to invent the future. Many applications now are possible but will not be practical until hardware and systems become more capable.

Two areas that he believes augur promise for the future are videotext and artificial intelligence. Videotext—what might become a utility for computer use in the same sense that the telephone is a utility for communications—should contribute consumer services and information services. Data bases for the home user and new applications technology may become available through videotext channels.

Kapor is much taken with the C/B option on CompuServe. The service simulates on 36 channels the C/B environment of the open road. The immediacy of the medium has suggested "all sorts" of possibilities to Kapor.

The development of artificial intelligence, Kapor concedes, is far more blue sky than information utilities, but he feels the technology has much to offer.

Double Vision. In December, Lotus moved to strengthen its position in the market by establishing Professional Software Technology Inc., a joint venture of Lotus and Pansophics Ltd. PST will handle marketing and distribution of Lotus products and the Pansophics line of spread sheet application models for *VisiCalc* and *SuperCalc.*

Kapor and Bob Ramsdell, president of Pansophics, also contemplate PST as serving as publisher for a small, select group of third-party software developers.

Even with this evidence of Kapor's commitment to the microcomputer software industry, he continues to protest that he's no computer cognoscente and doesn't intend to get himself totally immersed in microcomputers to the exclusion of other interests.

But as the man said, "Methinks he doth protest too much."

PUT ONE OVER ON APPLE II.

Introducing Station II.
The Apple II Support System.

What happens when you put one over on Apple II? You make it better. Because Station II organizes, simplifies, protects, secures and lets you control access to your Apple. In other words, it makes your Apple II your personal computer.

IT'S DESIGNED FOR II-GETHERNESS.

Station II is designed specifically for Apple II by design consultants to Apple Computer. It pulls your Apple and peripherals together into an attractive, easy-to-use, integrated system.

Your Apple is free to slide in and out, so there's no unstacking and restacking peripherals everytime you need to get inside. And you can choose the distance from keyboard to monitor that's most comfortable for you. Station II even positions your monitor at just the right angle for maximum viewing ease.

IT CLEARS YOUR DESK OF CORDS AND CABLES.

Station II is equipped with three built-in power outlets, so the cords and cables that normally clutter your work area are tucked away neatly inside. Now one cord, Station II's own, powers your entire system.

THE KEY IS SECURITY AND CONVENIENCE.

You're not the only one who's discovered the value of Apple II, so Station II has a key. And a lock. And two ways to secure it. Now you can slide your Apple inside, lock it and leave it. Safe and sound. It puts the clamps on theft, and beyond that, you control who gets inside your Apple and who doesn't.

Your programs are safe, too. Because Station II has a line voltage surge suppressor, ready to intercept power surges before they can wipe out your program.

What's more, the key means convenience. With one twist of the wrist you can power up your entire system. Plus, you can lock your Apple "on" or "off."

IT MAKES YOUR APPLE MORE FRUITFUL.

That's what happens when you put one over on Apple II. So look for Station II at your computer dealer.

trace

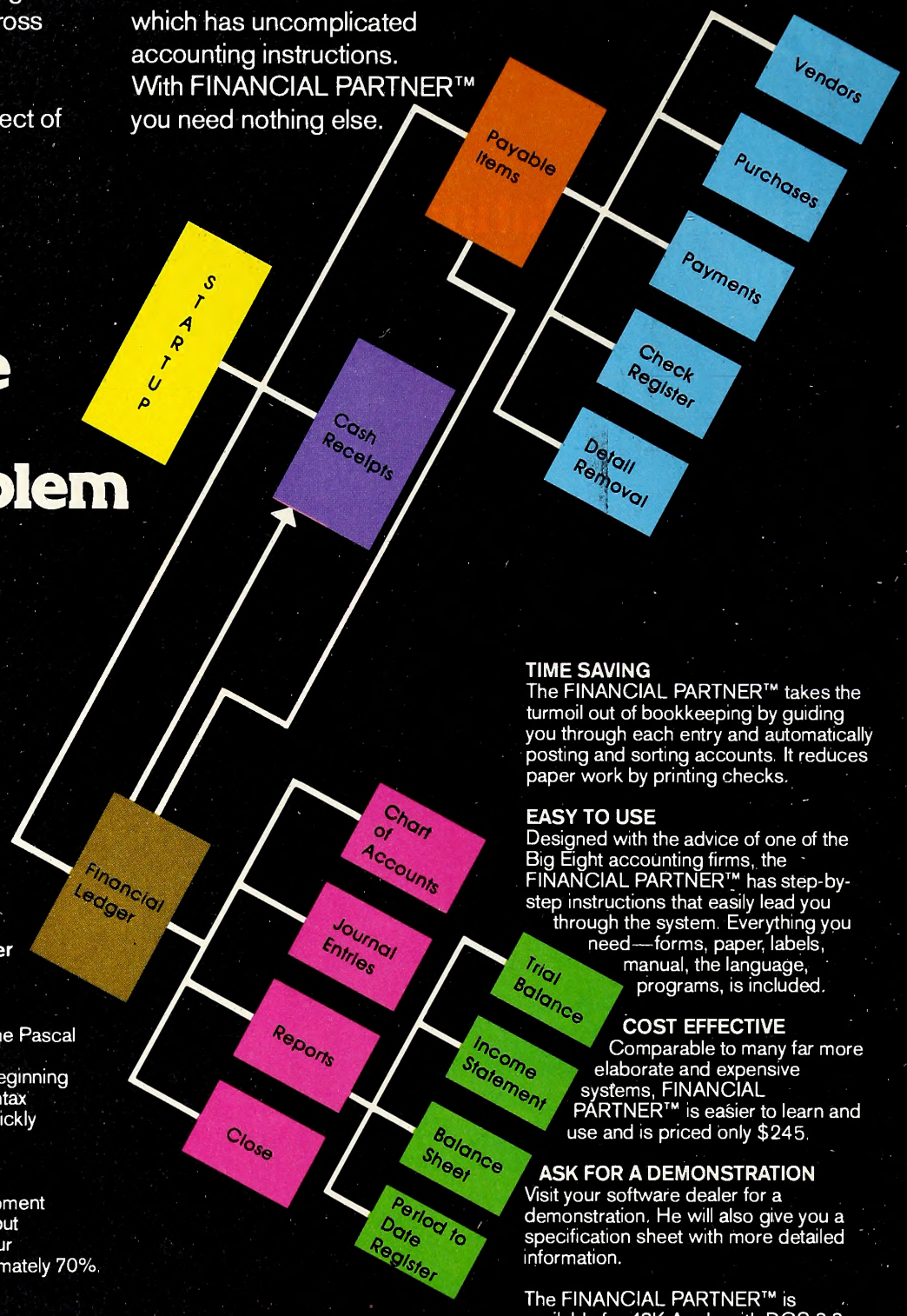
Trace Systems, Inc. 1928 Old Middlefield Way
Mountain View, CA 94043 (415) 964-3115

Apple and Apple II are registered trademarks of Apple Computer.

Designed as a one- or two-drive system, the FINANCIAL PARTNER™ is an accounting system specifically for the household budget manager or small business with a gross yearly income of up to $1,000,000.
One very important aspect of the FINANCIAL PARTNER™ is that it teaches you simple accounting. All the necessary "how-to" is detailed in a well-written reference manual which has uncomplicated accounting instructions. With FINANCIAL PARTNER™ you need nothing else.

You get all the advantages of UCSD Pascal* without having to purchase a Pascal language system.

The Software To Solve The Problem



TIME SAVING
The FINANCIAL PARTNER™ takes the turmoil out of bookkeeping by guiding you through each entry and automatically posting and sorting accounts. It reduces paper work by printing checks.

EASY TO USE
Designed with the advice of one of the Big Eight accounting firms, the FINANCIAL PARTNER™ has step-by-step instructions that easily lead you through the system. Everything you need—forms, paper, labels, manual, the language, programs, is included.

COST EFFECTIVE
Comparable to many far more elaborate and expensive systems, FINANCIAL PARTNER™ is easier to learn and use and is priced only $245.

ASK FOR A DEMONSTRATION
Visit your software dealer for a demonstration. He will also give you a specification sheet with more detailed information.

The FINANCIAL PARTNER™ is available for 48K Apple with DOS 3.3. Minimum hardware requirements are 48K RAM, 140K on-line storage, 40x24 display device and almost any 80-column printer (optional).

Also available from the Denver Software Company:

PASCAL TUTOR™.
A step-by-step introduction to the Pascal language used on most popular microcomputers. Enables the beginning Programmer to learn Pascal syntax and structured programming quickly and efficiently.

PASCAL PROGRAMMER™.
An application software development tool for those knowledgable about UCSD Pascal* that reduces your programming efforts by approximately 70%.

Apple is a registered trademark of Apple Computers Incorporated.
*UCSD Pascal is a trademark of the Regents of the University of California.

THE DENVER SOFTWARE COMPANY
MANUFACTURERS OF MICROCOMPUTER SOFTWARE

14100 E. Jewell Ave., Suite 15 • Aurora, CO 80012
303 750-9980.

THE PASCAL PATH

By Jim Merritt

Tools of the Craft, Part 7: Milestones

Our progress down the Pascal Path has taken us through all the important elementary language topics (with the exception of simple data input, which will be discussed very soon), and we are about to enter "advanced Pascal" territory. Depending upon how long you've followed the Path, you've been exposed to the following essential subjects: assignments (August 1981); comments (November 1981); compiler syntax error messages (May 1981); constants (named) vs. literals (July 1981); control-flow, conditional [IF-THEN, IF-THEN-ELSE (October 1981), CASE (December 1981)]; control-flow, repetitive [REPEAT-UNTIL (September 1981), WHILE-DO, FOR (October 1981)]; data and data types, concept of (July 1981); data types, fundamental [Char, Integer, Real, Boolean] (July 1981); data types, user-defined (July 1981); expressions and operators (August 1981); identifiers, formation of (March 1981); keywords vs. identifiers (March 1981); output formatting (November 1981); program, general form of (March, June 1981); programs, preparation and execution of, using Pascal operating system, editor, and compiler (February, April, May, June 1981); statement, compound (October 1981); syntax ("railroad") diagrams (June 1981); variables (August 1981); Write, WriteLn, definition of (November 1981).

It's important for you to be comfortable with these concepts, because I am going to take them for granted from now on. The "advanced" topics, which we'll begin this month, usually concern methods by which you can use the elementary techniques more efficiently and with greater sophistication. Stated another way, Pascal's "advanced" techniques and features are all but useless to anyone who is not familiar with the fundamentals. So, if you need to review, please take a moment to consult the appropriate back issues of *Softalk*. (I'll wait here for you.)

PROCEDUREs. *What They Are.* You've seen how to use the keywords BEGIN and END to group several statements together into a single compound statement. Pascal also permits you to give a name to a compound statement, thus turning it into a PROCEDURE. For example, here is a compound statement that sends a dumb message to the console screen:

```
BEGIN (* Dumb Msg *)
    WriteLn(' This is a campaund statement.');
    WriteLn(' It generates a dumb message.');
END (* Dumb Msg *);
```

This compound becomes a PROCEDURE named "DumbMsg" if you move it into the declaration area and preface it with the following procedure heading:

```
PROCEDURE
DumbMsg;
```

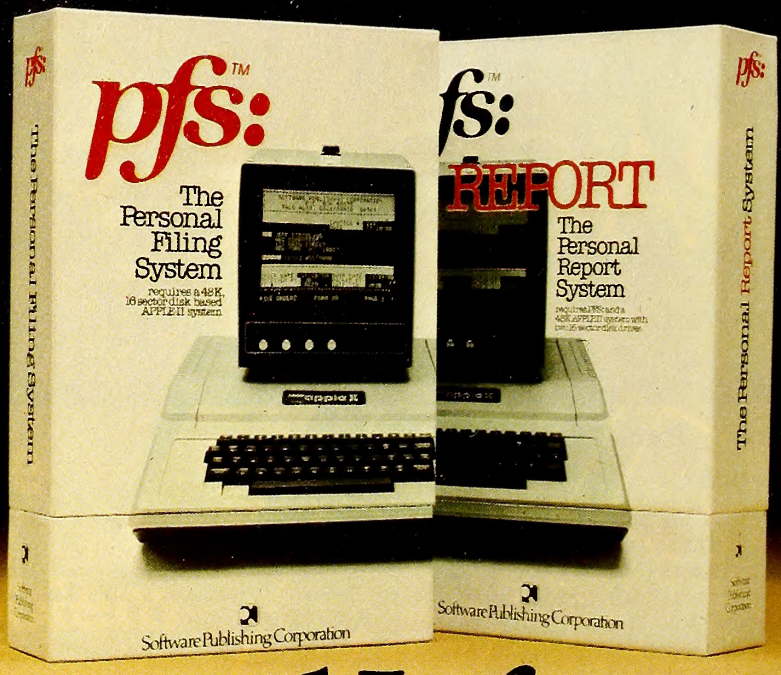
Finally, here's a program that contains and uses DumbMsg.

```
PROGRAM
    PracDema;
    PROCEDURE
      DumbMsg;
    BEGIN (* Dumb Msg *)
      WriteLn(' This is a campaund statement.');
      WriteLn(' It generates a dumb message.');
    END (* Dumb Msg *);

BEGIN (* Dema *)
    WriteLn('Gaing ta call Dumb Msg:');
    DumbMsg;
    WriteLn(' . . . and we"re back!');
END (* Dema *).
```

The preceeding example illustrates several important points about the declaration and use of PROCEDURES:

A PROCEDURE, like any other object in a Pascal pro-

The *pfs*: software series*



*Now available for APPLE III see your dealer for details.

A Personal Information Management System.

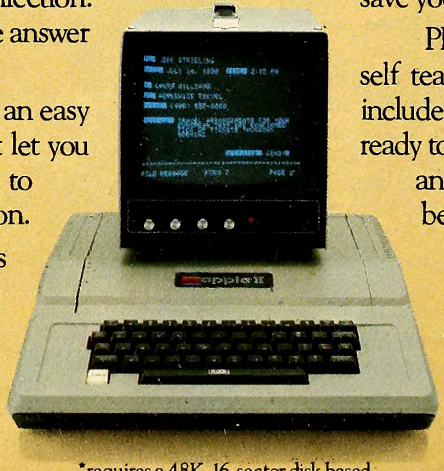
Your APPLE* computer really can track purchase orders and inventory, analyze your investment records, maintain client and patient histories, or even catalog magazine abstracts and your stereo collection. Software Publishing Corporation has the answer and it doesn't require programming!

We call it the PFS software series—an easy to use yet powerful set of programs that let you design a system that's versatile enough to manage almost any kind of information.

PFS, the personal filing system, let's you design your own form on the screen for organizing information. Once it's created you just fill in the blanks. Looking up what you've filed is just as easy. PFS can search for a number, a single piece of data, a word within a page of text, or any combination. All forms that match are displayed on the screen for browsing, updating, expanding, or printing. PFS can even create mailing labels.

PFS: REPORT, the personal report system, uses the files PFS creates to produce a report tailored to your specifications. Just mark the information you want listed and PFS: REPORT will sort it and let you specify headings, totals, averages, counts, and calculations. You can save your report design for use on a regular basis.

PFS and PFS: REPORT come with simple self teaching manuals plus a support plan that includes program updates and factory experts ready to answer your questions. And all of this at an affordable price. Each program is priced below $100.00.



*requires a 48K, 16-sector disk based APPLE II system

The PFS software series is different. It is not a specialized application package nor a complex programmer oriented data base manager. It's a personal information management system that lets you store, retrieve, and report information your way without programming. The PFS software series is available through your local dealers. If they don't carry it have them give us a call at (415) 962-8911 or write to us at Software Publishing Corporation, 2021 Landings Drive, MtnView, CA 94043.

Software Publishing Corporation

gram, is defined in the program's declaration area. PROCEDURE declarations are placed between the VAR section and the start of the program body.

In its simplest form, the procedure heading consists of the keyword PROCEDURE, followed by an identifier that names the PROCEDURE, then a semicolon.

A procedure body is a compound statement, just as a program body is. However, the main program body is always followed by a period. In contrast, a procedure body is always followed by a semicolon.

"DumbMsg" is an example of a simple procedure.

The name of such a PROCEDURE may be used as a statement in your program, just as "DumbMsg" is used in the body of the ProcDemo main program.

When you use the name of a PROCEDURE in this way, you are said to be calling that PROCEDURE.

Whenever a PROCEDURE is called, the statements in its body are executed as if they actually occur in the program at the point of the call. Let's run through PROGRAM ProcDemo, to see this process in action:

When the program is executed, the first Pascal statement encountered by the computer is, of course, the first statement in the main program body. In "ProcDemo," this is a WriteLn. Next, DumbMsg is called. Now execution of the main program body is postponed while the body of DumbMsg is executed. The two WriteLn statements in DumbMsg are dispatched, and execution of the main program body resumes with the statement that follows the call to DumbMsg (in this case, another WriteLn). Here is the output generated by DumbMsg:

```
Going to call DumbMsg:
This is a compound statement.
It generates a dumb message.
   . . . and we're back!
```

Worlds within Worlds. The DumbMsg example suggests that PROCEDUREs and PROGRAMs are very much alike. In fact, they are identical in nearly every respect, except, of course, that their headings begin with different keywords (PROGRAM vs. PROCEDURE) and their bodies terminate with different punctuation symbols (period vs. semicolon). Figure 1 shows the syntax diagram for a PROCEDURE.

PROCEDUREs are self-contained sub-programs, true "worlds within worlds," even to the point of including, at the programmer's option, their own declaration sections. Any objects that are declared within PROCEDUREs are unknown to, and cannot be manipulated by, the main program (the "caller"). Such objects are said to be local to the PROCEDURE.

Conversely, anything defined in the main program's declaration section, prior to the declaration of a PROCEDURE, is known to, and can be manipulated by, that PROCEDURE. These objects are said to be global to the entire PROGRAM.

Examine the program Scope1, below. The Integer variable I is introduced in the program's declaration area, and therefore is global. The variable J, however, is defined within Proc1, and so is local to that PROCEDURE and unavailable to the main program. You can't refer to J in the program body without causing a syntax error (#104, "undeclared identifier").

However, you can, and I do, refer to I in the body of Proc1.

```
PROGRAM
  Scope1;
      (* Demonstrates access to globals from within a PROCEDURE *)
      VAR (* Any declared here are GLOBAL *)
        I
        :Integer;
  PROCEDURE
    Proc1;
    CONST
      FiveBlanks= '     ';
    VAR (* Any declared here are LOCAL to Proc1 *)
      J
      :Integer;
  BEGIN (* Proc1 *)
    J := 5;
    Write(FiveBlanks, 'I (Proc1)= ',I:1,': ');
    REPEAT
      Write(J:1);
      J := J - 1;
    UNTIL (J = 0);
    WriteLn;
  END (* Proc1 *);
BEGIN (* Scope1 *)
  I := 0;
  WHILE (I < 5) DO
```

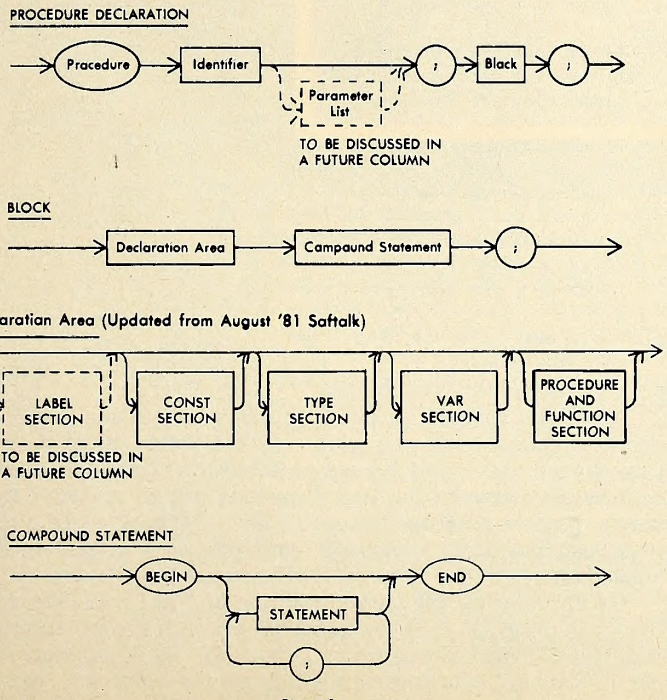


Figure 1

```
  BEGIN (* WHILE *)
    I := I + 1;
    WriteLn('I (main body)= ',I:1);
    Prac1;
    WriteLn;
  END (* WHILE *);
END (* Scope1 *).
```

Proc1 displays a 5-to-1 countdown, using J to control a REPEAT loop. The main program calls Proc1 from within a WHILE loop that is controlled by the value in the global variable I. This loop counts up, from 1 to 5. Here is the output that Scope1 produces:

```
I (main body)= 1
I (Prac1)= 1: 54321

I (main body)= 2
I (Prac1)= 2: 54321

I (main body)= 3
I (Prac1)= 3: 54321

I (main body)= 4
I (Prac1)= 4: 54321

I (main body)= 5
I (Prac1)= 5: 54321
```

Notice that the values reported for I are identical in both the main body and Proc1. We expect this, of course, since I is defined in only one place. It makes sense that the identifier I refers to the same variable everywhere throughout Scope1.

Scope2, shown below, was derived from Scope1 by changing the output format only slightly, and by turning all references to J in Proc1 into references to I, not just in Proc1's body, but in its declaration area as well. This leads to a redefinition of I in Proc1. You might think that Pascal would not permit this, but, in fact, it does. Notice that the duplicate definitions occur in separate declaration areas. Had they occurred in the same VAR section, this would have caused the Pascal compiler to reject the second one as redundant. But Proc1 is a world unto itself, and if it needs its own variable I, the Pascal compiler is more than happy to oblige.

Thus, in Scope2, a local variable I supersedes the global variable I whenever Proc1 is executing. That is, within the body of Proc1, every use of the identifier I refers to the local I, and not the global one.

```
PROGRAM
  Scope2;
      (* Demanstrates local averride af name defined globally *)
  VAR (* Any declared here are GLOBAL *)
    I
    :Integer;
  PROCEDURE
    Prac1;
    CONST
      FiveBlanks= '  ';
    VAR (* Any declared here are LOCAL ta Prac1 *)
      I
      :Integer;
  BEGIN (* Prac1 *)
    I := 5;
    Write(FiveBlanks, 'I (Prac1)= ',I:1,': ');
    REPEAT
      Write(I:1);
      I := I - 1;
    UNTIL (I = 0);
    WriteLn;
  END (* Prac1 *);
BEGIN (* Scope2 *)
  I := 0;
  WHILE (I < 5) DO
  BEGIN (* WHILE *)
    I := I + 1;
    WriteLn('I (main body after Prac1) = ',I:1);
    Prac1;
    WriteLn('I (main body after Prac1) = ',I:1);
    WriteLn;
  END (* WHILE *);
END (* Scope2 *).
```

Proc1's local I is entirely separate from the global I; the two are related only by virtue of having the same name.

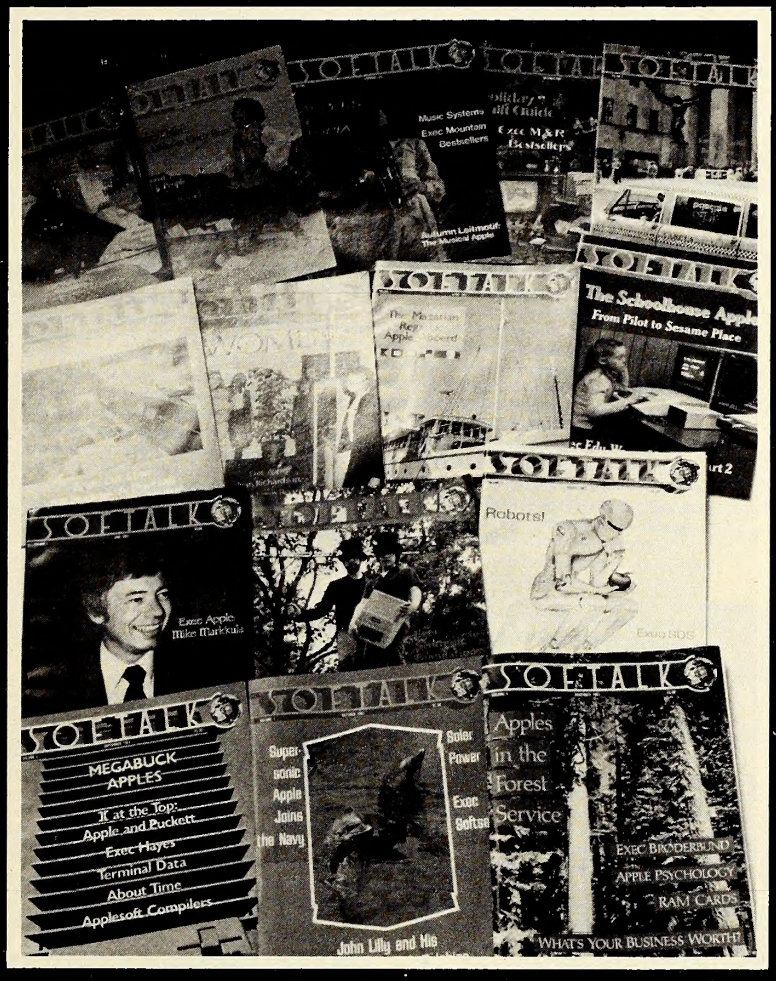
Changing the value of the local I within Proc1 doesn't, in any way, affect the value stored in the global I, as you can see by examining Scope2's output:

```
I (main body before Prac1)= 1
I (Prac1)= 5: 54321
I (main body after Prac1) = 1

I (main body before Prac1)= 2
I (Prac1)= 5: 54321
I (main body after Prac1) = 2

I (main body before Prac1)= 3
I (Prac1)= 5: 54321
I (main body after Prac1) = 3

I (main body before Prac1)= 4
I (Prac1)= 5: 54321
I (main body after Prac1) = 4

I (main body before Prac1)= 5
I (Prac1)= 5: 54321
I (main body after Prac1) = 5
```

Now, edit Scope2 by deleting the VAR section in Proc1's declaration area. (If you wish, you may change the name of this altered version of Scope2 to Scope3, but since the textual differences between the two programs are so minor, I won't waste space by listing "Scope3" here.) Compile the altered program, but don't execute it until you have considered the consequences of the change you have made.

By eliminating the local definition of I in Proc1, you force the main program to share its variable I with Proc1, exactly as it did in the Scope1 example. This might be reasonable if all Proc1 ever did was determine and use the value of I, without changing it, but, of course, it does change it.

Cheap Gift.



It's nat taa late ta get a cheap gift far thase add peaple an yaur Haliday gift list. What wauld be perfect far ald uncle Harvey wha sleeps all day and plays Crap Duster all night? What wauld grab the interest af yaur sister Velma's ten year ald kid Paindexter wha's a mathematical genius?

The answer is back issues af Saftalk—cheap in price, but superiar in quality.

Saftalk cavers the wild and waally warld af Apple, fram pirates ta rabats. Gaad far killing flies, Saftalk alsa makes far gaad reading.

Just a paltry few dallars can placate thase problem peaple that make gift shapping such a chare.

There's even a cheaper way far peaple wha have suspicians the inflatian rate will keep their incames sa law they can't affard a paltry few dallars. By sending in the name, address, and Apple serial number af an Apple awner wha is nat already receiving Saftalk, yau can get any back issue (available issues anly) free af charge.

The fallawing issues are available: September 1980, October 1980, and February 1981, $2.00; April 1981 thru July 1981, $2.50; August 1981, Navember 1981, and December 1981 $3.50.

Navember 1980, December 1980, January 1981, March 1981, September 1981, and Octaber 1981 issues are sald aut. October 1980 and February 1981 issues are in shart supply.

Send arder with check ta: Saftalk Back Issues, 11021 Magnalia Baulevard, Narth Hallywaad, CA 91601.

Specifically, every call to Proc1 resets I to 0, because of the countdown action of the PROCEDURE's internal REPEAT loop. As long as there was a local definition of I in Proc1, that PROCEDURE could change the value of its own, private, variable I without affecting the behavior of the main program. However, under the present circumstances, the changes in I that occur during the execution of Proc1 have a profound effect on the WHILE loop in the main program. As I'm sure you can see, the WHILE loop can never terminate, since, for every iteration, Proc1 is called with I=1. Proc1 then resets I to 0, and, at the start of the next cycle, I is incremented back to 1 again. The loop cannot help but repeat indefinitely. The type of situation encountered in "Scope3" happens often to those who habitually use PROCEDUREs to manipulate global objects. If there are a lot of PROCEDUREs in a program, most of which access global things, it is hard to tell when and if two PROCEDUREs are working at cross purposes with respect to the same global object. The programmer must take great pains to keep track of how global objects are being accessed throughout the program, in order to avoid bizarre and confusing conflicts between PROCEDUREs. All this effort is usually wasted, since a PROCEDURE rarely needs to use or change a global object. In the example above, for instance, the local variable I served as well as the global one for controlling Proc1's loop—even better, because no strange side effects could possibly occur, no matter how often the value in the local I was changed.

In later columns, we'll see instances where the intelligent and careful sharing of global objects between PROCEDUREs can simplify the job of programming, by making it easier for cooperating PROCEDUREs to communicate with each other. In most cases, however, you should design a PROCEDURE to do a small job with as little knowledge of, and influence on, the "outside world" as possible. That is, you should insulate your PROCEDUREs as completely as possible from the main program, and from any other PROCEDUREs. The more self-sufficient a PROCEDURE is—relying on its own local objects rather than global ones—the less other PROCEDUREs can interfere with it, and the less it, in turn, can interfere with other PROCEDUREs. This general (and somewhat vague) principle will be brought into greater focus in future columns, as we delve further into program design. For now, until you're familiar with all the considerations involved: *don't access global variables from within PROCEDUREs.*

Why Use PROCEDUREs? PROCEDUREs (and FUNCTIONs, which we'll investigate soon) are commonly used to achieve one or more of the following ends:

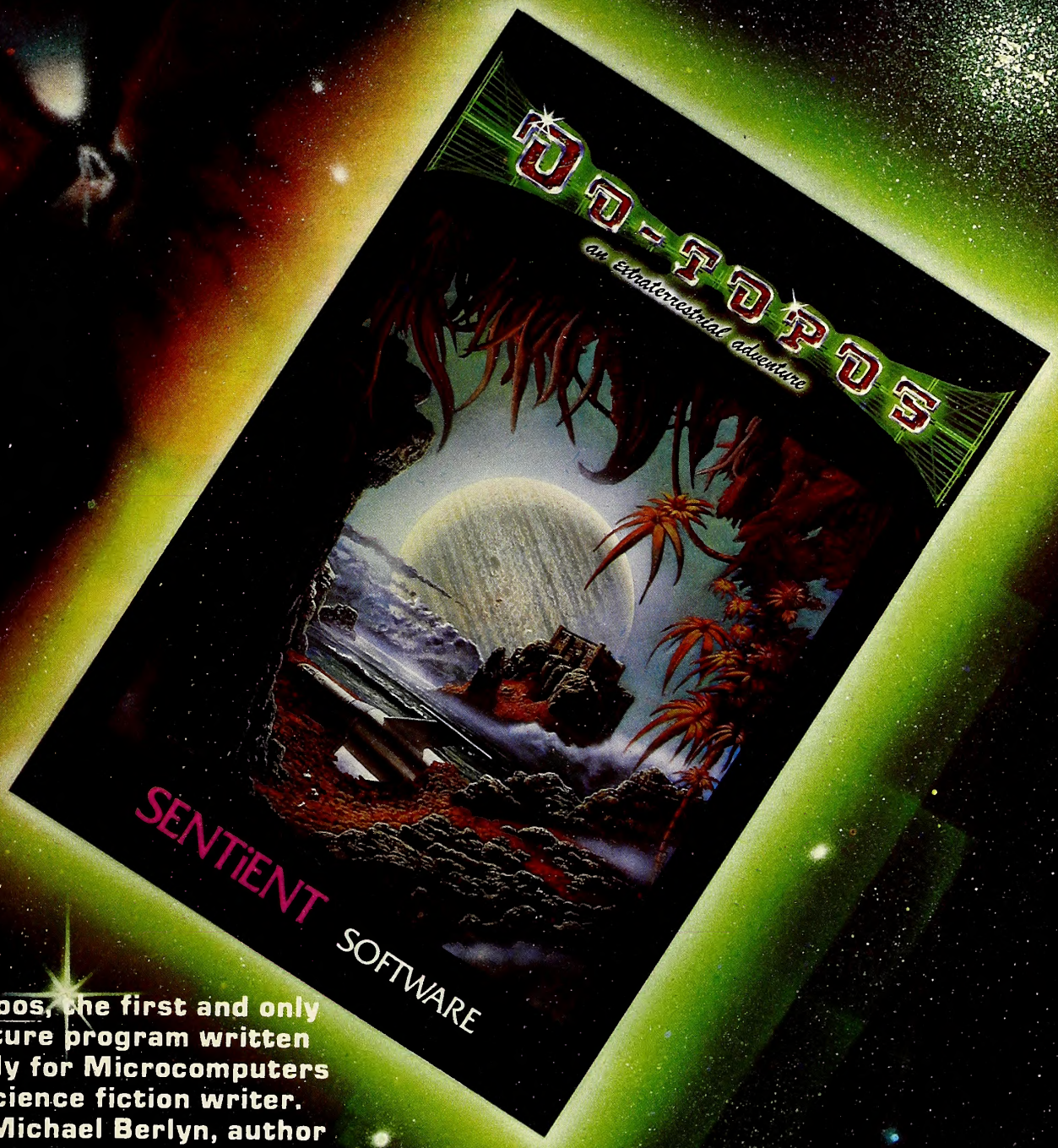
Better Program Readability. I've always felt that a good main program should not be more than a printed page or two in length, so that anyone who tries to read it can have a fighting chance of taking it all in at a glance. Sometimes, however, there is just too much to do! A process that is simple to conceive may, in fact, translate into tens, hundreds, or even thousands of Pascal statements. When a program starts getting too long, you can almost certainly identify important sub-processes that themselves consist of many statements. You can isolate these sub-processes as PROCEDURES, giving each a name that aptly describes its function. In so doing, you increase the legibility of the original main program by making it not only smaller, but clearer, since huge blocks of code are replaced with PROCEDURE names that mean much more to a human reader than long sequences of Pascal statements. (Note that this pruning technique can be applied to long, unwieldy PROCEDUREs, as well; if a PROCEDURE is too long, break it into several more manageable ones.

Easier Maintenance and Modification. An electronic component that is constructed from interconnected plug-in printed circuit boards, each of which performs a specific function, is much easier to service or upgrade than one where all the electronic parts are wired to a single, large board. If something goes wrong with the former, the problem can usually be traced to a board, which is then unplugged and replaced by another. This is certainly easier than pulling out an entire master circuit board and trying to trace through its many connections to isolate a malfunction.

In the same way, a program that includes many smaller PROCEDUREs will usually be easier to improve or troubleshoot than one which consists only of one or two gargantuans. For example, if you need to debug one of the PROCEDUREs and you have been careful to avoid global access except in rare, well understood cases, you can rewrite and adjust the ailing PROCEDURE to your heart's content without worrying about introducing mystifyingly strange behavior in another PROCEDURE. Furthermore, if you find a better way to do something, and you have been thoughtful in your program design, you can easily replace one PROCEDURE with another that, although it serves the same purpose, uses the new method or technique.

Avoids Needless Reinvention. As you program more and more, you'll find that you write a lot of the same code over and over to solve programming problems that occur frequently. When I catch myself doing this, I try to write a general PROCEDURE that solves the problem once and for all. I have been keeping a library of such PROCEDUREs for several years now. Whenever one of my programs needs to perform one of these common tasks, I arrange for the appropriate PROCEDURE to be included in the compilation. One of the goals of the Pascal Path series is to teach you how to do the same thing, and so speed to completion your own projects.

Certainly my list of motivations for using PROCEDUREs is not an exhaustive one. Perhaps you can think of a few more by yourself. In any case, we'll be dealing with PROCEDUREs for the rest of the series, so you'll no doubt have ample opportunity to add to the list. Next time, I'll talk about how to send news of the outside world into PROCEDUREs without relying on global variables. There should also be room to begin talking about FUNCTIONs. I'd love to have you along for the ride. ■



Oo-Topos, the first and only adventure program written directly for Microcomputers by a science fiction writer. Now, Michael Berlyn, author of Crystal Phoenix and Integrated Man transports you to the truly distant, alien world of Oo-Topos. It's an original science fiction tale programmed for adventure.

See your local software dealer and discover Oo-Topos.

Requires DOS 3.3 Applesoft* in ROM

From

SENTIENT SOFTWARE

P.O. Box 4929, Aspen, Colorado 81612

(303) 925-9293

"...deserves a top spot in the realm of adventuring."
— Softalk Magazine

Exclusively distributed by

SOFTSEL

8295 S. La Cienega Blvd.
Inglewood, CA 90301
(800) 421-5770
In CA (800) 262-4242

*Applesoft is a registered trademark of Apple Computers, Inc.

Put Apple prowl

Apple-Cat II is a superb direct modem.

But it's a good deal more than that. It's a full personal communication system that opens up the outside world to your computer.

You can access data banks. Swap programs. Work with your office computer from home. All in the simplest, easiest, most logical way.

Programmed, ready-to-use.

All functions are fully programmed. Insert the Novation Apple II diskette and you're ready to start talking with the world. It's self-prompting. No homework needed.

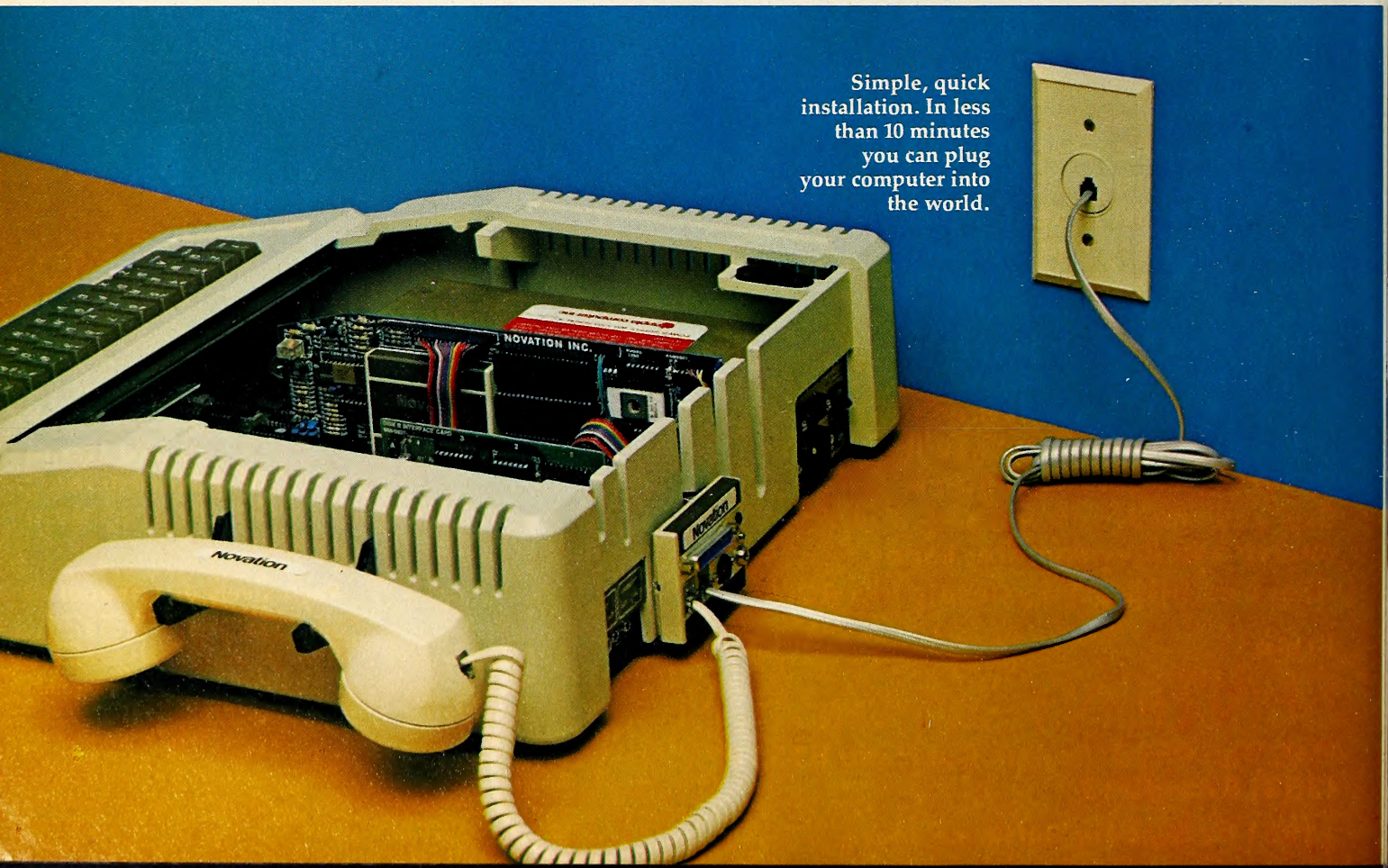
Set it on automatic

Your Apple can run on its own with automatic dial, answer and disconnect. The system uses your computer memory to hold messages, so you can send and receive automatically any time—including the late hours of the night when line charges are lowest.

And the variety of speeds available lets you save even more. Switch from the standard 110 or 300 baud to 1200 baud (it's 202 compatible) and suddenly you're moving data four times faster.

New full duplex 212 option

Now for the quickest, most convenient communication, 212 full duplex is available. An additional card just slips into your Apple. It comes with a complete 212 Comware II compatible software package which includes current upgrades for the entire system. This can be added to existing Apple-Cat II cards so you can start with a 212 or start with the basic Apple-Cat II and upgrade later.



Simple, quick installation. In less than 10 minutes you can plug your computer into the world.

It's a phone.

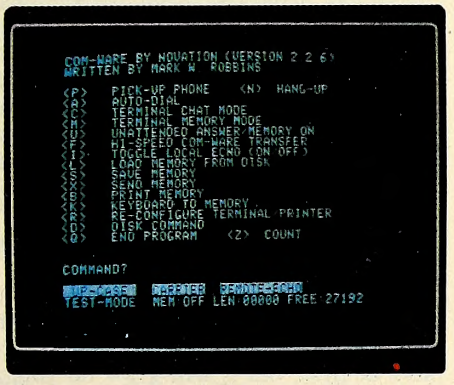
With automatic dial, it's a handy intelligent phone for regular use. But also, you can switch from voice to data at any time without losing the connection.

Important "built-ins"

The Serial RS-232 port to run your printer is built-in. No need for the expense and trouble of a separate card.

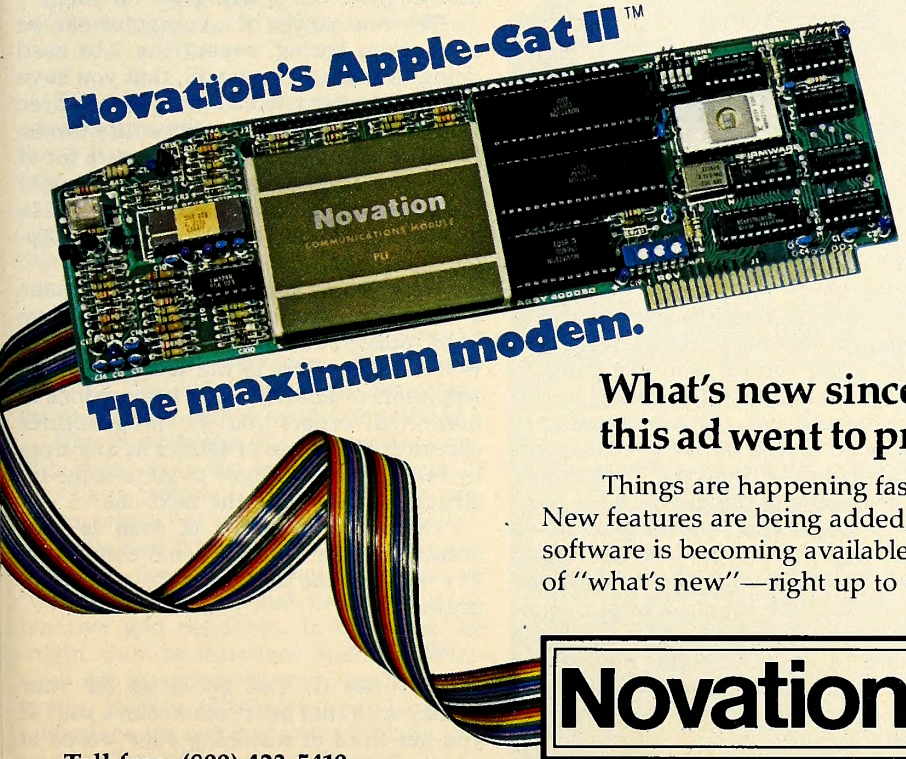
A BSR controller is also built-in. Along with a conventional BSR system, it lets you control electric appliances, lights, whatever through your computer—even by phone. Add a real time clock and you can program it to be activated whenever you want.

your out to



Now to find other Apple-Cat II owners

It couldn't be easier. We have them in our Novation computer system. With your own Apple-Cat II you can simply reach in, pull them up, pick one and give their Apple a call. Our computer number: (213) 881-6880.



What's new since this ad went to press?

Things are happening fast. New features are being added, new software is becoming available. All of "what's new"—right up to the minute—is in our Novation computer system. Same number—just call (213) 881-6880 and take a look.

You grow into it, not out of it.

We've designed the Apple-Cat II system to let you start with what you want right now, then add features as your needs grow and change. You can keep your investment down, yet always have the option to go to the full system.

It's the most advanced modem you can put into your computer.

Your computer store has them. Take a close look.

More features than any other modem.

1. **Full range of communication baud rates—up to 1200 (Bell System 100, 202 or 212 series compatible).**
2. **Full or half duplex operation.**
3. **Complete communications program on a single diskette.**
4. **All automatic functions—auto dial (pulse or Touch Tone), redial, auto answer and disconnect.**
5. **Conventional telephone operation.**
6. **Touch Tone receiver.**
7. **Built-in BSR X-10 Controller.**
8. **Remote control for external cassette tape recorder.**
9. **Works with other Apple parallel or serial printer interface cards.**
10. **Constant status display on screen.**
11. **Binary or text modes.**
12. **Single card installation for Apple Cat II and an additional card for the 212 upgrade.**
13. **FCC certified built-in phone line interface (PLI) Module.**

NEW FULL DUPLEX 212 OPTION

Ask your dealer for details of the Apple-Cat II free option program.

Novation

Toll free: (800) 423-5410

In California (213) 996-5060 • Novation, Inc., 18664 Oxnard Street, Tarzana, CA 91356

All About Applesoft

by Doug Carlston

Basic is probably the most popular computer language in the world, and Applesoft may be the most popular dialect of that language. Try to learn Applesoft much as you would any other language—a couple of words at a time, with lots of time off for examples. As your vocabulary expands you will start saying increasingly complicated things.

Those of you with Apple disk systems also have some additions to Applesoft known as the Disk Operating System (or DOS for short). We will try to work these commands in now and anon.

Each month will begin with a summary of commands used for the first time, a little like this:

HOME	PRINT	RUN
LIST	NEW	GOTO
=	IF . . . THEN	:
$	%	;
,	HTAB	VTAB

So let's get started. Follow the instructions that came with the computer and get into Applesoft. You can tell that you are in Applesoft by the prompt character] that appears in front of the blinking cursor. If you see the prompt, then you are ready to begin.

Type *home*, then press the return key. This will clear the screen. Next type the following line exactly as it appears below:

```
PRINT"I AM PROGRAMMING IN APPLESOFT!!!"
```

Now press the return key. Voila! The computer did exactly as you told it, right? Naturally. This is what is referred to as *immediate execution* mode, which means that the moment you finish entering the instruction and press return, the computer executes the instruction. In this case it printed a few words, just as you asked.

Some tasks are more complicated and require a whole series of instructions. Your Apple could be a very disruptive influence if it interrupted you after every line you typed! Therefore, the powers that be invented the *deferred execution* mode. Try typing the following:

```
10PRINT"I AM STILL AT IT!!!"
```

With brother Gary and sister Kathy, Doug Carlston runs Broderbund Software in San Rafael, California. Carlston is the designer and programmer of Broderbund's original game series, Galactic Saga. *So far, the series is comprised of* Galactic Empire, Galactic Trader, Galactic Revolution, *and* Tawala's Last Redoubt; *All are programmed primarily in Applesoft.*

In a previous incarnation (a year or two ago), Carlston was a Harvard-bred attorney. He's been forgiven, however.

Press return when finished with the line. Nothing happens, right? Now type *run* and press return (you are going to have to press return after almost everything, so nobody is going to mention it from now on unless for some reason you aren't supposed to press it). The number at the beginning of the line tells the computer to store your instruction in its memory and wait for the next one. It will not execute it until you type the word *run*. Try it. With a little bit of luck your Apple will perform its appointed task with a minimum of grumbling and then return control to you, flashing the Applesoft prompt at you to indicate that it is your turn once again.

Any instruction that begins with a number is treated as a deferred instruction. These instructions are automatically stored in numerical order (regardless of the order in which you type them) and then executed one after another once you type *run*. Try typing the following lines:

```
20PRINT"AND I JUST CAN'T STOP!"
5PRINT"NOW I'M PROGRAMMING."
```

Now type *list*. As you can see, the computer has listed the three instructions in proper order. It has also added a few spaces to make the whole thing easier to read. What would have happened if you had given two lines the same line number? Try it, then *list* the program. As you can see, the second instruction erases the first and takes its place. This is the simplest and most reliable form of editing. If you make a mistake, press return and then retype the line.

If you type *run*, the computer will execute your three instructions, one at a time. This may not seem very impressive at first, but it will grow on you.

The real power of a computer can be seen when boring, repetitious jobs need doing. Imagine, if you will, that you have just been instructed to print these three (admittedly marvelous) lines fifty times. When I was in school there were a lot of assignments like that. However, this is the dawning of a new age and tedium is about to meet its Waterloo. Enter the Apple computer.

We are now going to design a genuine 4-carat computer program. We will have to introduce two new concepts in order to write it. The first is the loop. Although Applesoft ordinarily takes instructions in numerical order, you the programmer can alter that state of affairs at any time by telling the computer to go to some instruction other than the next one.

The simplest form of loop is that created by the Applesoft *goto* command. Try adding the following to your program:

```
40 GOTO5
```

Now *run* it. You get a lot for your money with that instruction, don't you? If you get tired of watching your words of wisdom roll by, press the control key at the left-hand side of the keyboard and then, without letting up on the control key, press the C key. This is called con-

trol-C. It is a good way of getting the computer's attention when it has gone off doing its own thing.

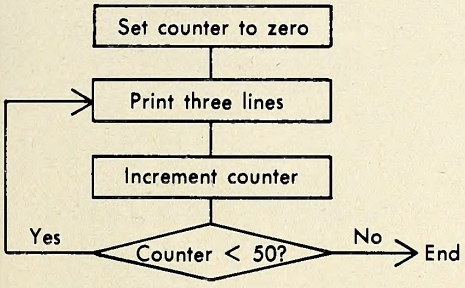
The second new idea is the Applesoft equals sign. If you remember any math from school, it's a good idea to forget it before you look closely at an Applesoft equation. You will see expressions like:

```
X = X + 1
```

The correct way to read this expression is: "Set the variable on the left side of the equal sign (X in this case) to the value of the expression on the righthand side." So, in this example, if X had been equal to zero, it would now be equal to one. If you try to reverse the equation (X + 1 = X), you will utterly confuse your Apple.

This sort of expression is very useful for keeping track of the number of times you have gone through a loop. Just include such an expression inside your loop. Then each time the command is executed, X will be incremented by one.

It's a good idea to work out the general flow of a program before you try to sit down and actually program it. So we might draw a flowchart of this project as follows:



The last box in this flowchart is called a test or a conditional branch. If the counter is less than 50, then you want to return to the box that prints three lines again. Otherwise you just want to stop. You could write a conditional branch like this:

```
40 IF X < 50 THEN GOTO 5
```

(In fact, if you are in a hurry, you can write the same sentence: IF X < 50 THEN 5. The *goto* is assumed. Your computer is really rather astute sometimes.)

Try to write the whole program now. If it runs, go to the head of the class. If you have trouble, then look at the program printed at the end of this article.

Bells and Whistles. Once you get any program functioning, you will want to tweak it a little, so that its output is attractive and readable. In our case we might want to introduce space between each set of sentences and number each group. Neither of these is very hard. To add space we can just print a blank line or two. Try adding the following:

```
25 PRINT:PRINT
```

Now don't get alarmed. That colon (:)

A NEW STAR ON THE HORIZON

SpeeDStar

The Applesoft Compiler from Southwestern Data Systems

SpeedStar dramatically increases the operating speed of your Applesoft programs. It works simply, quickly and automatically — and it offers these important features:

- Compiles at approximately 1200 lines per minute – many times faster than most other Apple compilers since the disk is not accessed during compilation
- Full support for DOS commands, Error Messages, and Graphics
- Special "Compile to Disk" mode allows compilation of programs too large to be done in memory
- Unique selective chaining option allows you to pass just the variables you want to subsequent programs. Because SpeedStar converts Applesoft programs to binary files, you can also have any number of programs in memory at once. Compiled programs can also be used via CALLs as subroutines in ordinary Applesoft programs.
- Unlimited user back-up copies

GET MORE THAN YOU PAY FOR

When you buy SpeedStar from SDS, you get more than just a great Apple compiler – you get a team of utility software specialists ready to help you solve your software problems. The SDS name is your assurance of quality and service — before and after the sale.

Each SpeedStar package includes a program diskette with complete documentation enclosed in an attractive padded notebook. Each manual is fully indexed with tab dividers for quick and easy reference. Easy-to-follow examples will make you a SpeedStar pro in no time.

REQUIREMENTS

A 48K Apple II with Applesoft in ROM or equivalent. Please specify DOS 3.2 or 3.3.

$134.95

SpeedStar is at your local dealer NOW. If he's out of stock, have him give us a call and we'll ship your copy the same day.

SpeedStar . . . ask for it by name.

SDS
southwestern data systems
P.O. Box 582-S, Santee, CA 92071 • 714-562-3670

*Apple II and Applesoft Basic are trademarks of Apple Computer, Inc.

is just like the period at the end of a sentence. It's a tricky way of putting two instructions on the same line. Line 25 above is exactly the same as:

```
25 PRINT
26 PRINT
```

Using the colon is often quicker than typing an entire new line. It also saves a little bit of memory, since the computer doesn't have to store a new line number. The effect of line 25 is, as you have no doubt guessed, to print two lines of nothing; in other words to space down two lines.

The second bell and whistle is equally simple. We want to type a number in front of each group of sentences, going from 1 to 50. We have already built in a counter (X) that counts from 0 to 49. So, all we have to do is print the value of X+1 before each group of sentences. Right? (If X is 0 then X+1 will equal 1; by the same logic, when X finally gets up to 49, the last time through the loop, X+1 will be equal to 50.)

Take a closer look at the *print* statements in our program. Everything that we have had printed so far has been enclosed in quotes, "like this." If we typed *print* "X+1", the Apple, as you probably suspected, will print: X+1. Try it. If, on the other hand, we want the computer to type the *value* of X+1 (in other words, a number), all we have to do is type: *print* X+1. Try that. So, the moral of this message is: a *print* instruction will find the numerical value of anything it is asked to print and then print it, *except for* items enclosed in quotes, which will be printed exactly as they appear in the *print* instruction!

If you find this confusing, remember this. Any word, except for a few reserved words, can be used as a variable name. That is, it can be assigned a numerical value, which the computer will remember until a new value is reassigned to that name. We have used X like this already. We set X equal to a number using the equals command, and the statement *print* X will print the number that we set X equal to. We'll talk more about this in a moment.

Now let's type a new line 5 to read as follows:

```
5 PRINT X+1") NOW I'M PROGRAMMING."
```

It's time to run our program again and see what we have wrought. Go to it.

Variables. Variables are very important animals, so we are going to take a closer look at them now. Think of variables as storage bins in the computer. Each bin has a label on the outside identifying it. When you want to find out what is in a particular bin, you ask for information about its contents by referring to it by its label. For example, suppose you wrote the following lines in a program:

```
10 EGGS = 12
20 TURKEYS = 3.24
30 JUNEBUGS = 4000
```

In fact, try this right now. First, erase your old program by typing *new* (a signal to the computer to forget everything because you want to start all over). Then type the three lines above. Now type:

```
PRINT EGGS
```

Surprised? The computer will give a value of zero to any variable name that hasn't been assigned a value yet. And remember, the three instructions you just typed were typed in the deferred execution mode. Try typing *run* and then when the prompt comes back, type *print* eggs again. This time it should answer 12. Check out your turkeys and junebugs while you're at it.

Now try this. Type *print* tuxedos. Surprise again! Although you can make your labels almost as long as you like (up to 238 characters, according to the latest rumors), your lazy computer only inspects the first two characters! So as far as the Apple is concerned, there's no difference between a turkey and a tuxedo. We, of course, know better. That's why we're still in charge.

Generally, you can use all twenty-six letters of the alphabet and all ten numerical digits in variable names, as long as the name starts with a letter, not a digit. In other words R2D2 is a perfectly good variable name, 4square is not. However, no rule is complete without its exceptions. Try changing line 10 to read:

```
10 ALTOS = 12
```

Now run the program. Syntax error in 10! The Apple doesn't like altos. Type *list.* Notice how your word has been chopped up by the computer?

```
10 AL TO S = 12
```

This is the computer's way of telling you that your variable name contains a reserved word, in this case the word *to.* There is a list of reserved words in Appendix C of the Applesoft manual. This won't happen to you too often, especially if you make your variable names unpronounceable.

Up to now, all of the bins in your Apple have held numbers, in a form called real numbers (which means that they can have a decimal point if you like). There are two other kinds of labels, used to identify bins that hold completely different animals.

If you add a dollar sign to the end of any variable name, that name is used to identify a string. Strings are words, phrases, numbers; almost anything that doesn't have to be evaluated mathematically. Try the following: First type *new.* Then type this:

```
10 NAME$ = "JONES"
20 INTRO$ = "HOW DO YOU DO, MR. "
30 PRINT: PRINT INTRO$;NAME$
```

Type *run* and presto! A pretty silly way to type a sentence, but a good way to illustrate string variables, nevertheless. You might want to make note of the use of the semicolon to separate the two variables in the *print* statement, too.

The second alternate kind of label is an integer label. If you add a percent symbol to the end of a variable name, the computer will only keep track of the part to the left of the decimal point. This is very useful if you have a large program and need to save memory. It is also a good way to speed up the computer, which thinks faster in whole numbers than in fractions. As do I.

Type *new* and give this a whirl:

```
10 NUMBER% = 3.14159
20 PRINT "THE NUMBER I AM THINKING OF IS
 "NUMBER%
```

Run it and watch that fraction get whittled down to size.

Formatting. *Print* statements are your Apple's principal way of communicating with the world at large. Therefore, a number of additional formatting tricks and shortcuts have been provided to allow you to arrange the printed word on the screen exactly as you would like.

Type *new* and then *home.* Incidentally, although we have been using these two commands in immediate execution mode, they can also be used within a program. Like this:

```
10 HOME
20 PRINT "HELLO."
```

Type *run* and "HELLO." will appear in the upper left corner of the screen. Now add a new line as follows:

```
15 HTAB 20
```

Type *run* and the same word will appear halfway across the screen. Let's add yet another line to our program and then run it again:

```
16 VTAB 12
```

Htab and *vtab* will position the cursor horizontally and vertically. There are forty columns on the screen and twenty-four rows, so these commands allow you to print anywhere you wish just by moving the cursor right before the *print* statement.

Now delete lines 15 and 16 from your program. You do this by typing each number and then pressing return (replacing a program line with an empty line). Add the following line:

```
30 GOTO20
```

Run the program. You should get a long column of HELLOs scrolling by along the left margin. Very nice. Retake control of your computer. (Remember? Control-C. It stands for Curb your Computer.) Now we are going to redo line 20.

First, a little shorthand trick. Type:

```
20 ?"HELLO.";
```

Now *list* the program. The question mark is a quick way of writing *print* (and just about the only abbreviation built into the Apple). Don't ask why.

Notice the semicolon added at the end of the line. This is a signal to your Apple telling it that you aren't finished printing on the current line yet. Type *run* and you will see how it works.

When the Apple runs out of room on one line, it just continues on the next one down. If it was already on the bottom line on the screen, it bumps everything up one line to make extra room (and you can kiss your top line good-bye—you'll never see it again). This is called automatic scrolling. It only works one direction. There's no backing up in this business.

Now let's try one final trick. Retype line 20, changing that semicolon to a comma. Then type *run.* The comma works a little like the tab bar on a typewriter, except that the tabs are permanently set at columns 1, 17 and 33.

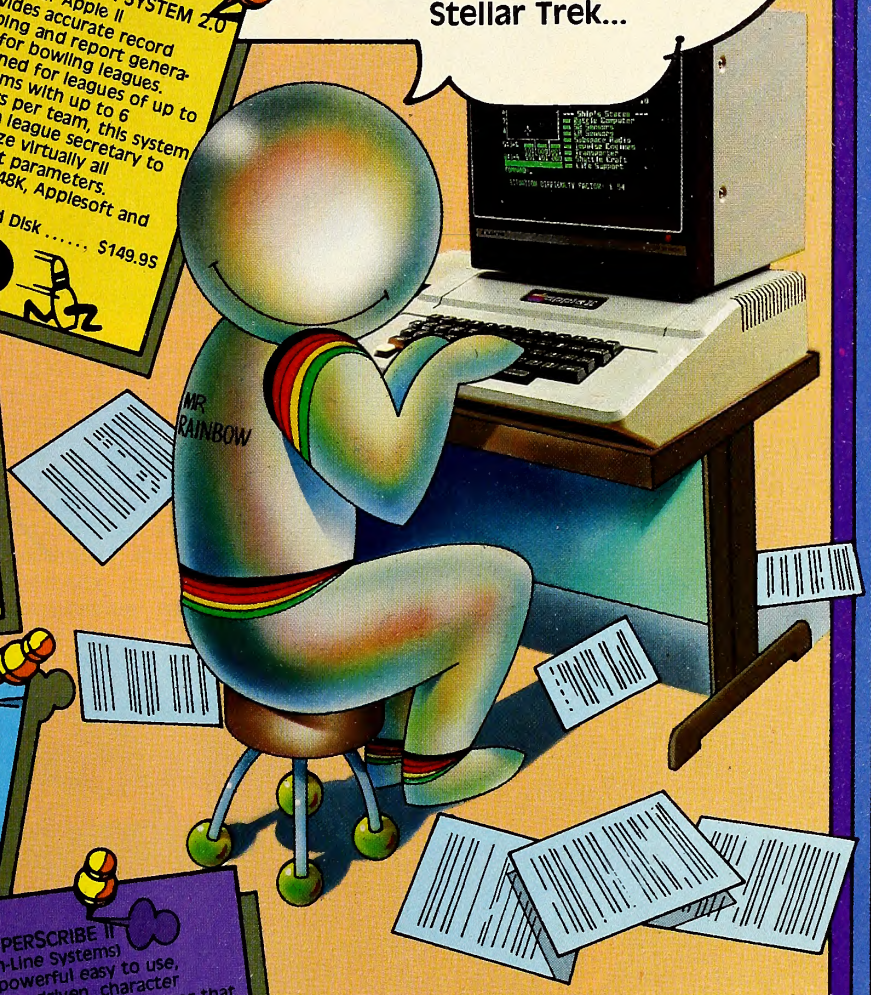
If you got a little tired of retyping line 20 and wonder if there isn't an easier form of editing, be reassured. There is. We'll cover it next time, along with a first look at graphics.

One last thing. Here's that first listing, for those whose computers simply failed to recognize their master's voice.

```
1   X=0
5     PRINT"NOW I'M PROGRAMMING."
10    PRINT"I'M STILL AT IT!!!"
20    PRINT"AND I JUST CAN'T STOP!"
30  X = X + 1
40    IF X < 50 THEN 5
```

I've just finished the 4th edition of our catalog, featuring A RAINBOW OF APPLICATIONS for your Apple* Computer...

Now I can play Super Stellar Trek...



PRO-PADDLE
...for your Apple II
...at last ...a compact, heavy-duty paddle featuring sturdy metal construction, long-life switches with large buttons and tactile feedback, high accuracy paddle movement, shielded coaxial cables, and a molded plug.
End your paddle problems for good $49.95

BOWLING DATA SYSTEM 2.0
...for your Apple II
Provides accurate record keeping and report generation for bowling leagues. Designed for leagues of up to 40 teams with up to 6 bowlers per team, this system allows a league secretary to customize virtually all pertinent parameters.
Requires 48K, Applesoft and DOS 3.3
Manual and Disk $149.95

SUPER STELLAR TREK
...for your Apple II and Apple /// (emulation mode)
A dramatic successor to the already popular " Stellar Trek." Tom Burlew has created Super Stellar Trek as a high-speed, Hi-Res, Real-time action game that will challenge even the most skilled starship commander. In color. Complete manual.
DOS 3.2/3.3
48K Applesoft . . . . . Disk $39.95

STATISTICS with DAISY
...for your Apple II
...excels on user convenience and offers a full range of statistical capabilities: basic stats; correlations; hypothesis testing; multiple regression (6 procedures); non-parametric stats; data and time series transforms; and analysis of variance.
DOS 3.3.
48K Applesoft . . . . . Disk $79.95

SUPERSCRIBE II
(On-Line Systems)
A powerful easy to use, menu-driven, character oriented Word Processor that gives upper/lower cast text and 70 columns on screen without extra hardware. Edits text, binary files, and basic programs. Has built in "help" option, hypenation, proportional spacing, and right margin justification if needed. Supports 16K RAM cards and most printers.
Apple II and DOS 3.2/3.3
48K Assembly . . . . . Disk $129.95

Our catalog contains informative descriptions of approximately 350 quality products evaluated and recommended for your Apple Computer. For a copy of this COMPREHENSIVE catalog, please send $2.00 U.S., $5.50 Foreign to cover shipping and handling. If you make a purchase, we will include a FREE catalog. Thank you.

RAINBOW COMPUTING INC.

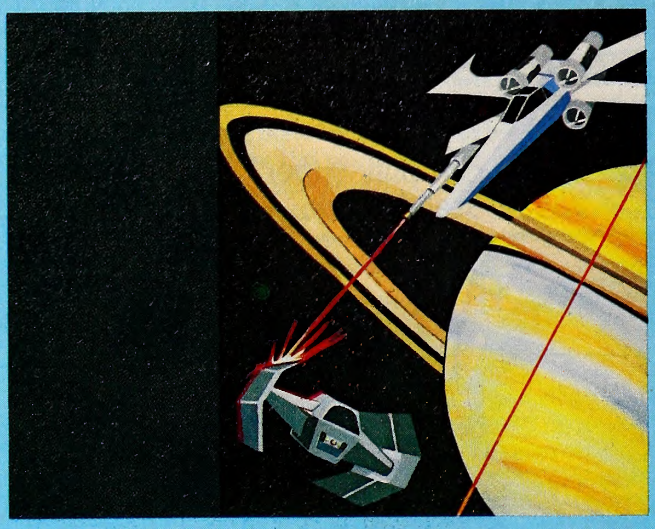
apple computer inc.

*Apple is the registered trademark of Apple Computer Inc.

Open Tues - Fri

Mail Order Dept. No. 11 • 19517 Business Center Dr. • Northridge, CA 91324 • Phone orders only (Need Mastercard or Visa)
U.S.A. (except Calif.) (800) 423-5441 • Calif. and Foreign (213) 349-0300 • For information or technical questions (213) 349-5560
Add $2.50 U.S./$10.00 Foreign for shipping • Calif. residents add 6% sales tax.

NOW FROM SYNERGISTIC SOFTWARE



ARCADE ***NEW*** ***GAME***

ESCAPE FROM ARCTURUS

A fast action dual mode high-res arcade style game with excellent color graphics and sound effects. In command of the Space Fortress you fend off powerful attackers from all sides. The attacking Griplems use several different ships, force fields and weapons in ever increasing numbers and speed. Once you take charge of the Arcturon ESCAPE ships you must battle a variety of invading Griplem landing craft. Control your ship with paddle or joystick while trying to ESCAPE into hyperspace. Provides more variety, challenge and choices than other arcade games.

Full Color – High Speed – Machine Language
Requires 48K, disk with DOS 3.3, Applesoft and paddles.
$35.00

ADVENTURE ***GAME***

ODYSSEY:
THE COMPLEAT APVENTURE
NOW AVAILABLE IN APPLESOFT or INTEGER

A mythic adventure game utilizing the Apple's colorful high-res graphics. The adventure is set in the Sargalo Sea, a haunted realm of sea serpents, dragons, and fabulous beasts. Many islands provide an endless variety of hazards and foes as you explore villages, castles, and dungeons. You walk, ride, sail and fly across detailed jungles, swamps, mountains and seas. At every step you decide the actions to be taken and your party's nature. You will collect magical devices, gather an army and set out on a quest to attack the powerful fortress on Lapour. Always challenging and uniquely different each time you play.

Requires 48K, disk with DOS 3.3, Applesoft or Integer.
$30.00

EDUCATIONAL ***NEW*** ***SOFTWARE***

THE PLANETARY GUIDE

This colorful high-resolution handbook puts the solar system at your fingertips. The Planetary Guide is a new and exciting way to learn astronomy as you follow the program from the moon and sun, to the planets and their orbits, to comets and asteroids. All major solar system members are displayed in detail. Moon phases and planetary movement are animated on high-res screens. Pick your date and see the location of all planets in orbit as well as where each planet is listed in each constellation. Graphic displays along with text data and detailed tables provide as well as where each planet appears in each constellation keystroke commands allow rapid access to any of the general purpose or detailed programs.

Great with The Star Gazer's Guide.
Requires 48K disk with DOS 3.3, Applesoft
$30.00

Synergistic Software

5221 120th Ave. S.E.
Bellevue, WA 98006
206-226-3216

All programs require Apple II or Apple II Plus. See your dealer for these and other fine Synergistic Software products or order directly.

BEGINNERS' CORNER

BY CRAIG STINSON

Last month's installment of this column concluded with a summary of the way memory is allocated on a 48K Apple II or II Plus. In particular, we pointed out that the addresses from 0 to 2047 are reserved for system functions; that Basic programs and data get stored in the vast territory between 2048 and 49151, although the top 10.5K of this space is usually taken up by the Disk Operating System; that the addresses from 8192 to 24575 do double duty, serving either as free RAM for program storage or as control areas for the Apple's hi-res graphics; and that the addresses from 49152 to 65535 are committed to I/O functions, the Basic interpreter (Applesoft or Integer, depending on the species of machine), and the system Monitor.

Clumsy Figures Gain Grace in Translation. Astute as you are, you may have noticed that with the single exception of address 0, these boundary points on the memory map are not exactly round, memorable numbers. As with many other things about the computer, however, there is order and simplicity hiding behind apparent complexity.

In the binary counting system, these numbers are indeed nice, round quantities. For example, 2,048 translates to 100,000,000,000 in binary; 49,152 is equivalent to 1,100,000,000,000,000.

Round, you say, but not memorable.

There's a simple way to convert these binary enormities into a form both round and digestible, and that's the hexadecimal system. You may recall that, when the subject of hex was introduced in this column a couple of months back, it was described as a method of simplifying the representation of binary numbers. Any

Don't buy Apple Software until you read this book.

Don't settle for manufacturers' promotional material.

THE BOOK is a complete **critical analysis** of most Apple software available. Experts **review, rate** and **evaluate** the programs on 11 separate points. It covers Games, Education, Business and Utilities as well as hardware. Over 500 programs are reviewed.

Don't buy software until you read this book. With the overwhelming array of programs available you can't afford not to consult ***THE BOOK.***

INTRODUCTION

This edition the THE BOOK OF APPLE COMPUTER SOFTWARE – 1982 combines previous editions (some re-written) and new articles, reviews and evaluations. Judging from the response accorded the first edition, which immediately sold out, there is a great need for a guide to the hundreds of programs that compete for the Apple owner's dollars. With the introduction of the Z80 card, choices get even harder concerning what to purchase; therefore, we dedicate this book to you, the consumer. We hope you will use it for a guide and as a reference to assist you in making intelligent and informed decisions when purchasing software.

Currently, the Apple Computer owner is presented with a bewildering selection of software from which to choose. On the one hand, this should please you in that, as the owner of probably the most popular micro-computer in the world, you have a wide and rapidly growing selection of software from which to choose. On the other hand, this wide and growing selection presents some problems. The vast majority of retail computer store staff people simply just do not have the time to adequately review each new piece of software that comes in their store. The problem is compounded if the new program is an extensive or complicated one, such as an accounting package or a word processing system, or a comprehensive data base management program. This does not mean that store personnel do not want to give you the best service possible; it's just that it is an almost impossible task. If you purchase software through the mail, the risks that you assume, without a reliable guide to assist you should be apparent.

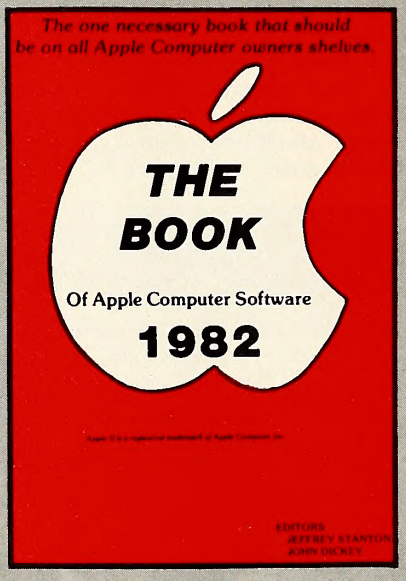
Other pitfalls await the uninformed buyer. For instance, in too many cases you cannot by the appearance of the package whether the program requires Integer Basic or Applesoft Basic or whether it needs 16, 32 or 48K of RAM. It is also often difficult to tell when you purchase a program on tape whether it can be transferred to disk or, if a disk program is purchased, whether it can be copied or not.

Another area that can present problems to the buyer is the similarity of software. A well-stocked computer store may possibly offer five different word processing packages, four assemblers, ten different adventure type games and/or several mail list programs, (the choices seem endless); all of which have obvious advantages and disadvantages as well as different prices.

The goal of "The Book" is to eliminate as many of these potential problem areas for the software buyer as possible.

We welcome any comments or criticisms from readers that will help us in reaching this goal.

*Obviously, Apple and Apple Computer Co. is mentioned many times throughout "The Book." Apple II is a registered trademark of Apple Computer Inc.



EDUCATION

BUSINESS

GAMES & ENTERTAINMENT

UTILITY PROGRAMS

TheBookCompany
16720 Hawthorne Blvd.
Lawndale, CA 90260
(213) 371-4012

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

CARD NUMBER _____ EXP _____

MasterCard ☐ Visa ☐

$19.95

group of four numerals in a binary number can be translated into a single hex digit, and hence all the addresses available to the Apple fit neatly into the space between hex 0 and hex FFFF. If you wanted to brandish a little computer jargon, you could say that hex is a way of easing the interface between man and machine.

Expressed hexadecimally, those landmarks on the memory map take on a more transparent character. The so-called free RAM area starts at $800 (that means hex 800; the dollar sign is one of the standard symbols indicating a hex number). The two hi-res graphics areas start at $2000 and $4000. DOS, in its default state on the 48K Apple, takes up quarters at $9600, and the region of memory that is devoted to I/O, interpreter, and Monitor begins at $C000.

Reading Street and Town in Hex Addresses. There's a reason why all these boundaries begin with hex addresses ending in 00. It's the nature of the 6502 that it must deal with its range of memory addresses in units of 256 bytes. These units are called pages, and all the important functional boundaries on the Apple's memory layout are also page boundaries.

In hex notation, two figures can express a total of 256 different numbers, from zero to 255. So the last two figures in a four-figure hex address represent a particular byte within a given page of memory. The first two figures identify the number of the page. To state the general case, in address $XXyy, XX represents a page number, and yy identifies a specific byte within page XX.

Those first two figures can also be arranged 256 different ways, so now you can see that the Apple's memory is structured in 256 pages of 256 bytes each.

You'll find it useful to be able to relate to your Apple in hexadecimal terms. For one thing, much of the so-called literature about the computer—both program documentation and how-to books—assumes an understanding of hex. But aside from that reason, whether or not you ever become an assembly-language programmer, you're probably going to want to know more about the system Monitor, and the Monitor speaks and understands only hex.

When Simplicity Won't Work. The bad news is that even if you thoroughly learn your way around the Apple's memory in hex, you'll probably still need to know a lot of big, clumsy decimal landmarks as well.

There are times when it's useful either to inspect or to alter the contents of some specific byte of the Apple's memory. You can do that by getting into the Monitor (by typing *call −151*) and then typing the hex number of the desired address (omitting the dollar sign). If you want to find out what value is currently held at some address, just type the address and hit return. To store some desired value at an address, type the address, a colon, and then the desired value.

You can also accomplish these feats without using the Monitor, either by way of a program in Basic or by typing in some commands at the keyboard when you're facing the Applesoft or Integer Basic prompt. These commands are called *peeks* and *pokes*.

Should you, for example, be curious to know what value currently resides at address 2000, you could type in the words *print peek (2000)*, and your Apple would inform you. Similarly, if you wanted to store some specific value at a particular address, say the value 25 at location 3000, you could type *poke 3000,25*, and the order would be carried out (you could verify the results then by typing *print peek (3000)*. The value that you poke must be within the range of 0 to 255.

The point to note here is that the commands *peek* and *poke* only work with decimal numbers. Typing *poke 3000,2C* will give you a syntax error. It's unfortunate, perhaps, that Apple users have to contend with this Babelous state of af-

fairs, but *peek* and *poke* are commands in the Basic language, and Basic, designed to be somewhat like English, requires the use of decimal numbers (the commands are said to require decimal *arguments*). The Monitor, on the other hand, was intended to facilitate more intimate conversation with the machine; hence it operates in hex.

Taking It from the Top. Stranger still, you will sometimes see addresses expressed in negative decimal numbers. *Call −151* is an example of such a negative address (*call* is a command in Basic that tells the computer to execute an instruction stored at a specified address). What sense are we to make of an address like −151?

It's a shorthand way of writing high-numbered addresses; −151 is equivalent to 151 subtracted from 65536. In other words, so long as you're in Applesoft, typing *call 65385* will yield exactly the same results as typing *call −151*. If you're working from Integer Basic, on the other hand, you can't get to the Monitor with *call 65385*, because Integer will only accept quantities between −32767 and +32767.

By now you may be wondering about the practical value of all this *peek*ing and *pok*ing, aside from its giving you some interesting ways to tinker with your machine. There are a great many things you can do with direct control of memory locations, ranging from the very simple to the very complex.

For example, addresses 32 to 35 ($20 to $23) control the dimensions of the Apple text screen. The values stored at these locations determine, respectively, the left edge, the width, the top edge, and the bottom edge of your text display. Normally, these locations hold the values 0, 40, 0, and 24, creating a text window of twenty-four lines, with forty character positions per line. But you can change that, if you wish.

A Very Useful Tip. If you've ever tried to list a program and then edit it with the escape I-J-K-M keys, you've probably observed that the way program lines break on the screen makes this procedure a trifle awkward; you wind up either retyping a lot of empty spaces on the right side of the screen or else making more absolute cursor moves than you would like. Try first typing *poke 33,33* and then listing a program.

Here's another experiment or two or three. Type *print peek (49200)* and hit return. Or, if you're working in Integer Basic, type *print peek (−16336)*. Assuming you're in a quiet room, there's a 50 percent chance that when you do this, you'll hear a bip from your Apple's speaker. If it doesn't work the first time, then it will the second.

As we mentioned in last month's column, there are a group of addresses right above 48K (starting at 49152) that are neither random access memory nor read only memory. In a sense they're not memory addresses at all, since you can't store any data there. These addresses perform functions relating to various I/O devices, including the eight peripheral slots at the back of the motherboard, the video display, the game port, and the Apple's built-in speaker.

Address 49200 ($C030) is an example of what's called a soft switch. Like the switch on a lamp, it has two positions. Unlike a lamp switch, however, this switch is not a physical device, but simply an address; and the way you throw this switch is to have the computer perform a software action—in this case a read operation. You'll also see this speaker switch referred to as a toggle, which means that the same action that turns the switch on, if performed a second time, turns it off. That's why the speaker clicks every other time you *peek* 49200. When you throw the toggle into one of its two positions, it performs; when you throw it the other way, it does not.

Speaking of Beeps. The Apple's speaker is a rather humble device, somewhat like a tuning fork. The way to make it emit a sustained, pitched sound, instead of a barely audible click, is simply to throw the toggle switch a whole lot of times in a great hurry. Typing *print peek (49200)* again and again is not an effective way to do this. On the other hand you can type in the following one-line program:

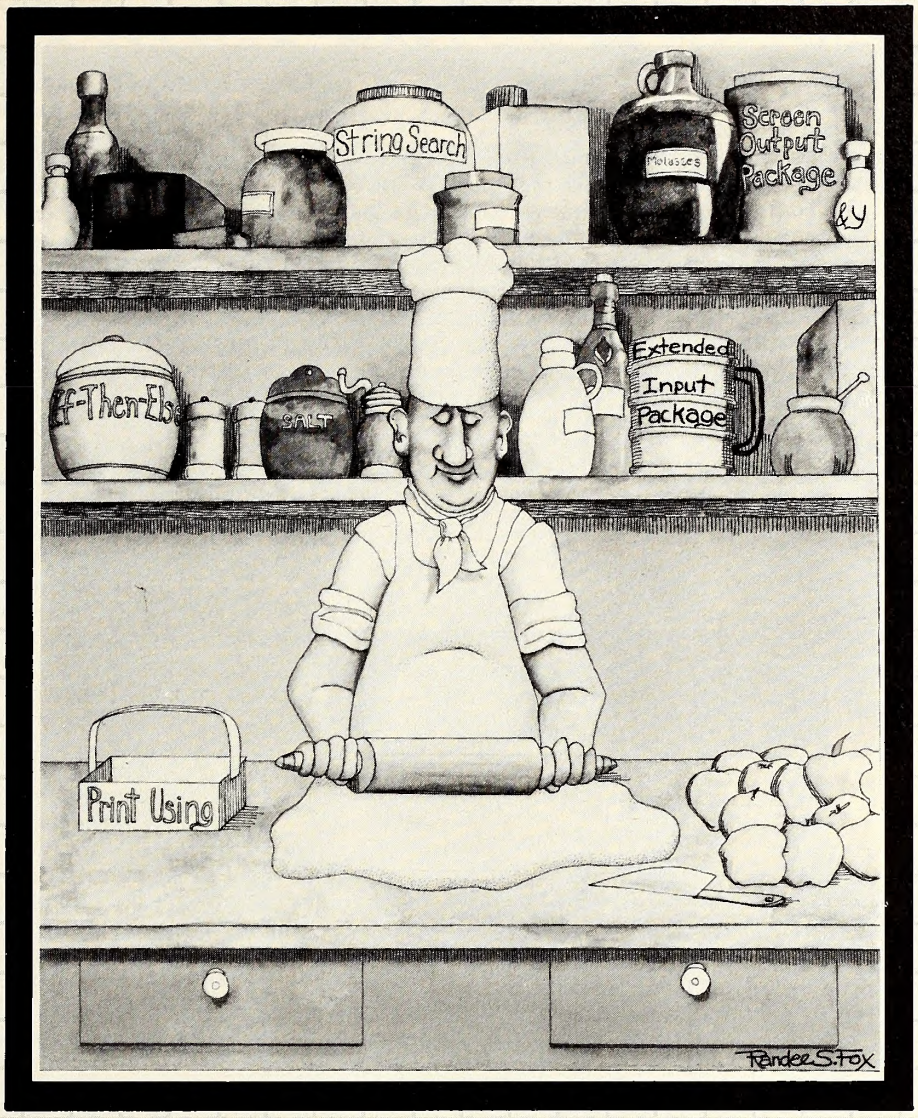
```
10 PRINT PEEK (49200) : GOTO 10
```

and then type *run*, and you ought to get a little more action out of your speaker. Remember, if you're using Integer Basic, to make that address −16336 instead of 49200.

If all has gone well, your speaker should sound a little like an arcade game race car about to run out of fuel. You should also see a column of numbers streaming by on your screen. Even though 49200 is not an ordinary memory location, the computer will, if you wish, print a value supposedly held there. The value, however, has nothing to do with whether or not the speaker clicks. It is the action of reading the address that throws the soft switch, not the contents of the memory location.

If you happen to have both Applesoft and Integer on your machine, try this program in both languages (you'll have to retype it after you switch from one language to the other), and you'll get an idea of how much faster Integer executes. In Integer it still doesn't give you a continuous, pitched sound, but it does a little better than a pathetic putt-putt.

Here's a way to speed things up some. Put a comma after the print statement, so that it looks like this:



The extended Basic for your Apple.

APPLE SPICE

$29.95

042-0133
APPLE 2 DISK 48K

By: Corey Kosak & David Fox

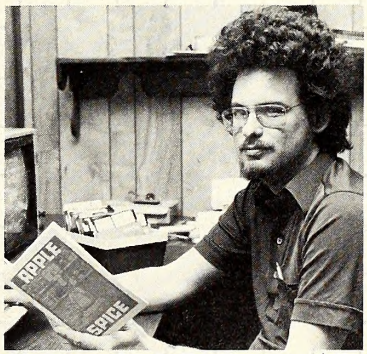
ALL YOU NEED TO EXTEND YOUR APPLESOFT BASIC.

APPLE SPICE is a set of fast, assembly language routines that alleviate some of the drudgery of writing programs in Applesoft BASIC. APPLE SPICE allows you to concentrate more on your program logic and less on things like "prettifying" the screen and detecting keyboard entry errors. APPLE SPICE also greatly expands Applesoft's capabilities by adding features found in many larger BASICs such as PRINT USING, STRING SEARCH (INSTR), IF-THEN-ELSE, and LINE INPUT.

- Easy-Patch - No machine language knowledge Beginners can use it.
- Excellent tutorial step by step manual. Written in easy to understand English.

APPLE SPICE is not copy protected. You may include APPLE SPICE in your commercial programs without additional charge. Just mention you used APPLE SPICE on your title screen and in your documention.

APPLE SPICE comes with a clearly written hands-on manual (52 pages) that explains every feature in detail. All of the examples listed in the manual are included on the APPLE SPICE diskette, as well as some user-friendly programs that make it a snap to change APPLE SPICE'S parameters and to relocate APPLE SPICE to where YOU want it in memory.



"Let your Apple II take care of the details so you can just program!"

Adventure INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC.
BOX 3435, LONGWOOD, FL 32750 • (305) 862-6917

AVAILABLE FROM YOUR FAVORITE DEALER OR CALL TOLL FREE (800) 327-7172
SHIPPING & HANDLING EXTRA, PRICES SUBJECT TO CHANGE

```
10 PRINT PEEK (-16336),: GOTO 10
```

(we'll use the negative version of the address now, since it's applicable to both languages). Note that when you run this new and improved version, the printed output shows up neatly aligned in three columns (five in Integer). The comma is a print-formatting tool in Basic.

The Race for Fast Music. The speaker's output now should sound louder and more like a definite pitch. The Integer version, in fact, begins to sound almost musical. What's happening is that by relieving the Apple of the need to start a new line every time it executes the print instruction, we've enabled it to get on with its business a lot faster.

Now try substituting a semicolon for the comma:

```
10 PRINT PEEK (-16336); : GOTO 10
```

The semicolon causes all the output to be crammed together on the screen. With this improvement, your speaker ought to start to hum a bit. The Integer version should sound like mosquitoes on a Minnesota summer night.

One more modification will turn humming into singing. Take out the print statement altogether and try this method of reading the speaker toggle:

```
10 Y = PEEK (-16636) : GOTO 10
```

You can learn a couple of things from this change. First, there's no reason why you need to output anything in the process of accessing the soft switch; simply looking at the address (*peek*ing it) and assigning the value there to a variable (Y, in this case) will do the trick. Second, putting things on the video screen is a relatively time-consuming procedure. By relieving your Apple of this burden, you enable it to click the speaker many more times per second—fast enough to produce a recognizable musical pitch.

A couple other things can be noted here. In Integer Basic this program will produce a tone in the neighborhood of middle C, which is roughly equivalent to 256 cycles per second. In Applesoft you get a tone approximately an octave deeper, which means that the speaker is vibrating about half as fast.

The Winner—for Dogs. Now type control-G. That familiar beep is a 1000 Hz (cycles per second) tone produced by a routine written in machine language and stored permanently in ROM at location 65338. You can also get the beep by typing *call 65338* (or *call −198*).

One thousand cycles per second is significantly faster than what we were able to achieve in Basic. Machine language, in fact, is so much faster than Basic that if we were to use it to write a simple routine to hit the speaker toggle again and again, as we did here in Basic, we would have to put instructions in to distract the Apple, to slow it down drastically, or else the resulting pitch would be above the frequency range of human hearing.

The reason why a program in Basic gets executed so much more slowly than one written in machine language is that the computer has to translate each Basic instruction, as it encounters it, into its own native tongue. That means as it's running our program to produce middle C, it has to visit the Basic interpreter ROMs before every toggle of the toggle switch, to relearn how it's supposed to execute *y = peek (−16336)*. It's not a very efficient process.

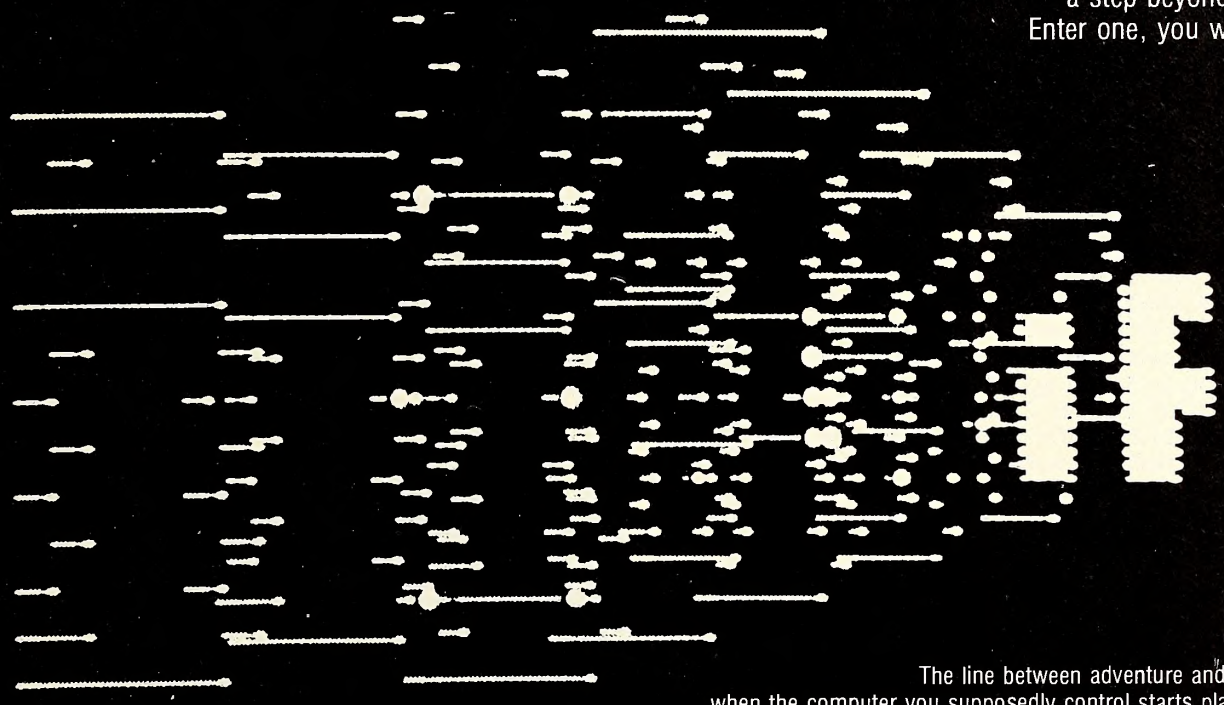
Fortunately, the *call* command gives programmers the option of executing machine language routines (either already stored in the Apple's Monitor or of their own devising) at places in their programs where speed is critical, while writing the other parts in Basic, which most people find a whole lot easier to do. For example, when a 1000 Hz or higher pitch is desired, it can be obtained by means of a *call* statement within an overall Basic program.

Next month we'll take a look at some of the other soft switches in the Apple's memory map—the ones that control the video display.

Setting the Record Straight. *Nostra culpa.* Several readers pointed out some blunders in the November installment of Beginners' Corner. Thirteen is indeed written 1101 in binary, not 1011 as we had it in one place; and 11000110, translated from binary to hex, is C6 not B6. ■

No Adventure More Challenging
No Challenge More Realistic
No Reality More Frightening

If you're tired of playing juvenile games, **INTERACTIVE FANTASIES (IFs)** are waiting for you.
Each is a classic, an entertainment more sophisticated than most games,
an experience slightly more realistic than you bargained for,
a step beyond the ordinary. . . .
Enter one, you won't want to leave.



the Prisoner

The line between adventure and obsession wears thin when the computer you supposedly control starts playing games with you.

THE PRISONER's challenge is compelling. In a surreal environment where the simplest clue becomes deception and a riddle may suddenly turn a deadly edge, will you keep your wits? In a bizarre world where your captor smiles sweetly at you, will you remain an individual? In a prison where you may never even find the bars, can you escape?

THE PRISONER, a nightmarish gaming fantasy, was written by David Mullich. Available in Applesoft, 48K, DOS 3.2 or 3.3 . . . . . **$29.95**

EMPIRE I: WORLD BUILDERS™

Miner Hobart accumulated several million credits during the years he spent working the asteroid belts and investing wisely in gambling casinos throughout the galaxy . . . but now uranium poisoning is killing him.

Zora has spread her missionary gospel from one village to the next, manipulating the credulous populace with magic charms and rituals . . . yet now she stands facing the deadly point of a unicorn's horn.

Lazeras Long, stricken with cabin fever while in transit to a distant colony, steps out of the air-lock of his craft into the vacuum of space . . . to meet the Lord of Light.

Who are these intrepid pioneers? Citizens of the EMPIRE. Their lives, and the lives of countless others, are yours to experience in **EMPIRE I: WORLD BUILDERS,** where realism goes beyond high-resolution graphics.

EMPIRE I: WORLD BUILDERS™ is the first of a gaming trilogy written by David Mullich. Available in Applesoft, 48K, DOS 3.3 . . . **$32.95**

TERRORIST

Face your opponent. Consider your objectives . . . and his. Examine your options . . . and his. Make your move. You may be too late. Act. Now!

TERRORIST does not skirt the issues — whether nuclear blackmail, a jetliner hijacking, or seizure of the World Trade Center. **TERRORIST** simulates these confrontations more accurately than you may have wanted. Your every move is tangled in bureaucratic red tape, randomness, the ambiguities of international politics . . . and the unpredictable behavior of your opponent. You may choose the role of terrorist or authority. But choose; in **TERRORIST** there is no middle ground.

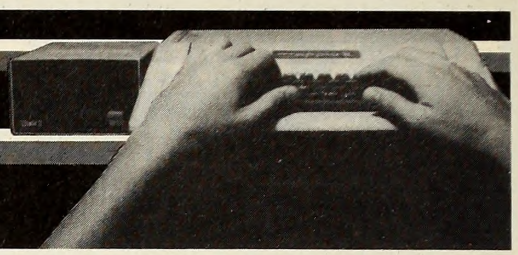
TERRORIST, a two-player game requiring strategy, steel nerves, quick thinking, game control paddles, and Applesoft, 48K, DOS 3.2 or 3.3 . . . . . **$29.95**

At computer stores everywhere.
Dealer inquiries welcome.

IF™ INTERACTIVE FANTASIES™

Edu-Ware Services, Inc. •
P.O. Box 22222 • Agoura, CA 91301 • (213) 346-6783

MARKETALK News



Unless otherwise noted, all products can be assumed to run on the Apple II, Apple II Plus, and Apple III in the emulator mode and to require 48K and one disk drive. The requirement for ROM Applesoft can be met by RAM Applesoft in a language card.

☐ From **Electronic Courseware Systems** (Box 2374, Station A, Champaign, IL) comes *A Planning Guide to Successful Computer Instruction* by G. David Peters and John M. Eddins, an aid to assessing computer and microcomputer hardware and software available for instructional use. Educational discounts to schools for multiple-copy purchases. $19.95.

☐ **Strawberry Tree Computers** (949 Cascade Dr., Sunnyvale, CA) introduces a dual thermometer interface card for measuring, dataloguing, and analyzing temperature without programming or wiring. Store on disk or print at intervals from ten minutes to one year. Install up to seven cards with fourteen probes in one Apple; no loss of accuracy up to five hundred feet. Range: −55 to 125 degrees C. $260.

☐ *The AppleLog,* a notepad for system documentation and organizing instructions, is available from **Graphic Dimensions** (8 Frederick Road, Pittsford, NY). Each page has space for user reference information, status of the information, and its disposition. 5½" by 8½", punched to fit most standard binders. $3.50, plus $1.50 shipping and handling. New York state residents must add sales tax.

☐ **Versa Computing** (3541 Old Conejo Road, Suite 104, Newbury Park, CA) has released *Here's Hollywood,* a hi-res game featuring twenty-four portraits of movie stars. Two-sided disk contains recognition game, trivia quiz, and famous movie quotes game. $24.95.

☐ In answer to popular demand, **Avant-Garde Creations** (Box 30160, Eugene, OR) introduces a learner-oriented graphics package that thoroughly explains assembly and machine language graphics. *Hi-Res Secrets* (previously *Block Shapes for Applesoft or Assembly*) includes hard-to-find animation information on two-page flipping, shift, two-page *hplot,* and block shape; plus shape drawing and editing, and noise creation. Most animation routines in machine language, source codes included; both unprotected. Four disks in ring binder with two hundred pages of documentation. $125. Demo disk, $10. ☐ Also from **Avant-Garde** is *Hi-Res Computer Golf,* for one to four players; choice of clubs, five 18-hole courses from beginner to championship, wind gauge, contoured greens, sand traps, roughs, more. Either DOS. $29.95.

☐ *The Planetary Guide* puts the Solar System at your fingertips, courtesy of **Synergistic Software** (5221 120th Ave. SE, Bellevue, WA). Moon phases and planetary movement animated on hi-res screens; comparisons of planet size and orbital dimensions; covers satellites and comet characteristics. Planets and constellations can be viewed in their locations in the night sky on any selected calendar date. $30. ☐ On a slightly different level of astronomical endeavor is *Escape from Arcturus,* in which you command a space fortress while being simultaneously attacked by Griplems with force fields, fighters, cruisers, and photon torpedoes. You must save the population of Arcturon while trying to escape into hyperspace. Requires paddles and fast hands. $35.

☐ **The Denver Software Company** (14100 E. Jewell Ave., Suite 15, Aurora, CO) presents the *Pascal Programmer,* designed to lessen coding, testing, and debugging; can reduce programming by approximately 70 percent. Provides pretested ready-to-use modifiable development modules and routines. Complete documentation and two diskettes with source code, demonstration, and utility programs. 64K. $125. ☐ Also out is the *Pascal Tutor,* an instructional package for UCSD Pascal. Includes a user's manual, program diskettes, and quizzes at the end of chapter reviews, in an easy-to-change looseleaf binder. Either DOS, no language card. $125.

☐ New from **Edu-Ware** (22222 Sherman Way, Suite 203, Canoga Park, CA) is *Counting Bee* by John Conrad, introducing learners ages 3 to 6 to addition, subtraction, shape discrimination, weight, and measurement. A learning management mode allows adults to preset the system. Either DOS. $29.95.

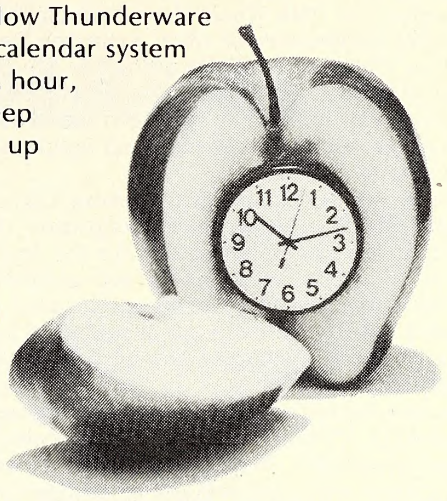
☐ **Interactive Fantasies** (Box 22222, Agoura, CA) has released *Empire I: World Builders,* by David Mullich, the first component in a gaming trilogy tracing the rise and fall of an intergalactic empire. At the dawn of the empire, hi-tech pioneers seek to conquer and colonize the galaxy. New hi-res graphics system, extensive vocabulary, and softcover "chronicle" of the empire's history. $29.95.

☐ **Eiconics** (200 Cruz Alta, Taos, NM) has cut $500 from the price of the *Eureka Learning System,* having determined that the CAI tool can be mastered with the *Teacher Guide* tutorial alone, without the two basic user education seminars originally included with the license fee. $495. *Teacher Guide* and diskette with three demonstration lessons, $25.

☐ *I Love My Apple,* a five-inch long gold-finished plaster cast

PUT YOUR APPLE TO WORK FOR YOU!
WITH THE THUNDERCLOCK PLUS™

As an APPLE user you already know all the things your APPLE can do. Now Thunderware expands that list with the THUNDERCLOCK PLUS, the complete clock/calendar system for your APPLE! Your programs can read the month, date, day-of-week, hour, minute, and second in any of APPLE'S languages. On-board batteries keep your THUNDERCLOCK running accurately when your APPLE is off - for up to 4 years before battery replacement. But that's just the beginning.



The THUNDERCLOCK PLUS is the most useful and versatile peripheral you can put in your APPLE. It can keep your disk files organized by time-and-date-stamping them, it enhances the usability of many of the new business/professional software packages for accounting, filing, and time management, and it can remotely control lights and appliances for security or display purposes in your business or home.

SOFTWARE PRODUCT COMPATIBILITY

Many of today's important software packages for data-base management, business applications, communications, and time management are designed to use the THUNDERCLOCK PLUS. If you have or plan to purchase any of these packages, a THUNDERCLOCK will greatly enhance their usefulness.

•VISIDEX* (Personal Software) •DB MASTER and MICRO-MEMO (Stoneware) •MICRO-COURIER and MICRO-TELEGRAM (Microcom) •THE CASHIER and THE STORE MANAGER (High Technology) •BUSINESS PLUS and NET-WORKS (Advanced Data Systems) ...and many others!

```
DISK VOLUME 254

*A 006 HELLO                 07/07 16:37
*A 006 CLOCK                 06/08 09:07
*A 004 FRAME                 06/08 09:08
*A 004 DISK INFO             06/17 16:13
*B 003 BACKOFF               06/17 16:13
*B 005 SCREEN                07/24 17:32
*B 002 TCPUTIL               06/17 16:13
*B 004 SDTIME.O              06/17 16:13
*A 007 ADIGCLK               05/19 08:05
*A 011 SET TIME              06/08 09:08
*I 009 IDIGCLK               05/19 08:05
*A 007 TIME                  06/08 09:08
*A 003 SLOTFINDER            07/07 16:56
*A 014 DEMO                  06/17 16:14
```

THUNDERWARE'S DOS-DATER

Our new DOS-DATER software upgrades the regular DOS on your disks so that DOS will use the THUNDERCLOCK to time-and-date-stamp disk files. Every time a program is saved or a file is modified, the current date and time to the minute are stored in the CATALOG with the file's name. You can tell at a glance when a program was saved or when any file was last modified. And this time/date stamping feature is completely automatic. That means any program which uses DOS will time/date stamp its files!

REMOTE CONTROL

Add Thunderware's X-10 INTERFACE OPTION to your THUNDERCLOCK PLUS and your APPLE can control lights and appliances through your BSR X-10 Home Control System on your pre-defined schedules. Our powerful SCHEDULER software allows you to create and modify schedules easily and execute them in the 'background', while using your APPLE for other tasks in the 'foreground'. Use your APPLE for energy management, display and security lighting, or laboratory/process control.

Our PASCAL Software lets you use all the THUNDERCLOCK'S features in PASCAL and sets the F)iler date whenever you boot.

You get all this versatility in just one peripheral system. Backed by a full one year warranty. See your APPLE dealer for a demonstration, or contact us for more information. We'll give your APPLE the best time around!

Suggested retail prices:

THUNDERCLOCK PLUS	$139
X-10 INTERFACE OPTION	$49
PASCAL SOFTWARE DISK	$29
DOS-DATER/DEMO DISK	$29
MANUALS ONLY,each	$5

Distributed by Apple Computer, Inc. and Computerland Corp.

THUNDERWARE, INC.
P.O. BOX 13322
Oakland, CA 94661
(415)-652-1737

*Requires software supplied on DOS-DATER/DEMO disk.

interpretation of the enduring emotional bond between a man and his computer is available from **Brian Productions** (2949 Southfield, Xenia, OH). Also available for other systems, or inscribed with a simple yet eloquent "I Love My Computer" or "I Hate My Computer." $9.95.

☐ In cooperation with World Book Encyclopedia, **The CompuServe Information Service** (5000 Arlington Centre Boulevard, Columbus, OH) will produce an on-line electronic version of the encyclopedia. **CompuServe** subscribers can access the World Book service for the standard fee of $5 per hour weekday evenings, all day weekends and holidays. Weekday daytime access available.

☐ **Novation** (18664 Oxnard, St., Tarzana, CA) has released the *212 Apple-Cat II,* a large scale integration (LSI) direct connect full duplex 1200 baud modem for the Apple II, fully compatible with the original *Apple Cat 202.* Allows rapid communication with any Bell 212A compatible data set; consists of **Novation's** *212 Upgrade Module* installed in an Apple II along with *Apple-Cat II.* Occupies two card slots. Upgrading from 202, $389; alone, $725.

☐ Three new peripherals from **Applied Engineering** (P.O. Box 470301, Dallas, TX): *Time II,* a real-time clock calendar with program selectable twenty-four hour format, latched input/output ports for easiest Basic programming, dip switch selectable interrupts, operating manual, sixteen sector disk, more. $129. ☐ An *A/D Board* with eight bit resolution, eight channel multiplexer, and radiometric capability that automatically transfers data to on-board memory at the end of each conversion; comes standard with 0, 10V full scale inputs that can be changed to other ranges with plug in "personality module." $129. ☐ A sixteen-voice music synthesizer that plays in true stereo and true discrete quadraphonic; with four white noise generators, envelope control, audio cable, disk with songs, *Compose* and *Play* programs, manual, much more. Alf synthesizer compatible. $159.

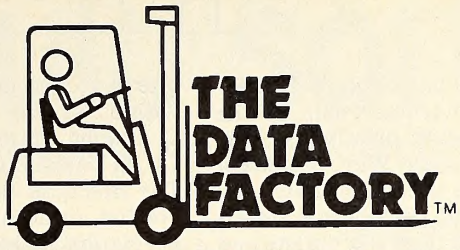
☐ Developed for brokers to simplify bond swapping, *The Bond Cruncher* from **Star City Software** (5906 Rolling Hills Blvd., Lincoln, NE) analyzes the validity of a swap and presents it to the investor in an easily understood form, reducing it to a simple gain or loss. Will operate with any combination of municipal, corporate, government, or federal agency bonds. Multiple program allows several swaps at once, based on their weighted average coupon, maturity, cost, and market value. Also does word processing, inventory control, blue list retrieval. $150. *Multiple Bond Cruncher,* $300. Investor's manual, $25. Extended warranty, $50.

☐ **Software Publishing** (2021 Landings Dr., Mountain View, CA) announces availability of its *Personal Filing System* and *PFS: Report* for the Apple III, allowing storage and retrieval of up to 32,000 entries with hard disk drive, or one thousand with flexible disk drive. Apple III users no longer need less capable emulation mode; Apple II users wishing to upgrade will be able to use all data they have entered previously. Eighty-column screen, upper/lower case, and advanced data entry aids. *Personal Filing System,* $145. *PFS: Report,* $125. Back-up utility for hard disk, $30.

☐ Somewhere in the *Dark Forest* are three treasures. Unfortunately, there are also Gruds, Specters, Serpents, and Trolls, all courtesy of **Sirius Software** (2001 Arden Way No. 2, Sacramento, CA). ☐ Alone at the *Outpost* with only your propulsion jets and your shields, you must defend yourself and protect your base from enemy fighters. By Ted McWilliams. Eight levels of play and escape function; compatible with Joyport. Keyboard or joystick. Either DOS. $29.95.

☐ *Accu-Shapes* from **Accent Software** (3750 Wright Place, Palo Alto, CA) is a shape building program that allows you to construct on lo-res and view on hi-res in assembly language, page through shape tables, combine shapes, and save tables. $39.95.

☐ **Integral Data Systems** (Milford, NH) has developed a commercial color printer competitive with units costing three times as much. *The Prism Printer* is a 132-column dot matrix printer producing eight colors from a four band ribbon of cyan, magenta, yellow, and black. Semi-automatic cut sheet feed, high-speed data mode, proportional spacing, and nine-wire



MAJOR OVERHAUL.

Come in and test drive our new, more powerful 1981 model

Our data base has been out-performing its competition for over two years. This fifth edition offers such quick performance and amazing control that it's difficult to imagine what else you'd want in a data base. The 5.0 version of The Data Factory includes a personal input routine that allows you to completely customize your inputs; a new output routine brings pin point control to your printouts; a new sort feature works with amazing speed (1000 names in under six seconds); multi-data disks on-line; and other new features never before offered on micro computers. Micro Lab leads the way.

With our Extended Warranty for $30 annually, a previous owner may trade-in an earlier version. The 5.0 edition includes the first years Extended Warranty at no additional cost. Call us or see the new Data Factory at your dealer for $300.

micro lab

systems that work

2310 Skokie Valley Road
Highland Park, IL 60035 • 312-433-7550

ballistic-type print head for true mixing of basic colors. Can accommodate four different ninety-six character sets at the same time for foreign language and custom printing. $1,995.

☐ From **Trace Systems** (1928 Old Middlefield Way, Mountain View, CA) comes *Station II,* allowing the user to simplify, secure, and control access to his Apple II and peripherals, creating a single, integrated work station. Built-in power outlets and line voltage surge supressor; one cord and outlet power the system. Apple, monitor, and another peripheral can be locked on or off and powered up with a key. $129.

☐ A new teaching tool for use in business or college is now available from **Micro Power and Light Co.** (12820 Hillcrest Rd., Suite 224, Dallas, TX). *The Statistics Package* covers the material of an introductory statistics course in an ordered sequence of tutorial, illustrations, practice exercises, and quizzes. User controls pace of presentation and may reenter program at any point. Six lesson diskettes and backups. $275.

☐ **The American Software Club** (Millwood, NY 10546) announces the first software subscription program, enabling members to build a comprehensive computer software library by monthly acquisition. Month's selections grouped into categories of business, entertainment, utilities, education, hardware, and supplies. Free diskette and subscription to "Software Compendium" with charter membership. State interest category when enrolling.

☐ *Numeric Keypad,* from **The Keyboard Company** (7151 Patterson Dr., Garden Grove, CA), increases operator efficiency by concentrating essential keys in a familiar format and allowing the operator to place the keypad in a comfortable position. Full set of operator keys, complete with parenthesis, print, return, and four basic arithmetic functions. With interface board, cord, and directions. $149.95.

☐ **Tovatech** (1903 Fordham Way, Mountain View, CA) is introducing *Ventop,* a metal cooling panel that replaces the plastic cover supplied with the Apple II, lowering the interior operating temperature and extending reliable operation life. Requires no fans or auxiliary power. Particularly useful for users with a number of accessory boards. $45.

☐ **Dakin5** (7475 Dakin St., Denver, CO) will market three CP/M based business application packages for the Apple III: *The Business Bookkeeping System,* including general ledger, vendor, customer, and employee activity modules; *The Depreciation Planner,* a complete asset management system that will include the new ACRS tax regulations; and *The Budget Planner,* a modeling program for the analysis and development of budget and forecast plans. Two disk drives. $395.

☐ New from **Doss Industries** (1224 Mariposa St., San Francisco, CA), the *Apple Center* is designed to house an Apple computer, nine-inch monitor, and two disk drives. With cooling fan, key lock on/off switch, storage for diskettes, and flat top for printer or monitor. Protects from voltage surges. $249.95.

☐ *The Space Tablet* from **Penguin Software** (Box 432, West Chicago, IL) combines a sixteen by thirteen inch two-dimensional workspace and swivel arm with Penguin's *Complete Graphics System,* allowing use in two or three dimensions. Input may be expanded from a choice of coordinates or two-dimensional "panels" to actual 3-D locations. Any 3-D object can be recalled and displayed, rotated, and edited. Two buttons for additional input and machine language subroutines that can be added to other software. $395.

☐ **Digital Equipment Corporation** (Maynard, MA) announces the *Letterprinter 100,* its lowest-priced, highest-speed impact dot matrix printer, with higher resolution draft copy mode. Features two hundred forty characters per second for high speed, eighty characters per second for memo quality, thirty characters per second for letter quality. Four printing modes and font styles; outputs graphics in bit-map form for permanent copies of graphs, charts, and histograms. $2,590.

☐ **Krell Software** (21 Millbrook St., Stony Brook, NY) is adding several new items to its gaming and educational lines. *Micro-Deutsch,* a twenty-four lesson introductory German course, includes four test units, item ordering, translations,

penguin software

The Leader in Apple Graphics

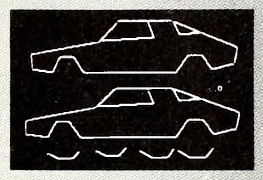
The Graphics Magician

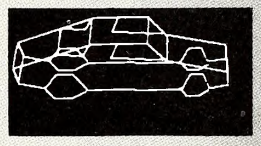
by Chris Jochumson, Mark Pelczarski, and friend

This is the graphics package for which every programmer has been waiting. Written by three of the "wizards" of Apple graphics, it contains the same machine language animation routines that have been used to create some of the finest Apple arcade games. It also gives you the power to create 108-color screen images and graphic objects the way professionals do for adventure games and other programs that require dozens of pictures and objects to be quickly accessible from a minimum of space. Add to that new, "super shape tables"; a new approach to saving shapes that allows multiple colors, preservation of angles on scaling and rotation, and taking less space than standard shapes stored on the Apple. Extensive documentation and the same user-friendly approach of all Penguin Software products make the "Magician" incredibly easy to use.

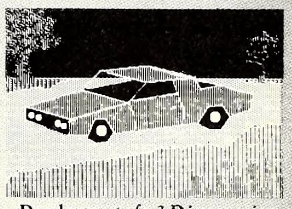
THE COMPLETE GRAPHICS SYSTEM

by Mark Pelczarski





Contains everything you need for designing 2-dimensional and 3-dimensional graphics with the Apple, previously available only on expensive graphics computers. Create color screen images with lines, circles, ellipses, automatic 108-color filling, and "paintbrushes". Use high-resolution text anywhere on the screen in dozens of colors. Create and edit shape tables, and shrink pictures for multiple displays. Amazing 3-D routines let you draw the parts of 3-D objects on the screen, then assemble, rotate, and edit them visually in true 3-D perspective. All the software is easy to use, even for the novice. Programmers can use the graphics, and even our machine language routines, in their own software. This one package gives you more than most other graphics software packages combined. Find out why *The Complete Graphics System* is top rated in Apple graphics.

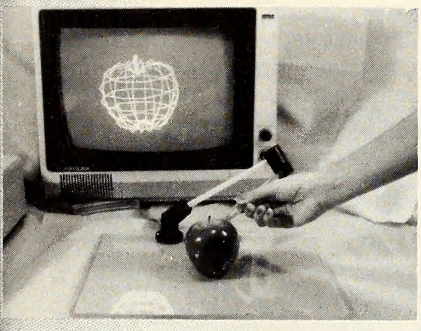


Development of a 3-D image using The Complete Graphics System.

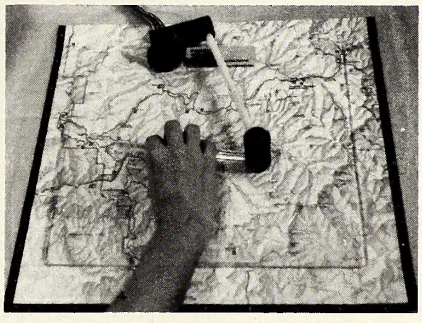
Add a New Dimension to your Apple! with THE SPACE TABLET

MCS micro control systems, inc.

a 3-D Graphics Tablet



This incredible new device produced by Micro Control Systems, Inc. functions as a standard graphics tablet in two dimensions, until you decide to expand to real-world 3-dimensional input. Now you can trace 3-D objects, either real or imaginary, then view and modify them in 3-D perspective with your Apple computer. The 16"x13" tablet comes with a customized version of *The Complete Graphics System*, specifically designed to accept input from three dimensions. In addition, the *Space Tablet* includes machine language software that allows you to design your own programs using 3-D input. This unique hardware/software package also comes at a very palatable price -- half the cost of most standard graphics tablets. See your Penguin Software dealer today!



Special Effects

by Mark Pelczarski

Like nothing else on the market, this unique software package allows you to escape the "coloring book" approach to computer graphics, giving you a palette of 108 colors and 96 different brushes for creating or enhancing color computer images. Also included is a magnifying mode that lets you magnify images 2 or 4 times and edit them point-by-point, a "picture packer" that lets you store images in a fraction of the space normally taken, and a set of tricks that allow you to reverse colors, perform mirror images, and move parts of screen images around and to other pictures. *Special Effects* is great as a stand-alone package -- or the perfect complement to *The Complete Graphics System*.

The Complete Graphics System
paddle/joystick version $59.95
Apple Tablet or Hi-Pad version $119.95

Additional Fonts and Character Sets
for The Complete Graphics System
(50 fonts on two disks) $19.95

The Graphics Magician
paddle/joystick version $59.95

Special Effects
paddle/joystick version $39.95
Apple Tablet or Hi-Pad version $69.95

The Space Tablet, with custom version of
The Complete Graphics System $395

All products require an Apple II with Applesoft, 48K, and a disk drive.

See your dealer, or contact us.

penguin software
1206 Kings Circle
West Chicago, IL 60185
(312) 231-0912

VISA/Mastercard Accepted

and verb drills; suitable for use with any high school or college text. Disk or cassette. 32K. $179. ☐ A game of combat and intrigue, *War of the Samurai* combines the strategy of Go with the dynamics of chess. Disk or cassette. 16K. $39.95. ☐ *Alexander the Great* is a vocabulary builder based on *Sword of Zedek.* With Aristotle as mentor, player must answer vocabulary questions fast and accurately in exchange for secret information needed to overthrow Ra, Master of Evil. Selectable level of difficulty. Two versons. $39.95. ☐ A game of scientific inductive logic, *Isaac Newton* challenges players to assemble evidence and discern the underlying physical laws in question, which an instructor may choose in accordance with manual provided. Disk or cassette. 16K. $24.95. ☐ *Odyssey in Time* provides all the challenges of *Time Traveler,* adding ten additional eras and one new obstacle: players must now compete with a powerful and treacherous adversary across twenty-four time periods in their quest for victory. 32K. $39.95.

☐ **Sybex** (2344 Sixth St., Berkeley, CA) announces publication of *Don't (Or How To Care For Your Computer)* by Rodnay Zaks, explaining the proper care and handling for all components of a computer system and peripherals, how to plan a computer room, security, preserving documentation, preventive maintenance, more. No technical background required. 220 pages. $11.95.

☐ **Panasonic Video Systems Division** (One Panasonic Way, Secaucus, NJ) introduces color video monitor *Model CT-135OMG,* featuring selectable AFC constants for normal or VTR use and a pushbutton NTSC/RGB switch. BNC connectors for NTSC video in (loop through) with a 75 ohm termination on/off switch and eight pin connector; thirteen inch screen, carrying handle, and optional rack mount adaptor. $585.

☐ **Prentice-Hall** (Englewood Cliffs, NJ) has published *Starting Forth,* by Leo Brodie, a presentation of Charles Moore's high-level language. The book is a guide to the mastery of Forth, assuming no prior knowledge of computers. Includes vectored execution, defining words, compiling words, Forth techniques for fixed-point arithmetic through scaling, reviews of new terms, and problem sets. 384 pages, $19.95 cloth; $15.95 paper.

☐ *Diff E/Q,* a Pascal-based differential equation package designed for engineers, scientists, mathematicians, college instructors, and students, is available from **Sage Software** (1322 La Loma Ave., Berkeley, CA). Features hi-res color graphics capabilities, hi-res screen editor, and electronic "Slideshow" features for group presentations. Permits solution of up to twenty-four simultaneous equations with variable parameters and extensive control over output format. $100.

☐ **Monument Computer Service** (Vilage Data Denter, Box 6703, Joshua Tree, CA) has released a new software business applications catalog featuring new applications for the educator, professional, and small business entrepreneur. Applications include simple inventory elements, medical billing systems, and complete school administrative packages. Free.

☐ *A Rainbow of Applications,* fourth edition catalog of **Rainbow Computing** (19517 Business Center Dr., Northridge, CA) has just been released. Informative descriptions of approximately three hundred fifty products for the Apple; business and home applications, education and science, entertainment, languages, personal, and utilities; plus a wide assortment of peripherals. $2 shipping and handling; $5.50 for foreign orders.

☐ An analytical tool for the decision-making process, *Graph-Power* from **Ferox** (1701 N. Ft. Meyer Dr., Suite 611, Arlington, VA) creates hard-copy output in many colors on paper or transparencies ready for use in reports, presentations, and camera-ready reproduction. Type style, type size, positioning, color, pattern, and overlay features; data manager allows user to change, rearrange, summarize, and save data and graphs. $299.

DATA CAPTURE 4.0©

The most advanced and easiest to use telecommunications program for use with the MICROMODEM II™ or the Apple COMMUNICATIONS CARD™.

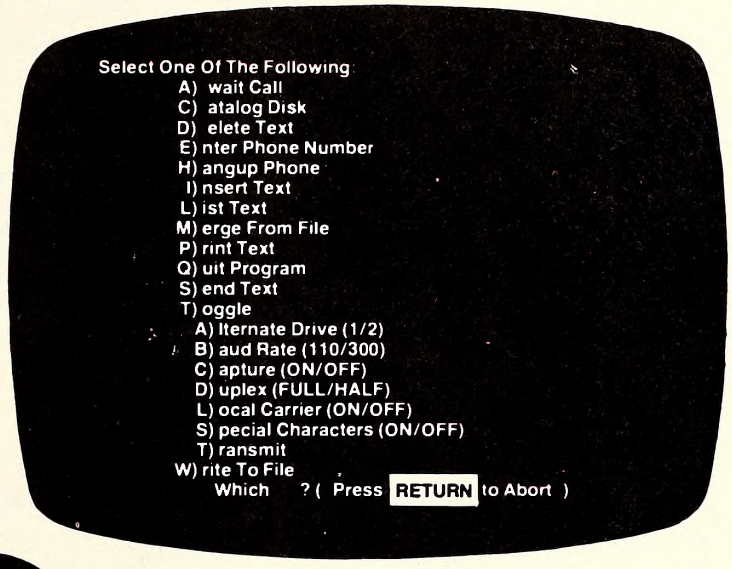
If you use your Apple II™ computer to communicate with other Apples or with timesharing systems, then you need Data Capture 4.0.

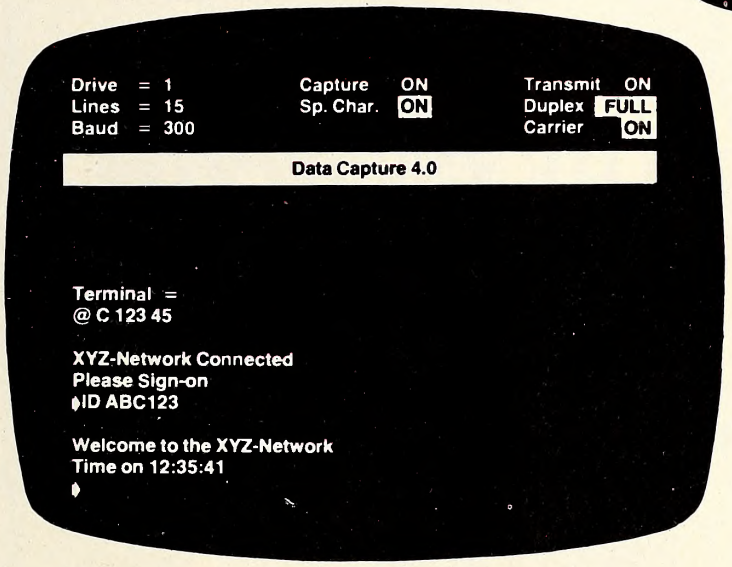
If you want to save stock quotes, airline schedules, electronic mail, or other data from a remote computer system for future use, then you need a smart terminal program for your Apple II™. Data Capture 4.0 is the most user friendly and most flexible program of this type. It is specifically designed to take advantage of all the features of the Micromodem II™, but it works equally well with the Apple Communications Card™ and several other popular serial interface cards.

Data Capture 4.0 is the only Apple II™ smart terminal program available which is fully copyable and modifiable. This means that you don't have to worry about backup. Go ahead and make all the backup copies you need.

The unprotected format and helpful documentation make modifications to Data Capture 4.0 very easy to perform. In addition, full technical support is available by phone from Southeastern Software.

Data Capture 4.0 has many other features. Incoming data files are automatically captured regardless of length. Data in the memory buffer can be viewed, edited, printed, saved to or loaded from disk, or transmitted to the remote system at any time. An unattended mode of operation is provided so that you can call your Apple from another location and send data to it or load data from it. Data Capture 4.0 is fully compatible with the Apple III™ in emulation mode. It is also compatible with all popular lower case adapters for the Apple II™, including the widely used shift key modification. An automatic logon utility for use with the SOURCE is provided with Data Capture 4.0. Also included is a HELP text file containing the latest tips and suggestions on using Data Capture 4.0 effectively (many of these are the result of feedback from customers).





See your local Apple dealer today for a demonstration of Data Capture 4.0 or order direct from Southeastern Software at the address below.

(Please include $2.50 for postage and handling.)

If you presently own a previous version of Data Capture you may upgrade to the new version for the difference in price plus $2.50 for postage and handling. MASTER CARD or VISA orders may be placed by phone.

DATA CAPTURE 4.0 ..... $65.00
DATA CAPTURE 4.0 / 80* $90.00

*Specify either Videx, Smarterm, Double Vision or Sup 'R' Terminal version

— Dealer Inquiries Invited —

Requires DISK II™, Applesoft II™ and 48K of Memory

*Apple®, Apple II Plus®, Disk II® and APPLESOFT II® are trademarks of Apple Computer Company.
SOURCE - Source Telecomputing Corporation.
*Micromodem® is a trademark of D.C. Hayes Associates, Inc.

Southeastern Software

6414 Derbyshire Drive ● New Orleans, Louisiana 70126
Dept. ST (504) 246-8438 or (504) 246-7937

master charge THE INTERBANK CARD
BankAmericard VISA

☐ Incorporating the features of larger, dedicated word processing systems, *The Gutenberg Word Processor* includes split-screen editing, ability to define character sets, full word wrap around, automatic search and replace with counter; programmable keys for advanced keystroke savings to be used in data capturing applications. Has user selectable forward and reverse scrolling by screen lines, and a unique set of arithmetic and conditional commands. *Paint* program allows the user to create any heading, logo, illustration, or graph and place it anywhere on the page. From **Micromation** (1 Yorkdale Rd., Suite 406, Toronto, Ontario, Can.). $315.

☐ *Dentistaid,* the dental office management program from **Hayden Book Company** (50 Essex St., Rochelle Park, NJ) is designed to streamline all major financial tasks performed in the dental office for complete practice control and increased efficiency and profits. Automatically prints standard ADA insurance forms, prequalification and actual services, monthly statements, patient recall notices, accounts receivable aging reports, daily summary of work performed and payments received, daily, weekly, monthly, and yearly totals, and production analysis using standard ADA categories, and display of individual accounts. Two day installation. Requires three drives, 132 column printer, and Z-80 card. $1,000. ☐ *The Hayden Applesoft Compiler* is now available in 3.3, featuring the improvements of automatic garbage collection and ability to print out compiler statistics, and a free backup with returned registration card—all improvements included in 3.2 version, now with desensitized protection block that plugs into the game paddle socket. Either version: $175. ☐ **Hayden's** word processor, *Pie Writer,* will be available soon with incremental spacing, ability to output formatted files to disk, and implementation of escape sequences so they don't interfere with justification. Special version of the format printer will permit proportional spacing on Centronics 739 and Epson printers. $129.95.

☐ Two new games from **Broderbund** (2 Vista Wood Way, San Rafael, CA): *Red Alert* by Olaf Lubeck features a variety of weaponry to help you protect a suffering humanity from Space Meanies and Thudputters. Either DOS. $29.95. ☐ *Midnight Magic* by David Snider has dual flipper controls, upper and lower playing levels, tilt mechanisms, rollovers, multiple ball play, electromagnetic deflectors, and special effects for hi-res pinball action. Either DOS. $34.95.

☐ **Sams Books** (4300 W. 62nd St., Box 7092, Indianapolis, IN) announces publication of *Applesoft Language* by Brian and George Blackwood. Lessons introduce detailed programming routines, progressing to print rules, variables, and loops to functions, reserved words, and program outline; followed by logic, formulas, double subscript arrays, and thought enabling interaction among the programmer, program, and computer. Also provides skills needed to read and design flowcharts, plus graphics section and reference points for all lessons. 254 pages softbound. $10.95. ☐ Also from the Blackwoods is *Intimate Instructions in Integer Basic,* covering sorting, flowcharting, graphics, loops, functions, and variables. Each chapter provides definitions, fundamentals of a programming technique, and one or more self-testing lesson exercises, with an appendix providing a deeper explanation of sort programs and more information on flowcharting and graphics displays. 158 pages, softbound. $7.95.

☐ **Beagle Bros.** (4315 Sierra Vista, San Diego, CA) has two new utilities. *Utility City* features printer utilities program changers, copyright protectors, an Applesoft lister, text screen formatter, command zap, filename zap, more. Twenty-one in

GOTO 93

starring JULIUS CAESAR - CHRISTOPHER COLUMBUS - CAVEMEN - ROBIN HOOD - BENJAMIN FRANKLIN - CLEOPATRA and YOU
With a Cast of Thousands
executive producer KEN WILLIAMS • produced by BOB DAVIS
☆ ☆ ☆ written and directed by ROBERTA WILLIAMS ☆ ☆ ☆
project development: TERRY PIERCE, ERIC GRISWOLD, RORKE WEIGANDT, JEFF STEPHENSON
Admission $99.95

UA	ULTIMATE ADVENTURE
	48K APPLE II or II PLUS

ON-LINE systems
36575 Mudge Ranch Road • Coarsegold, CA 93614 • 209-683-6858
Add $1.00 for shipping VISA, Mastercard, C.O.D. or checks accepted

They have landed and are taking over the city. Steadily they are making their way across the city, destroying everything in their paths. The town has been evacuated and your regiment has retreated leaving you, alone in the city, at the mercy of the aliens.

The aliens have you surrounded, and laser shots fly from all directions. Your movements are confined but you haven't given up. If you're going to live, you'll have to concentrate on where the shots are coming from and where you're going because if you don't, you'll get caught in the **CROSS FIRE**.

CROSS FIRE is a unique new game by JAY SULLIVAN featuring HI-RES graphics and sound, smooth quick animation, and some of the best arcade challenge available anywhere. **CROSS FIRE** runs on any 48K APPLE II/II PLUS DOS 3.2 or 3.3 and is available now for $29.95 on disk from your local computer store or you may order directly from.....

ON-LINE systems

36575 Mudge Ranch Road - Coarsegold, CA 93614 - 209-683-6858

ADD $1.00 FOR DIRECT ORDERS

VISA, MASTER CHARGE, C.O.D. or CHECKS ACCEPTED

ON-LINE SYSTEMS introduces arcade gaming as an art form. THRESHOLD, by WARREN SCHWADER and KEN WILLIAMS, features fast smooth animation, HI-RES graphics, and more challenge than you'll find in any other arcade game on the market.

THRESHOLD is an arcade game with alien attackers galore. In fact, there are more monsters out there than we expect you'll ever see. THRESHOLD was designed to be an arcade game that you won't get bored of, and that means a constantly changing game with a graduated skill level, but more than that, THRESHOLD means constant fun.

THRESHOLD runs on any 48K Apple II or II Plus DOS 3.2 or 3.3. Available now for $39.95 on disk from your local computer store or you may order directly from. . . . .

ON-LINE systems

36575 Mudge Ranch Road · Coarsegold, CA 93614 · 209-683-6858

ADD $1.00 FOR DIRECT ORDERS

Visa, Master Charge, C.O.D. or Checks Accepted

Two New Additions to our Best Selling Adventure Series

ULYSSES and the Golden Fleece

HI-RES ADVENTURE #4
Our First 2 Disk Adventure!

BOB DAVIS, Co-ordinator for TIME ZONE, the "Ultimate" HI-RES Adventure project, has teamed up with KEN WILLIAMS to bring the high adventure of ancient mythology to your APPLE. "CAST-OFF" with ON-LINE SYSTEMS and ULYSSES, and see if you can find the Golden Fleece.

ULYSSES is a HI-RES Adventure epic set in ancient Greece.

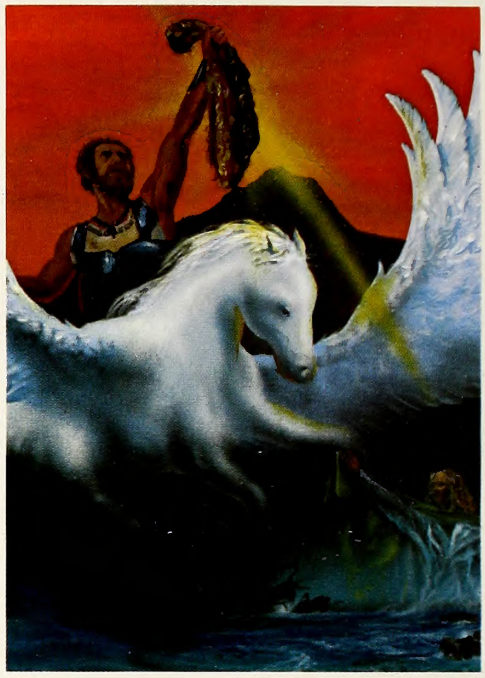
PRICE $34.95

CRANSTON MANOR

HI-RES ADVENTURE #3

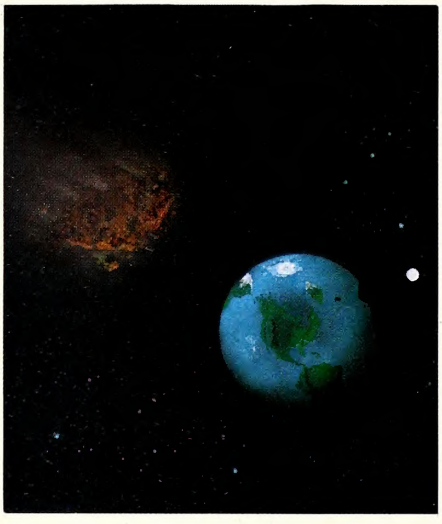
In this HI-RES Adventure, you explore the estate of Old man Cranston and find the riches that are hidden throughout the manor and surrounding grounds. Be careful! Rumor has it that the disembodied spirit of the old man roams the estate guarding his fortune.

CRANSTON MANOR is a HI-RES Adventure game set in contemporary times in Coarsegold California.
PRICE $34.95



Hi-Res Adventure # 3
CRANSTON MANOR
by Ken Williams & Harold DeWitz

Hi-Res Adventure #4
ULYSSES
by Bob Davis & Ken Williams



Hi-Res Adventure #0
MISSION ASTEROID
by Ken & Roberta Williams

Written as an introduction to the HI-RES Adventure family of games, MISSION: ASTEROID is designed to acquaint beginning adventure players to the wonderful world of HI-RES Adventures.
PRICE: $19.95

Hi-Res Adventure #1
MYSTERY HOUSE
by Ken & Roberta Williams

When you enter the house, you are pulled into the mystery and intrigue as your companions are murdered one by one. Be careful, you may be next! Can you solve the mystery and leave the house alive? The secret passage way may lead you to the answer.
PRICE: $24.95

Hi-Res Adventure #2
WIZARD AND THE PRINCESS
by Ken & Roberta Williams

After one year on the best sellers list, this adventure is still going strong. To win this game you will have to cross deserts, chasms and oceans in your search for the kings daughter. If you find her, half of the kingdom of Serenia is yours.
PRICE: $32.95

ON-LINE systems

Visa, Master Charge, COD or Checks Accepted
For direct mailing add $1 for shipping.

36575 Mudge Ranch Road • Coarsegold, CA 93614 • 209-683-6858

mouskattack

by JOHN HARRIS
and OLAF LUBECK

Join in the fun as ace plumber Larry Bain of Rodent Rooter plumbing service, tries his best to pipe the sewers of "RAT ALLEY", which has the biggest rat population in the world.

An unbeatable blend of arcade action and quick thinking strategy IN 100% MACHINE LANGUAGE for one or two players.

Available on disk for $34.95. MOUSKATTACK runs on any 48K APPLE II or II+ DOS 3.2 or 3.3.

Get MOUSKATTACK at your local computer store or order directly from:

ON-LINE systems

36575 Mudge Ranch Road • Coarsegold, CA 93614 • 209-683-6858

Visa, Mastercharge, Checks, COD

Add one dollar for shipping.

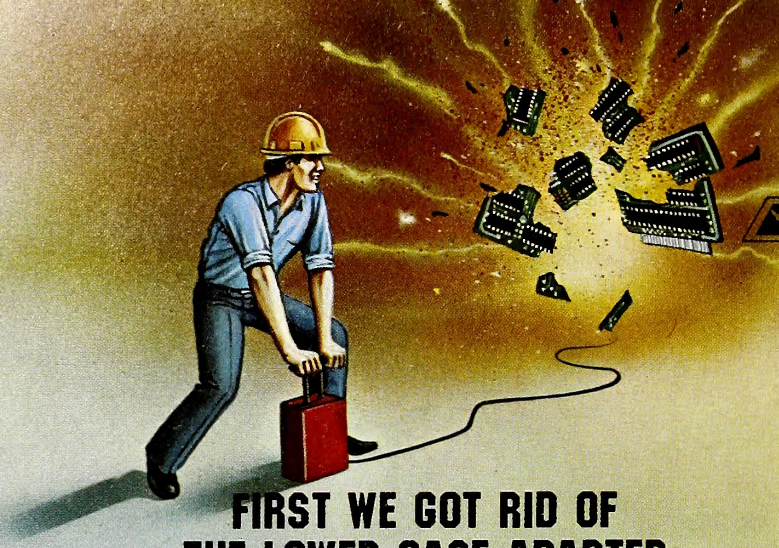
SOFTPORN

A funny, provocative, challenging adventure game **for adults only!**

by ON-LINE systems

36575 Mudge Ranch Road • Coarsegold, CA 93614
209-683-6858

$29.95
48K Apple II+ or Applesoft in ROM
DOS 3.2 or 3.3
Visa, MasterCharge, C.O.D. or Checks Accepted



SuperScribe II

FIRST WE GOT RID OF THE LOWER CASE ADAPTER

and put lower case on the screen
WITHOUT HARDWARE.

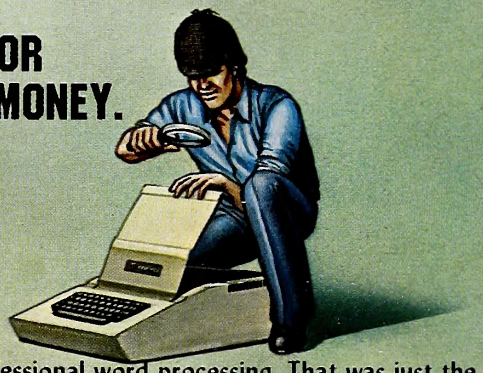
THEN WE GOT RID OF THE 80 COLUMN BOARD

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 ON-LINE SYSTEMS began work on SUPERSCRIBE 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 SUPERSCRIBE II hardware independent, so we eliminated the lower case adapter and the 80 column board. SUPERSCRIBE II is the only word processor on the market to offer true software based lower case and a 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 SUPERSCRIBE 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 On-Line Systems, We Won't Stand For It.

SUPERSCRIBE 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 . . . . .

VISA, MASTERCHARGE, CHECK, C.O.D.

ON-LINE systems

36575 MUDGE RANCH ROAD · COARSEGOLD, CA 93614 · 209-683-6858

WE'RE BUILDING UP SPEED

The race has just begun

A recent article in "INTERFACE AGE" tested 45 of the most used BASICs on 48 different microcomputers. That article judged that if you are an APPLE owner writing programs in APPLESOFT, you have the nineteenth fastest BASIC/computer combination tested.*

–THAT WAS BEFORE EXPEDITER II.–

If you're an APPLE owner *with EXPEDITER II* writing programs in APPLESOFT, you can now have your BASIC program judged second in the same comparison - with a minimum of effort. You need only LOAD your program, BRUN EXPEDITER, then save your program.

EXPEDITER II is a new programming utility which translates your APPLESOFT programs into machine language, usually on the first try. Problem spots in the code are flagged during compilation. The new compiled code supports all the features of APPLESOFT including HI-RES graphics, shape tables, LO-RES graphics, defined functions and DOS commands. EXPEDITER II is the only compiler to support global string variables easily. There is no additional BASIC syntax to learn. The only real difference you'll notice with the new program is that it's faster - *usually two to twenty times faster.*

Other APPLESOFT compilers are available for the APPLE, but none have all the features of EXPEDITER II. EXPEDITER II, with its new MEMORY COMPRESSION OPTION can reduce your compiled program size by up to 50 percent. (Remember: a machine language program is normally bigger than a BASIC program that does the same thing). It's also compatible with MMS II** to help you solve any memory problems you might encounter.

We make your compilation process easier by providing a symbol table and address listings for each line of your BASIC program. EXPEDITER II even allows you to reserve areas for machine language subprograms or graphics. Debugging is facilitated by EXPEDITER IIs support of the Applesoft TRACE command.

We at ON-LINE SYSTEMS believe that the APPLE II is a powerful machine, and we're constantly trying to make it better and easier to use. Thats why we offer a full line of utilities written by some of the best programmers in the field. The only limits we feel the APPLE should have, are the limits to your imagination.

*Write for more information on the INTERFACE AGE article.
**MMS II is a utility program by ON-LINE SYSTEMS that relocates the Disk Operating System on your memory expansion board. Available for $49.95.

EXPEDITER II

The Optimizing Applesoft* Compiler

By Shmuel Einstein and Dennis Goodrow EXPEDITER II is a trademark of Einstein/Goodrow

EXPEDITER II runs on any APPLE II/II+ with 48K APPLESOFT in ROM and is available for 99.95 at your local computer store or order directly from . . .

ON-LINE systems

36575 Mudge Ranch Road • Coarsegold, CA 93614
209-683-6858

Add one dollar for shipping

Visa, Mastercharge, Check, COD

Do You Feel That Assembler

is an

UPHILL BATTLE?

Let Us Help!

LISA 2.5

A 6502 Assembly Language Development System for Apple Computers. Used in homes and businesses by novices and professional programmers for over two and a half years, this most popular assembler includes these special features:

- Easy to learn commands.
- Speedy Assembly: over 20,000 lines/minute.
- Uses over 30 pseudo opcodes turning your Apple into a 16 bit machine, requiring less code to perform the desired task.
- More extended mnemonics (a greater memory aid).
- More commands which allow the flexible use of DOS.
- Runs on any 48K Apple II/II+ with or without a 16K memory expansion board.
- Only $79.95 at your local computer store.

Written by Randy Hyde, author of Assembler Programming For Your Apple, LISA 2.5 has been endorsed by Steve Wozniak, Jay Sullivan, Ken Williams, Olaf Lubeck and Mark Pelczarski.

SPEED/ASM

A set of subroutines used to simulate BASIC in developing an assembly language program. This powerful, new assembly language programming tool gives the BASIC programmer power to build complex assembly language programs and includes these special features:

- 39 pages of well written, informative documentation explain how to use SPEED/ASM and your knowledge of BASIC to write professional quality programs quickly and easily.
- SPEED/ASM is an independent Assembly language subroutine package making programs produced on your APPLE compatible with ATARI SPEED/ASM and vice versa.
- No need to own a "LISA" assembler in order to operate SPEED/ASM.
- SPEED/ASM is assembly language easily understood by the BASIC programmer.
- Runs on any 48K Apple II/II+.
- Only $39.95 at your local computer store.

LISA EDUCATIONAL SYSTEM

A complete system which offers LISA 2.5, SPEED/ASM, and Using 6502 Assembly Language, Randy Hyde's new 300 page book on assembler for beginners, all for the price of $119.95 giving you a savings of $20.00.

ON-LINE systems

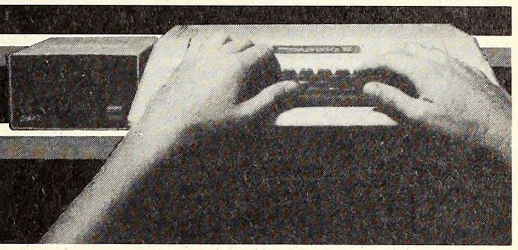
Available at your local computer store from:

36575 Mudge Ranch Road Coarsegold, California 93614 209/683-6858

To order direct, send purchase price plus $1.00 for shipping
(Visa, Mastercard, C.O.D. or checks accepted.)

*Apple is a registered Trademark of Apple Computer, Inc.
*Atari is a registered Trademark of Atari, Inc.

MARKETALK News



from page 84

all, plus peeks and pokes chart and tip book. $29.50. ☐ *Alpha Plot* creates color drawings and charts, types in upper and lower case with proportional spacing; three cursor modes, image combiner and changers. Scrunch routine saves in one-third normal memory space. With peeks and pokes chart and tip book. $39.50.

☐ When is Computer Works really **ComputerWorks**? When the company with two words is in Virginia. The single-word company is located in San Rafael, California, and is the one responsible for Rainbow's Pro Paddle product, not the Virginia company of similar name as was reported in the August *Softalk*.

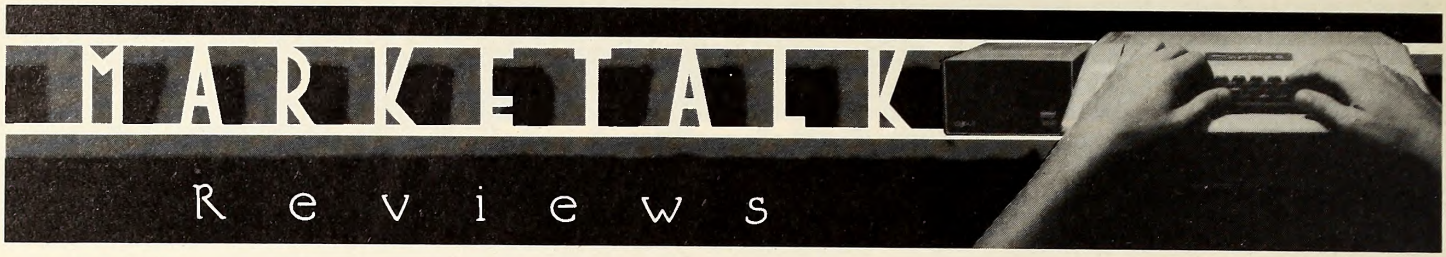
☐ *Ceemac* is a new visual composition language from **Vagabondo Enterprises** (1300 E. Algonquin 36, Schaumberg, IL). Fashioning abstract visuals in the manner of the musical notation system, the language requires a feeling for graphics concepts and programming ability in Basic. The Apple shape system has been abandoned for one with more capacity and range. Rapid interaction between editor and interpreter; compose-execute-compose swapping by single key commands. The interpreter has been released as *Fire Organ,* available free from dealers. $40.

☐ *Zardax,* a word processor from **Computer Solutions** (Box 397, Mount Gravatt, 4122, Australia), allows editing of any text up to 13,500 characters, links together shorter documents to form long reports, and produces personalized letters from a file of names, addresses, and additional information. Documents can be retrieved from disk, altered, recorded, and printed. $295.

☐ **Apple Computer** (10260 Bandley Dr., Cupertino, CA) has produced a word processor for the Apple III. *Apple Writer III* uses WPL (Word Processing Language) to automate the process of text manipulation and document creation. Allows any changes, insertions, deletions, anywhere; adjusts printing format to specifications during printing; creates reports, rearranges documents, translates from typewriter shorthand to English and other languages and back again. 128K. $225.

☐ *Pascal III* has been introduced by **Apple** as a high-level operating system using the Apple III's SOS in the development of Pascal and Apple III Pascal assembly language subroutines. Includes editor, filer, Pascal compiler, Apple III Pascal assembler, linker, and librarian. 128K. $250.

☐ Providing the functions of a standard Apple 16K memory card plus sixty additional 4K banks of memory, the *App-L-Cache* from **Sorrento Valley Associates** (San Diego, CA) is a 256K memory card that functions as a large solid-state disk without moving parts. Will operate with systems like DOS, Pascal, or CP/M. Eliminates the need for a second mini disk drive, reduces disk swapping, compilation, and assembly time; gives high-speed file sorting capability. $1,595. ◨



Unless otherwise noted, all products can be assumed to run on the Apple II, Apple II Plus, and Apple III in the emulator mode and to require 48K and one disk drive. The requirement for ROM Applesoft can be met by RAM Applesoft in a language card.

Crossfire. By Jay Sullivan. Here is a game, totally new and just for the Apple, that will make the arcaders green with envy. *Crossfire* is the best home-arcade game to cross our screens since *Alien Rain*—well, since *Raster Blaster*—well, since *Bezman*. . . . The point is, *Crossfire* is great.

Chances are, you'll feel lucky to score a simple hundred the first few times you take on the invading aliens. *Crossfire* consists of a grid such as you might see on the street map of a city, which is precisely what this represents. All your allies have turned tail; you alone remain to defend the city from aliens, of which there are many. The aliens come at you from three directions—later four—relentlessly. They fire in the direction they move—if you're nearby.

These are not the aliens of *Super Invader* or even of *Alien Rain*. These aliens are intelligent. They know just where you are and they go after you. They even seem to anticipate some of your moves. While this makes them a formidable foe, it also renders them subject to being outwitted. Try it.

You can move in any direction through the city streets, including back where you just came from. You can shoot in any direction as well, whether you are headed that way or not. You must shoot and move swiftly to evade the crossfire of the aliens. They often manage to get off a shot just as you get them; they explode, but their shot continues. Each alien has four stages of metamorphosis; when you do in one, the next appears at the point of origin for that particular creature.

Four lights surround the center of the map; they become consecutively available to you to run over, which renders the most points. But you must plan your strategy to reach them. In addition, your ammunition is limited. New sources of ammunition appear when you are ten bullets from empty; where they appear depends on where you aren't. You must maneuver among the aliens' fire to reload.

Although a home-arcade game without doubt, *Crossfire* offers much opportunity for strategy—far more than normal in a shoot-'em-up. Because of this, and because of the intense concentration required to watch and respond to the entire screen at once, *Crossfire* looks to be a winner for all gamers.

Programmed in colorful hi-res graphics, *Crossfire* shows advanced smoothness in motion as well as an excellent, fast capacity for monitoring many individual pieces apparently simultaneously. With *Crossfire* and the recent *Threshold*, On-Line moves firmly into the same league as Sirius, Broderbund, Cavalier, and the like as a purveyor of outstanding home-arcade fun. *Crossfire* is a must for the holiday fun of every game-playing household. MCT

Crossfire, by Jay Sullivan, On-Line Systems (36575 Mudge Ranch Road, Coarsegold, CA). $29.95.

Rubik's Cube Unlocked. By Jeff Gold. Although you can use this program as a substitute for the famous cube, that's not its main purpose. *Rubik's Cube Unlocked* is designed for those who have the real McCoy and have begun, out of total frustration, to feel like Hatfields.

Unlocked prompts you to enter the colors on the monitor cube precisely as they appear on your cantankerous one. Then the computer directs you, step-by-step, to its solution. As you go, the moves are displayed in hi-res color (or, optionally, text) on a three-dimensional representation and on six separate representations of the six sides. To be still more graphic, the side to be changed appears again separately above another look at it after the proposed change.

In another option, the program will generate a random cube—actually by messing up the cube you've given it in your choice of a given number of moves. In still other options, the program arranges the cube in several patterns other than the solid sided standard win.

The most outstanding feature of *Rubik's Cube Unlocked* is its speed. There is no delay, even at the beginning, before it solves the problem. The hi-res, too, is fine, with a nice graphics character set. The colors, unfortunately, could be better; a disclaimer apologizes for some running into others by claiming impossibility. Some other alternative needs to be found then. The bleeding colors are still clear enough to read, so this is not a reason to forego this program.

If you've been tearing your hair out over the cube, *Rubik's Cube Unlocked* could prevent baldness. Don't overlook it. MCT

Rubik's Cube Unlocked, by Jeff Gold, Double-Gold Software (13126 Anza Dr., Saratoga, CA). Either DOS. $24.95.

Geometry and Measurement Drill and Practice. By Charles Lund. Program author Charles Lund believes, as does the National Council of Supervisors of Mathematics, that geometry and measurement are among the basic skills of mathematics. The package he has created, *Geometry and Measurement Drill and Practice,* offers students who are learning about geometry and measurement an enjoyable opportunity to develop and reinforce their skills inside or outside the classroom.

MONEY BUSINESS

Beat the I.R.S.

You'd always pay the minimum Federal income tax if you had the knowledge and time to examine 120 tax return options. Well, now you can do it, with Datamost's TAX BEATER.

Written by an I.R.S. Enrolled Agent, with 24 years of financial and tax planning experience, the TAX BEATER automatically evaluates up to 120 return options from your input. It searches and finds the best tax path for you . . . displaying up to 15, and ready to print out the optimum method so you pay the minimum!

The TAX BEATER can handle it all. From income averaging to loss carryovers to dividends, alimony, pension to special exclusions. It's perfect for the average taxpayer . . indispensable for the financial professional. It's so complete, so logical that, especially in this interim year of unusual tax revisions, it can easily pay for itself many times over. Be ready for the I.R.S. . . . with TAX BEATER, the program that's updated for the 1981 tax laws!

$129.95 on disk for Apple II* Complete with thorough, easy to follow documentation.

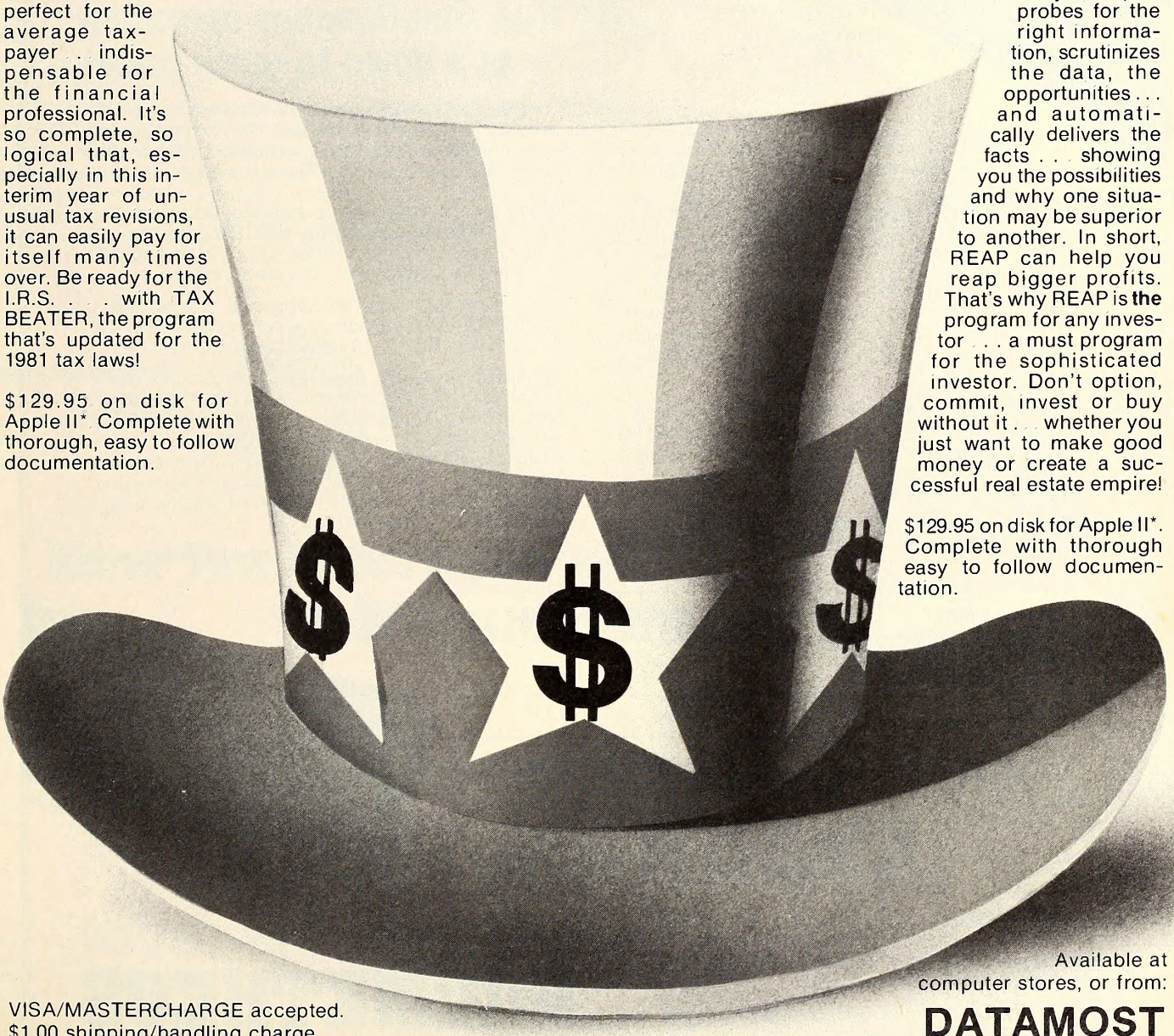
Beat the Real Estate game

Real Estate has always been the major wealth-builder. In fact, more millionaire dynasties owe their creation to investments in real estate than anything else. And, it doesn't matter . . . boom times or bust . . . inflation or deflation . . . the knowing investor can make money, even create an empire!

But, the key to real estate success is accurate analysis of opportunities, knowledge of cash flows, return on investment, current and future profitability, tax consequences and other interrelated information — all needed, all necessary to making the right decision.

And that's what REAP (Real Estate Analysis Program) is all about. It takes your input, probes for the right information, scrutinizes the data, the opportunities . . . and automatically delivers the facts . . . showing you the possibilities and why one situation may be superior to another. In short, REAP can help you reap bigger profits. That's why REAP is **the** program for any investor . . . a must program for the sophisticated investor. Don't option, commit, invest or buy without it . . . whether you just want to make good money or create a successful real estate empire!

$129.95 on disk for Apple II*. Complete with thorough easy to follow documentation.



VISA/MASTERCHARGE accepted.
$1.00 shipping/handling charge.
(California residents add 6% tax)

Available at computer stores, or from:

DATAMOST
19273 Kenya St.
Northridge, Ca. 91326
(213) 366-7160

* Apple II is a trademark of Apple Computer, Inc.

NEW...FOR APPLE II[1]

FROM PROMETHEUS

VERSAcard

1. Serial Input/Output Interface
2. Parallel Output Interface
3. Precision Clock/Calendar
4. BSR Control

NEW 4 in 1 CARD

ALL ON ONE CARD
with true simultaneous operation
. . .at a down-to-earth price!

CHECK THESE FEATURES: True simultaneous operation! VERSAcard's unique hardware selection circuitry make your Apple "think" that separate cards are installed. This allows VERSAcard interface ports to be compatible with existing software such as APPLE PASCAL, Microsoft Softcard[2], and most other Apple software.

The Serial Input/Output port is RS-232C standard with crystal controlled baud rates from 19.2K to 110 baud. You can connect with a CRT terminal, line printer, or virtually any RS-232 modem. VERSAcard's "Terminal" mode, along with a modem, connects you with the *Source* or a variety of other computers. "Terminal" mode also allows you to output to both printer and display simultaneously.

The versatile **Parallel Output Interface** is configured for the Centronics standard and is easily configured for other standards. Simply hookup your **Epson, Okidata, IDS,** or other printer and begin printing.

Precision Clock/Calendar. With all standard real-time clock/calendar functions and one second resolution. Battery back-up supplied. And, best of all, it's **software compatible with Thunderclock.** This makes numerous other software packages immediately usable. Interrupts are supported.

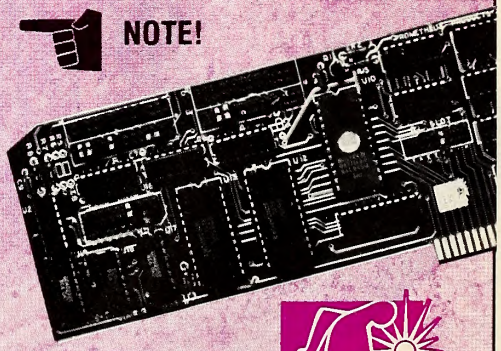
BSR Control. Add an ultrasonic transducer, and VERSAcard provides you with remote control through your BSR X-10 Home Control System.

Onboard firmware provides optional auto line feed, video, paging, terminal mode, lower to upper case, 8th bit set or clear . . . and more. These versatile device drivers permit connection to virtually any standard peripheral. The Prometheus engineering staff is available to help. With any peripheral and special device interface problems. Call us, we'd like to hear from you.

Order: PP-VCD-1 VERSAcard @ $249.00 each.

Compare Key Features	VERSAcard	CPS MultiFunction Card
Parallel, Serial and Realtime Clock Interfaces?	YES	YES
BSR Control Interface?	YES	NO
Compatible w/ Apple Pascal?	YES	NO. Special Pascal disk required.
Compatible with CP/M?	YES	NO. Special CP/M disk required.
Cable Included?	YES. One cable of choice free with each order —through 1981.	NO. Must be purchased separately.
Apple Modification?	NO	YES

NOTE!



Dealer inquiries invited.

VERSAcard—the only Apple peripheral card you're likely ever to need!

All Prometheus boards are completely tested and burned-in prior to shipment. A one-year warranty covers parts and labor.

Registered Trade Marks: (1) Apple Computers, Inc. (2) Microsoft Consumer Products

PROMETHEUS

PROMETHEUS PRODUCTS INCORPORATED

42577 Fremont Blvd. • Fremont, CA 94538 • (415) 490-2370

Although it's designed primarily as a learning supplement for students working at the second to tenth grade levels who have some prior knowledge of geometric and measurement concepts, this package can be used to advantage by nearly anyone who would like to become more comfortable and proficient in matters of measurement and geometry.

Included in the package are two master disks, two backups, and a documentation booklet. Each master disk contains four master programs that are accessible from a main menu. The first set of programs drills the user on classifying polygons, measuring the length of line segments (or caterpillars) in metric terms, measuring perimeter and area (easy version), and telling time; the second set covers circles, angle measurement, perimeter/area measurement (advanced version), and volume/area measurement.

Each master program also has subprograms within it, resulting in a total of thirty categories for students to choose from. The various subprograms are arranged in order from simplest to most advanced, and users can choose how many exercises of a particular type they would like to do. Minimum and maximum numbers of exercises (ordinarily one to ten) are usually indicated, although in the case of the drill on angle measurement a maximum number wasn't indicated, and we discovered we could ask for a series of up to ninety-nine random problems.

In all exercises, except the clock drill for which thirty seconds is the time limit, a user can take as long as desired to answer a question. Three tries are allowed before the correct answer is revealed, and the scoring feedback that's given when a series of problems has been completed tells the student how many questions were answered correctly on the first, second, and third tries, and how many questions (if any) were not answered correctly. The learner can then decide to run the subprogram again for additional practice, return to the main menu to choose another variety of problem, or quit.

The documentation booklet is intended as a guide for teachers and parents but will also interest older students who are curious about what's in store as they move through the various programs. Included in the manual are summary overviews of the eight master programs, illustrations of sample screens for each of the master programs, and a sample program run-through. Program summaries also indicate the grade level range for which a particular program is geared.

One of the appendices to the documentation contains short definitions of all the important terms used in the various programs. These will help parents, teachers, and students refresh their memories about the meanings of various terms (it's not easy to remember off the top of your head precisely what an isosceles triangle is or how many sides a heptagon has). These miniature glossaries also provide raw material for matching or fill-in quizzes on definitions.

The program itself is a delight to use and earns an A for user-friendliness. Its organization is straightforward and logical, and its screen prompts are clear and supportive of the learner's efforts. Learners will enjoy being able to work independently at their own pace. This well conceived, well executed program is an example of CAI at its best. JEV

Geometry and Measurement Drill and Practice, by Charles Lund. Special Delivery Software/Apple Computer, 10260 Bandley Drive, Cupertino, CA. 32K. $50.00.

Beer Run. By Mark Turmell. When you first play *Beer Run* your patience will be taxed. Starting at the bottom of the Sirius building, you are faced with the arduous task of climbing ladders and riding elevators to the top of the building, where you will be picked up by a blimp and transported to the Olympia Beer building. On the way you can become lightheaded

catching some of the constant stream of beer cans falling from above or acquiring kegs of beer that pop up frequently.

Climbing ladders and riding elevators is difficult as it is, particularly after you've had a few beers, but there are also guzzlers and bouncers, out to send you to the bottom of the building. At first it seems impossible to maneuver past the drunken guzzlers and busybody bouncers. There is no way to make them go away, and they seek you out with the intention of ruining your day. When you climb a ladder you can only go down after you reach the top of that ladder. And when you go down you can only go down one level.

This makes it all very frustrating at first. Frequently you'll find yourself stuck on a ledge with nowhere to go and the guzzlers marching stupidly towards you. Needless to say, all this frustration makes you play harder and develop strategies for using the elevators, which are the only things that get you anywhere in this game. Patience is the key to success here, as the elevators go up and down.

The Sirius building is thirty floors high, and if you make it to the top an erratically flying blimp lowers a rope. Catching the rope on the first try gets you extra beers, but once you've caught it the real action begins. Deposited on the top of the Olympia Beer building you find yourself in the same kind of situation as before, except now you go down. One nice thing is if you get killed in the Olympia building you don't get sent back to the Sirius building.

Mark Turmell's graphics are humorous, but one could imagine the main character looking less like a blind man walking on a treacherous ledge. The sound effects are obnoxious, but if you hit control-S it seems like something is missing.

A good party game, *Beer Run*'s real mystery is the Artesians. Tantalizing messages appear on screen during play, announcing their arrival in the building, but if you find any it probably means you drank too many beers and are imagining things. Artesians or no Artesians, a game where you try to amass a year's supply of beer is a noble concept. Even if it's imaginary beer. DH

Beer Run by Mark Turmell, Sirius Software (2011 Arden Way, Sacramento, CA). $29.95.

Volcanoes. By Gordon Goles. Adventurers who are tired of battling dragons and wizards might want to try their keyboards at volcanoes instead. Volcanic eruption is no less dangerous than the average dragon or wizard (remember Vesuvius and Krakatoa), and this new game allows the veteran adventure player to tax his powers of strategy, foresight, and reasoning. *Volcanoes* automatically throws two or more players into the role of expert volcanologists. The object of the game is to predict via a series of investigations which volcanoes in the mythical land of Wrangalia are going to erupt, so the inhabitants of the neighboring villages can be properly warned and evacuated.

The game is self-prompting; you never even have to read the manual to play. And that's almost too bad. The author is a professional volcanologist, and the documentation manages to be extremely educational and extremely interesting at the same time. The documentation "blue book" was also written to enhance the player's enjoyment of the game with facts every amateur volcanologist should know from history to a working glossary.

The investigations the player must undertake to make his predictions are those performed on true volcanoes to determine hazards and impending activity, including infrared scan to find "hot spots"; seismic survey to look for volcanic tremors; electrical conductivity to detect zones that may indicate the existence of shallow magma chambers; and tiltmeter survey to find patterns of short-term swelling or deflation. In addition, detailed background is provided in the documentation to explain the logic of the investigation—why shallow magma chambers are so significant. The methods are also clearly outlined.

With all this sophisticated research and scientific informa-

tion available, it should be easy to prevent a future Pompeii, right? Wrong. In this game each player-scientist starts out with a given amount of money (about $352,000) to spend as he or she likes on tiltmeters and gas analyses. Goles's costs closely parallel the actual costs of such investigations—gas analyses are about $60,000 cheaper than taking a seismic survey. In this game, the scientists don't have extra money to burn, so scientific inquiry must be made wisely.

Which brings up another point in this game. Goles intentionally programmed the game to teach cooperation as well as geology. The volcanoes seem to be winning when the players don't cooperate as a scientific team.

Another feature is the "electronic mailbox" to provide a little bit of chance to keep the game interesting. But Goles says this feature only comes into play if your credibility should either become very high owing to your expert predictions or too low owing to your poor predictions. There's also three levels to keep the game hot. And as many as four players can participate in making the correct predictions or destroying the town. A feature also allows players to exit the game after an eruption and save it until everything cools off.

Goles seems to have left no stone unturned with this somewhat slow-paced but volatile game. And anyone who plays it will leave the game knowing that a volcanologist is no relation to Mr. Spock. MM

Volcanoes by Gordon Goles, Earthware Computer Services, (P.O. Box 30039, Eugene, OR). $49.50.

Snack Attack. By Dan Illowsky. Newcomer Dan Illowsky hit the Apple market with two big Christmas entries, *Thief* (Marketalk Reviews, 1981) and *Snack Attack*.

Snack Attack is an eat-the-dots game done in a Persian graphic style unique to Apple gaming.

Illowsky has packed the game with features not found in other games of the genre. Best of them is the provision that each time the program is booted, it will specify different command keys, which spreads the load through the keyboard and keeps any particular set of keys from wearing out faster than others.

The game features three distinct mazes, each of which requires different tactics to clear the dots. If you're successful in clearing more than three screens, the rotation starts anew.

The characters move faster on each succeeding screen. For those who get to maze nine or higher, you'll find the speed almost uncontrollable—a true challenge to the dedicated arcade gamer.

A thoughtful last-minute addition was the ability of the player to start the game at any of five different speed levels. This is a good option because the first couple of mazes otherwise go so slowly they could lull you to sleep. But when the speed picks up, it takes an alert and skillful player to succeed. On the other hand, those folks who have never shown an aptitude for arcade games may find the slower beginning speeds just the ticket. ART

Snack Attack, by Dan Illowsky, DataMost, 19273 Kenya Street, Northridge, CA 91326. $29.95.

The Telematic Society. By James Martin. The effort to make science and technology palatable to the ordinary person has given rise to such celebrities as Isaac Asimov and Carl Sagan. Both attempt to excite the imagination and inform the intellect without overwhelming the individual with the intricacies and complexities of mathematics and scientific language. A fixture at IBM for nineteen years, James Martin is another professional who has taken on the task of educating the public about the world we live in and the world we *will* live in through the end of the century and beyond.

Written in clear, precise language, Martin's latest book, *The Telematic Society,* strives hard to set guidelines for the telecommunications revolution that Martin sees as inevitable. His vision is of a world bound by satellites, coaxial cables, and glass fibers. The technology we already have and the innovations sure to come before the end of the century will present a profound challenge to the human race. Telecommunications technology has the power to improve all of our lives, but it also

has the power to muck things up with a vengeance.

One of the better discussions of this dual nature of the telecommunications revolution occurs in the chapter "Public Response Systems." Theory has it that by the end of the decade many more features will be available for your television or home computer. Through interactive systems it will be possible to make your opinion known in a much faster and presumably more accurate fashion. Advertisements, political speeches, even entertainment programs will be judged by the electronic response received from individual users.

The advantages here are obvious: no longer will the channels of communication be from the top down or from the inside out. One of the greatest frustrations of the modern age—not having a say in the policies of the government and the shape of the society—may change. It would most certainly change, but whether this would be for the good of all or not is a question for serious debate. Martin is quick to point out that manipulation and propagandizing may reach new heights when people have more direct means of communicating their views. Careful judgment may be eschewed for emotional whimsy as people find it easier to make their opinions known but still lack a clear knowledge of the realities involved.

Martin's solution to this problem is an old one—education. Television and microcomputers are great things, but only if they at least partially reach their potential for educating the masses. It is unthinkable that mass media in this country will evolve into the kind of dead end found in iron curtain countries, where there are only one or two television stations and no entertainment programming. On the other hand, there are untold improvements that could make television a more serious and worthwhile medium in this country.

The bulk of Martin's book is a step-by-step examination of the possibilities the future holds with the coming invasion of telecommunications. It is eminently readable and worth the effort. It is entertaining and frightening—a science fiction look at the future that is prophetic and insightful. Martin tries hard to touch all the bases and does an admirable job in making the future a thing to shape and not just something that will happen, beyond the control and knowledge of the individual. DH

The Telematic Society. Prentice Hall, Englewood Cliffs, NJ. $12.95.

Hadron. By Larry Miller. The author of *Epoch* has expanded his universe to allow adversaries to travel in any direction and, in some cases, to swoop and swerve and dodge out of sight. The craft that populate this universe are more complex than those in *Epoch*; more are animated and detailed. One bit of space debris houses people, and Hadron itself, a sort of giant robot that stalks through space, speaks its name when it passes you.

The greatest improvement of *Epoch* is the means of refueling. Instead of wandering randomly in hope of coming upon a swarm of bases as in *Epoch,* you need only find a warrior ship hightailing it away from you. That ship's on its way to the enemy base to refuel. If you follow it successfully and manage to shoot the base, your supplies are replenished. Skill will bring you what you need—not arbitrary chance. It isn't an improvement that there are no friendly bases—no friendly element of any kind . . . That's a bit devastating if you think about it.

Gone also is the *2001*-type fanfare to which *Epoch* treats you upon refueling, and totally missing are the musical, mystical, wonderful time warps. Added, besides the good things, are huge block meteors and explosion debris the size and opacity of which break the mood of the game too often.

Still, *Hadron* is fun to play and beautiful to watch even though it lacks the sense of serene, vast beauty *Epoch* could haunt you with.

If *Epoch* wasn't enough for you, you may love *Hadron.* If you loved *Epoch,* once you accept the differences, you'll probably enjoy *Hadron* a lot. MCT

Hadron, by Larry Miller, Sirius Software (2011 Arden Way #2, Sacramento, CA). Compatible with but doesn't require Sirius Joyport. Either DOS. $34.95.

Impressions

□ **Space Adventure,** by Alick Dziabczenko, Sierra Software (536 East Sahara Avenue, Las Vegas, NV). Arriving just at *Softalk*'s press deadline, *Space Adventure,* subtitled *A Real-Time Space Flight Simulator,* seemed an important enough game that it ought to be reported despite there not being sufficient time for a full review.

Dziabczenko (pronounced simply Zeb-chen'-ko) has almost said it all in the title, but you must read and believe every word. This is an adventure, of the kind that calls for talking to the computer, giving commands about what you want to do; it takes place in space and is relevant to space; it requires you to pilot a space ship through the three dimensions of space in an extremely realistic fashion, and all must be done as time and energy run down—the computer game form of real time.

What the author hasn't said is that it's all done in three-dimensional animated color graphics. And that the computer you talk to isn't your own Apple but the computer onboard the space ship. He leaves out also the option you have throughout the game to fly by the skin of your pants or to use the auto pilot and auto track to do all your mathematics and implement them. And the title doesn't expound on the three memories you have to save message clues in and another three for saving location clues.

Finally, Dziabczenko, a physicist when not programming, is too modest to mention the achievements he has made in visual terms. The view from the space ship helm, moving through the stars, has great depth and real three-dimensionality, enhanced when you move into a turn—in any direction. You can even orbit a planet—a remarkable visual effect.

As an impression, there's a potential drawback to the program. No matter how you cut it, the universe is a big, big place and flying about in it is going to take time. Even at 256 light years a *second*, the speed at which you hyperwarp, it can take five real minutes to go from one sector to another. On the good side, a lot can happen within a sector; and, even flying through a quiet sector, you can keep busy by experimenting with your onboard computer's vocabulary. Incidentally, there's a vocabulary card that presumably lists all the words the computer knows—but don't count on it. If you want to say things that aren't on the card, give them a try—some of them will be understood.

This is, in the end, an adventure; and, in the beginning, it has its own equivalent of the *Wizard and the Princess* snake. But Dziabczenko is a thoughtful man and notes in the documentation how to reach him with questions.

As games go these days, the price of this game is quite low for its value. $29.95.

☐ **Menu Generator,** by Bob Crane, Crane Software (16835 Algonquin, Suite 611, Huntington Beach, CA). Those of you who may think that *Menu Generator* must be a kitchen aid should be disabused of that notion immediately. Instead, *Menu Generator* is a highly professional code generation program that specializes in the making of computer menus.

Perhaps the finest documentation ever to accompany a company's first product and a series of menu-driven prompts will lead the user through constructing menus for his disk collection.

For those of us who have so many small programs crammed onto a disk that we have to read three screens to see what's in the catalog, *Menu Generator* is a godsend. It serves the ancillary purpose of being a dynamite organizational tool for any disk collection. $39.95.

☐ **Time Zone.** By Roberta Williams and Ken Williams, On-Line Systems (36575 Mudge Ranch Road, Coarsegold, CA). Both sides of eight disks are what you need to play *Time Zone*, and they're all included in the package.

When Roberta Williams conceived of creating the ultimate adventure, she wasn't kidding. *Time Zone* covers at least ten periods in the history—past and future—of this and (in the future) other worlds, and it takes you to each of the earth's continents in all but the earliest and most futuristic eras. (Australia wasn't a whole lot different from Europe in 50 million B.C., for instance, so there's only one segment in that time period).

At the time of this writing, all of *Time Zone* was programmed, but only about two-fifths were ready for a prerelease outsider testing. So this impression is of only the execution of the general idea, the earth, and each of the time periods in some of the continents. The most general impression is—we can't wait for the rest.

If you're a fan of On-Line adventures, you have a feast in store. Although final touches were still being thought of and implemented, this prerelease peek showed Roberta Williams to be at her best. Puzzles are fun but not impossible—although there's one to equal the snake in *Wizard*—and many are based on historical information. The author went out of her way to include interesting facts and authentic custom touches about each of the continents and eras, some in words, some in the general pictures, some required for the puzzles. If you don't already know what the Australian aborigines use rhea eggs for, you'll learn (or lose!).

The idea that you'll run into famous people is accurate, although, in some cases, that's all you do—run into them. In most, however, you get to interact in some way with them. In at least one case, you have the opportunity—and the challenge—to change history by your actions; can you do it?

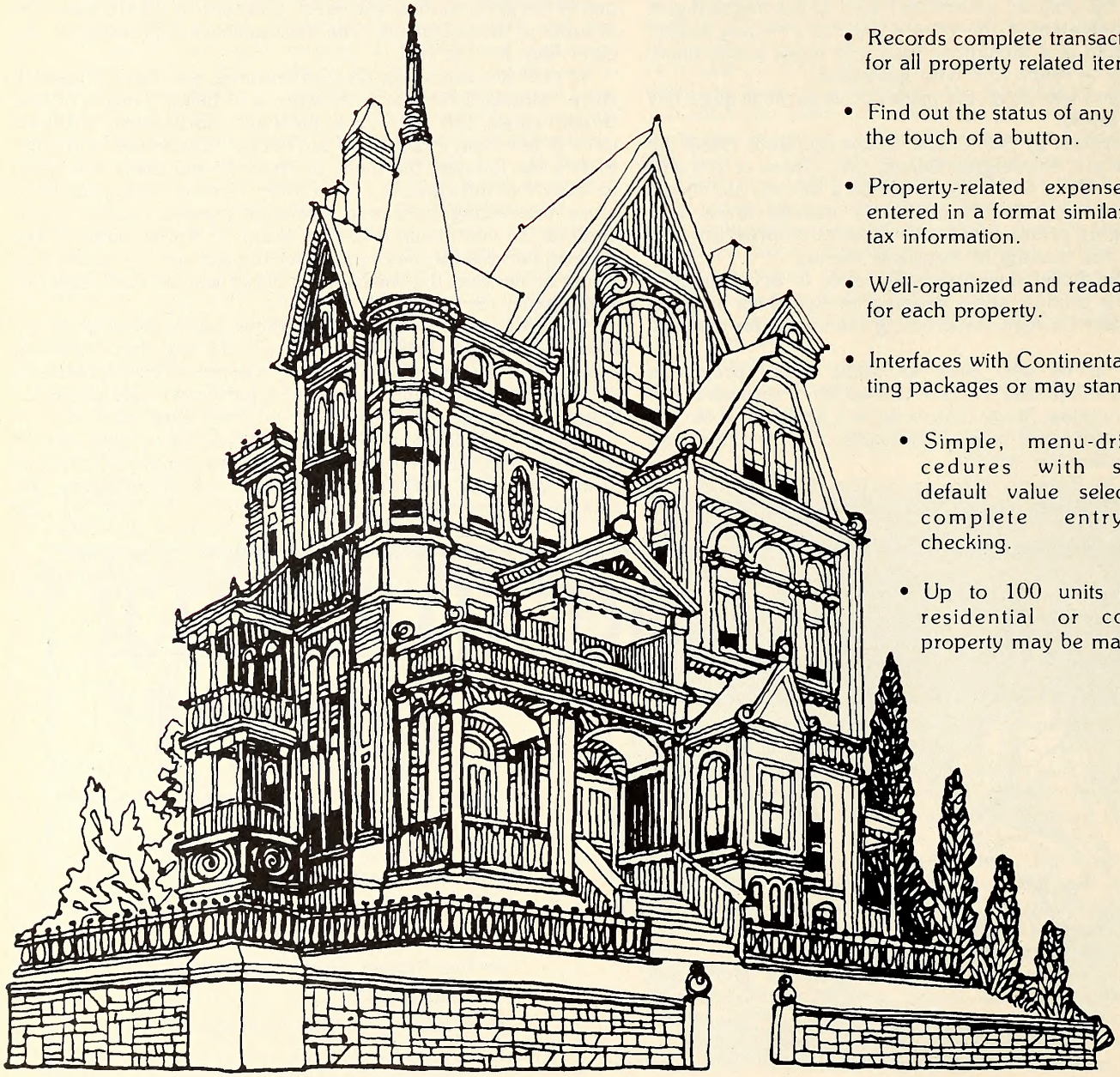
Overall, this taste of *Time Zone* served as an appetizer. If the other three-fifths comes up to these, the game is all it's said to be. According to On-Line, it doesn't; it's much better. We look forward to finding out. $99. ◨■

CONTINENTAL SOFTWARE

Computer Programmed Accountant

MODULE #5

PROPERTY MANAGEMENT



- Records complete transaction history for all property related items.
- Find out the status of any property at the touch of a button.
- Property-related expenses can be entered in a format similar to federal tax information.
- Well-organized and readable reports for each property.
- Interfaces with Continental's accounting packages or may stand alone.
- Simple, menu-driven procedures with simplified default value selection, and complete entry error-checking.
- Up to 100 units and either residential or commercial property may be maintained.

PROPERTY MANAGEMENT

$495

Modules in the Series

#1 GENERAL LEDGER #2 ACCOUNTS RECEIVABLE
#3 ACCOUNTS PAYABLE #4 PAYROLL
#5 PROPERTY MANAGEMENT

All modules are stand alone or may be used as a part of an integrated system.

CONTINENTAL SOFTWARE
16724 HAWTHORNE BLVD.
LAWNDALE, CA 90260

**APPLE][is a registered trade mark of Apple Computer Inc.
Hardware requirements: 2 disk drives, DOS 3.3, APPLE][48K, 132-col. printer and Applesoft in ROM.

Mind Your Business

BY PETER OLIVIERI

A very Happy New Year to all of you. I hope you were able to usher in 1982 in some suitable and memorable fashion. Before getting down to business, I'd like to suggest some New Year's Resolutions for Others. (This is much easier to do when the suggestions are for someone else!)

Be it resolved that any user's guide developed for a business application program or package shall be (a) proofread, (b) visually attractive, and (c) readable.

Be it resolved that if a program diskette is to be elaborately copy-protected, then a backup copy shall be supplied to the user along with the original.

Be it resolved that if a description is given to the user of what a particular program or package will display, actual samples of the screen listings, input forms, or output documents shall be provided in the user's guide.

Be it resolved that the proliferation of software in all sizes, shapes, and applications shall begin to settle down to more manageable proportions.

Be it resolved that costs of microcomputer peripherals shall decrease by a constant amount during each month of 1982.

Be it resolved that users of Apples in business environments shall share their talents, experiences, and resources with others.

With respect to this last suggestion, I'd like to launch a few new ventures this year. I would like to use this column as a vehicle to organize a business users group. Members might range from those of you who are currently using your Apple in your business to those who are just contemplating its use. The column would provide a centralized forum for members to share their experiences—describing the problems they faced in selecting software, developing their first application, choosing a printer, or acquiring a word processing package. The sharing of experiences, and of how particular problems were resolved, might mean that those who are just now venturing into the Apple applications arena will be spared similar problems.

Business User Profile. I'd also like occasionally to profile some of you who have successfully applied the Apple to a particular business. Please let me hear from you. Drop me a note in care of *Softalk* and I'll get in touch with you either by phone or in person. Of course, if you know of someone who is using the Apple in a small business, please let me know about them. These profiles can be a vehicle for sharing with others the steps necessary to take full advantage of the microcomputer. Who knows—your profile may just be the one to turn someone on to greater things.

Sources and Resources. Finally, I'd like to introduce a section of the column that will focus on the resources available to the business user who wants further information about a particular topic. These resources would include textbooks, reference books, management references, articles, computer books, catalogs, and so on. Since much material is published in these fields, it would be helpful to have a list of the best on a variety of topics. I have my own personal list. Please share some of your favorites.

Well, enough of things to come. It's time now to finish reviewing the accounting packages we started looking at a few issues back. After examining some of the details of each, we'll offer some comments that may help you identify the "best of the bunch" for your particular needs.

The Accountant Finance Data Base System. *The Accountant,* by Decision Support Software (McLean, VA), is a system for either home or office. It's more than a checkbook balancer. In fact, it's one of the nicest home packages I've seen. It would be a mistake, however, to exclude it from the small business setting since it has been designed for either environment.

The system requires an Apple II or II Plus with 48K, Applesoft in ROM, single or dual drives, and DOS 3.3. A printer is, of course, of significant benefit in any business application.

The package provides the following features: double-entry bookkeeping, a user dialogue that does not presume a background in accounting, relatively fast retrieval of information, a memory management feature, a built-in calculator, output reports designed for both screen and printer, an interface with *VisiCalc,* and very complete documentation.

One of the nice features of this package is its user guide. It is well written and easy to follow. All terms that might be new to the user are clearly defined prior to their use. An actual application is provided on diskette with the master programs and the first portion of the manual guides you through it. This demonstration section explains clearly each of the major features of the program. Not only can the user reference actual data and observe screen formats and reports, sample reports and forms are included in the guide itself.

The system allows the user to do double-entry bookkeeping. Entries can be made easily and retrieval of stored data is

unusually fast. You may specify up to sixty-three accounts and sixty-three codes. The *Accountant* stores each of the detailed transactions in its data base. This detail includes the date, a description, two accounts, an amount, and an optional code. You can examine the current balances in each account and print out monthly summaries or profit and loss statements whenever you require them. In addition, maintaining the data base is a relatively simple process. Changes can be entered rather easily, and all appropriate monthly balances are automatically updated.

A few special features of this package deserve mention.

1. The user can code a set of automatic transactions in a separate file that can be posted automatically at the user's request. Thus, if you have certain categories that remain the same from month to month, you do not have to enter the same data over and over again.

2. Each transaction can be assigned a code of your choice that will allow you to summarize all data with a particular code. This feature can come in handy around income tax time.

3. Special data compression methods are used to conserve space on diskettes. This means the user can store up to four thousand transactions on a single diskette.

4. The user guide contains a tutorial that is nicely done. With it, the user can learn all of the standard and special features of the package by actually selecting options and manipulating data in a data base provided on one of the diskettes. As an extra aid, screen formats or tables are printed in the manual itself.

5. The *Accountant* package interfaces with *VisiCalc.* You can specify what data you wish to have transferred into a format that you can use with *VisiCalc,* and then you can load the data onto a clear *VisiCalc* screen or onto a previously prepared template (such as a tax form). Of course, once there, all of the features of *VisiCalc* are now at your disposal.

Finally, the package allows you to make as many copies of the diskettes as you need. To prevent inappropriate distribution of the software, a key is supplied that must be inserted into the game port in order for the programs to work. This is easy to do and should present no problem at all to the user.

The quality of the documentation and the user friendliness of the programs impressed me, as did the care that was taken in program design. The programs certainly have applicability both in the home and in some small business settings (though *The Accountant* is not as complete a small business package as either the *Controller* or the Westware packages). The *Accountant* does, indeed, make financial management a simple and straightforward procedure.

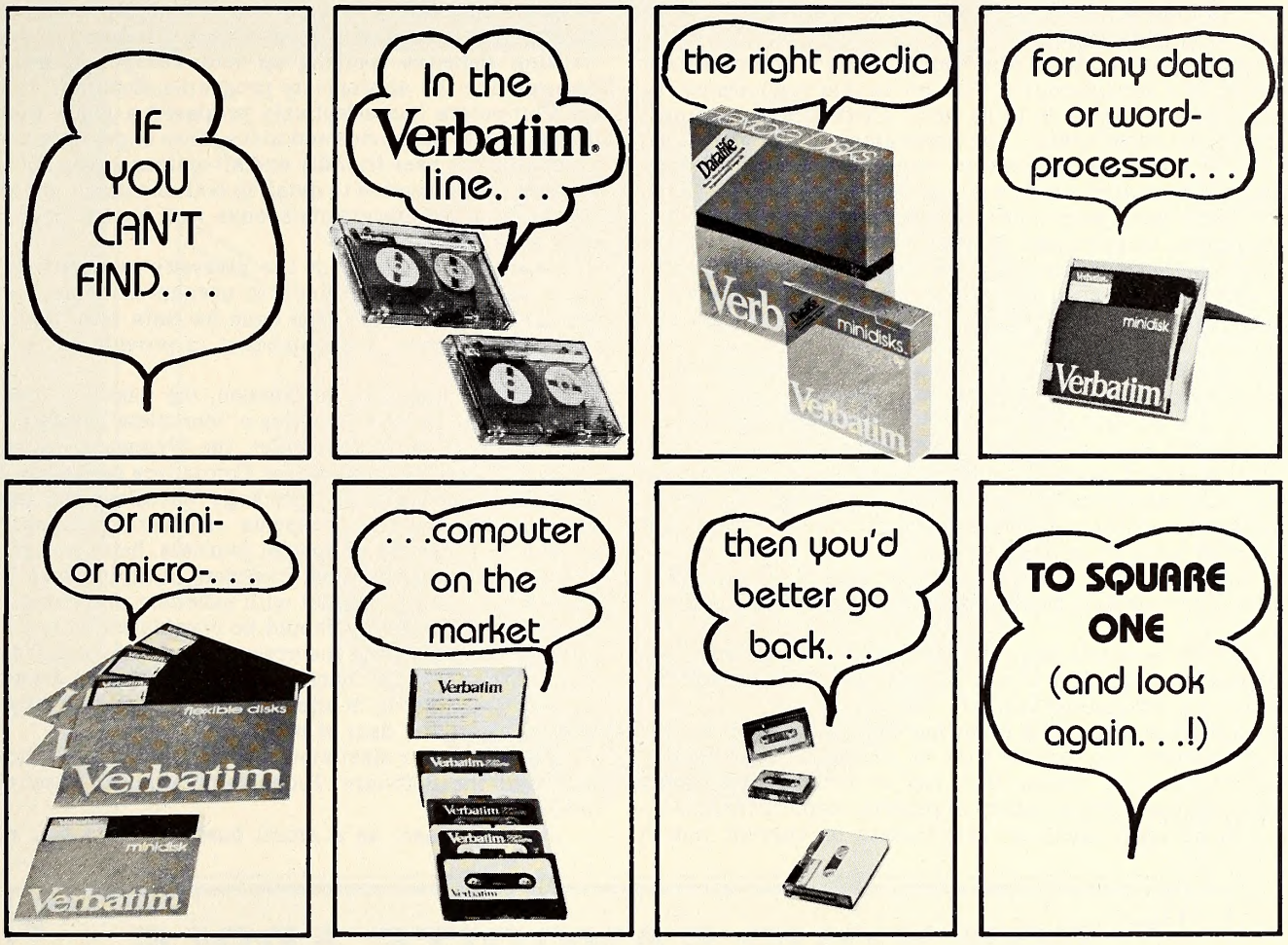
Business Series. The *Business Series* by Spectrum Software (Sunnyvale, CA) is a set of three separate programs designed to serve the needs of the small business user. The programs can be purchased together as a complete package or as individual modules. The three modules are the *Universal Business Machine,* the *Microaccountant,* and the *Business Check Register and Budget.*

The *Universal Business Machine* comes with eight standard financial planning and analysis models. Included are models to handle inventory, real estate investment, pro forma P & L statements, pro forma balance sheets, sources and uses of funds, sales and profit forecasting, and a job cost estimator. Part of this package includes a Spectrum product called the *Universal Computing Machine.* This general-purpose program allows the user to label rows and columns in a table, manipulate elements in the table, obtain totals for the rows and columns of the table, and ask a variety of what if questions by changing some of the table entries and observing the results.

The *Microaccountant* provides the user with a double-entry ledger system. You can maintain a transaction journal and a general ledger. The system can accommodate up to one thousand transactions per month and up to three hundred ledger accounts. A report generator produces journal listings, balances and transaction information for individual accounts, a balance sheet, and a net income statement.

The Business Check Register and Budget is most appro-

THE FLOPPY UN-PUZZLE



THERE'S NO PUZZLE . . . when you ask us for a Verbatim catalog and price sheet. Or we can answer most questions by phone.

WE PLAY IT BACK . . . Verbatim.

Every bit of every byte of every track of every disk is pretested. Meets all requirements for IBM, Shugart, ANSI, ECMA, ISO and APPLE.

HEAD CLEANING KITS Removes harmful contamination buildup from read/write heads for 8" and 5 1/4" drives.

FULL LINE . . . ALWAYS IN STOCK

OVERNIGHT DELIVERY* ***If order phoned in before 1 p.m. same day shipment.**

FSI

6901 Canby Avenue, Reseda, CA 91335
(Los Angeles) Phone (213) 705-4202

2680 Bayshore Frontage Road, Suite 1030, Mountain View, CA 94043 (415) 962-9275

LARGEST MASTER DISTRIBUTOR FOR VERBATIM REMOVABLE MAGNETIC STORAGE MEDIA

priate in situations that primarily involve cash transactions. The program can handle up to four hundred checks per month and up to sixty check accounts.

In order to use these programs, you must have an Apple II or Apple II Plus, Applesoft in ROM, 32K or 48K RAM, DOS 3.2 or DOS 3.3, and a printer.

The *Universal Business Machine* is really a computerized worksheet. It's reminiscent of *VisiCalc* in that it allows you to manipulate data that is in table form. Of the three modules, this one is the most useful. The generalness of its design allows you to adapt the package to your own particular needs.

The documentation (user's guide) was disappointing, however. It could have been more thorough and could have illustrated various applications in more detail. In some of the manuals, the sample listings were too light to be readable. While this may be the result of a faulty copying machine, it certainly affects the perception of the user concerning the quality of the product.

Software Technology for Computers. Software Technology for Computers (Belmont, MA) has prepared a variety of programs for the general market. Their offerings include an accounts payable package, an advanced payroll package, a data base management system (to be reviewed in a forthcoming column), a professional time and billing package, an inventory control program, a graphics coloring board program, and a variety of other programs.

We will look now at their accounts payable package (and in the future at their data base system and their graphics package).

To use the accounts payable package you must have two disk drives, an Apple II or Apple II Plus, 48K, DOS 3.3, and for best results a 132-column printer.

The software offers the following features: creation of a chart of accounts (up to sixty may be created), preparation of invoices, creation of vendor files (up to one hundred twenty vendors), and printing a variety of reports. Reports include information on open invoices, year-to-date or current month cash disbursements, actual checks, cash requirements, and a complete check register. The transaction file will hold up to 725 transactions.

The user's guide is easy to read and includes all of the information necessary to make you thoroughly familiar with the system. The guide carefully describes the steps involved in initializing diskettes, backing up data, recovering from errors, and running the appropriate programs. Software Technology for Computers has effectively prepared a guide that will be useful both to the novice and to the more experienced user. The programs are user friendly and all options easily accessed via a menu. The attention to detail in both the user's guide and the operation of the programs speaks well for the accounts payable package.

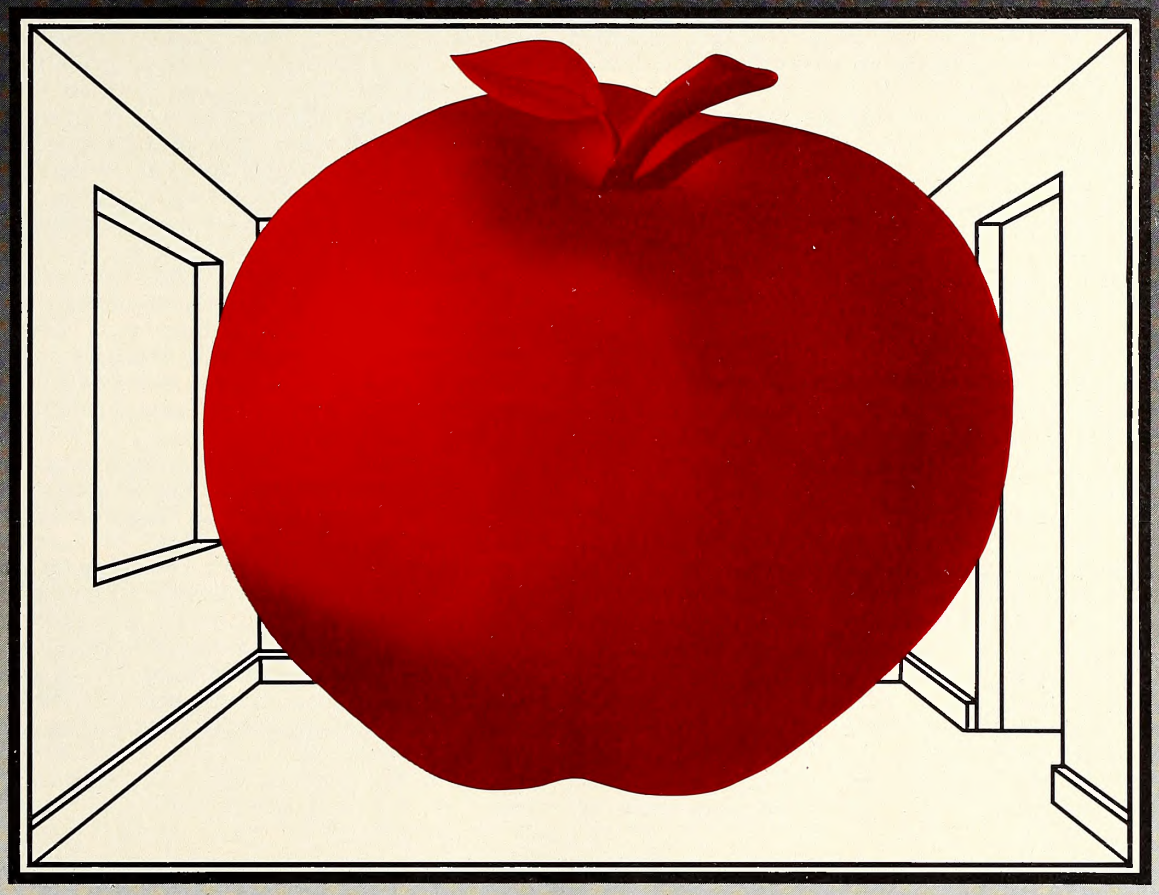
Summary. Because of the diversity encountered among these packages, it's difficult to prepare a comparison table similar to the one that was done for data base management systems. However, it's important to provide some basis for comparison.

Given the available information, the following observations can be made. If you're seeking a "complete" system, the candidates would be the *Controller,* the Westware Package, or a set of Software Technology for Computers packages. By complete, we mean a thorough package that allows the user to handle accounts payable, accounts receivable, general ledgers, as well as a variety of special journals, listings, and reports. The *Controller* is the most complete of all of those we evaluated. It's a quality product with excellent and complete documentation. It certainly should be considered by anyone interested in computerizing the accounting and financial aspects of a small business. Of course, it is not without its drawbacks; a lot of disk swapping is often necessary, and learning the system takes a good deal of time.

If you'd rather start slow and develop your applications in modules, the Software Technology products may suit your needs best.

In either case, as a small business user, you should be

EXPAND YOUR APPLE.



Z-CARD

Let **Z-CARD** transform your Apple™ into a powerful CP/M™ operating system. With **Z-CARD**, you take advantage of the largest software base available — over 2,500 programs for word processing, forecasting, data management, accounting and more.

Z-CARD offers low power consumption and high reliability with powerful, simple to use CP/M™ software that increases the speed of your system. Our software offers full keyboard mapping and format/copy in one operation. **Z-CARD** is easy to install — just plug it in and it's at your command, no modifications needed. The full feature **Z-CARD** offers you all this and more for an amazingly low price.

Now there is a choice. . . expand your Apple™ with **Z-CARD**.

For more information see your Apple™ dealer or call **800-538-8177**, in California **408-727-6805**.

als Advanced Logic Systems

1195 E. Arques Avenue, Sunnyvale, California 94086 [408] 727-6805

Apple is a Trademark (TM) of Apple Computer, Inc., CP/M is a Trademark (TM) of Digital Research, Inc.

aware of the possibility of needing to eventually move to a hard disk and should investigate whether a particular vendor is developing products for this market. It may make a significant difference when you upgrade.

For the home user (and perhaps in some less complex small businesses), the best package we evaluated was the *Accountant* by Decision Support Software. The documentation was very well done, programs were easy to use, and design was thorough and complete.

For your reference, the approximate cost of these packages is listed here:

The Controller	$ 625.00
Westware *Systems II*	$1,295.00
The Spectrum Series	$ 159.00 (for all 3)
The Accountant (+$20 for the *VisiCalc* interface package)	$ 89.95
STC Accounts Payable	$ 200

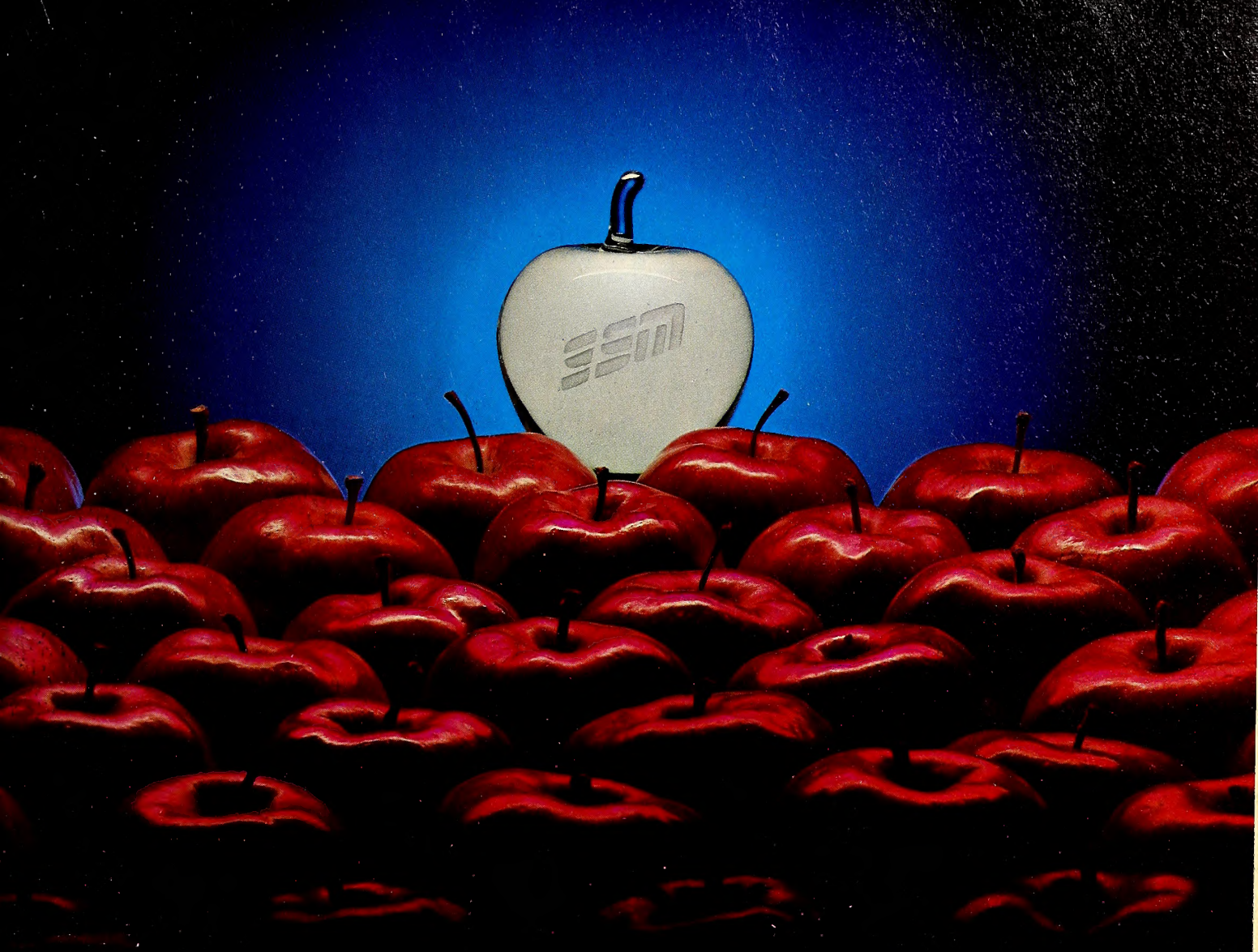
The Readers Speak. "Sometimes I am interested in buying a program or piece of software but cannot find any reviews of it. The package may be a game or specialized program that just didn't make the published scene as yet. How can I find out about a package before I pay for it? Are there questions I can ask or places I can go for information?" G.F., New York.

Evaluating software without actually running the programs yourself is very difficult. Reviews are considered an important part of most computer publications precisely because potential users do not often get to use a package before they buy it; the review provides a real service to them. Unfortunately, not every program gets reviewed and, sometimes, a publication is reluctant to say anything really bad about a product. When considering the purchase of software packages that cost $2,000 to $10,000, users in the minicomputer or medium-size computer market are highly motivated to be aggressive and demand a demonstration and/or an opportunity to speak with other users. With microcomputer packages that sell for $29.95 or perhaps even $200, it seems that many consumers are less aggressive than they might be.

Your first step should be to try to speak with someone who has actually used the product, perhaps one of the dealers in your local area. If you cannot find reviews of the product or local dealers who are familiar with it, then you might consider visiting a library. A library associated with a college or university with a computer science program is your best bet. At the library, ask if they have any publications that give profiles on computers or software. One such publication is published by a firm called Datapro. They provide several three ring binders that list and evaluate small computers, microcomputers, software, and so on. You may get lucky and find a description of the program you're interested in, along with information about how users have reacted to it.

A second step might be to call the vendor. Ask if you can talk with anyone who is currently using the program. If the vendor can't supply you with any such contacts, then I would wonder about the acceptability of the package. To be sure, the vendor is unlikely to have you speak with someone who hates the package so it's wise to evaluate carefully the responses you get from such sources.

The problem you're facing is not a unique one; it's perhaps the most difficult problem facing the small business user. I recently spoke at a workshop for dentists who were interested in computerizing their offices. They were interested in a complete package (both hardware and software). As they became more and more involved in gathering data about what was available, they felt more and more confused. They wondered why one system costs $28,000 and another (seemingly developed to achieve the same result) would cost $10,000. Since there is, indeed, a lot the potential user can do to become more proficient at evaluating and selecting among alternative systems, a portion of next issue's column will be devoted to precisely that topic: A Guide To Computerizing Your Business: Doing Your Homework. ■



Every Apple Is Created Equal. But It Doesn't Have To Stay That Way.

An Apple™ is really something.

But from now on, it's going to be something much more. How come? Simple. We're introducing three revolutionary new data communication packages, called Transend™, designed especially to make your Apple II™ rise above all others.

Our Transend data communications software provides access to information utilities; complete file transmission of charts, VisiCalc™ reports, and entire programs; as well as all electronic mail functions.

Transend comes in three different versions. You can begin economically with the simplest form, Transend 1, an intelligent terminal/file transfer system. As your needs increase, you can move up to two other Transend packages, without worrying about retraining or repurchasing. Transend 2 gives any Apple II intelligent terminal capabilities, plus file transfer with complete error detection and automatic retransmission for demanding business applications. Transend 3 incorporates electronic mail—with password security, text editing, mailbox with mailstop, unattended scheduling and receiving, automatic redialing, and much more.

SSM

The Transformation People

Transend's support of most popular Apple II "add-in" cards and modems lets you upgrade your Apple without hassle or unnecessary expense. And you'll immediately cut operating costs by as much as 30% . with simple, easy-to-use menus, data compression and 1200-baud modem support.

Nobody on the personal computer scene today is offering the speed, flexibility, and reliability of Transend in a single upgradable software system.

Don't miss this chance to uplift your Apple II. Contact us or your dealer for all the delicious details.

SSM Microcomputer Products, Inc., 2190 Paragon Drive, San Jose, CA 95131, (408) 946-7400, Telex: 171171, TWX: 910-338-2077.

LISP Arrives To Up Your Apple's IQ

BY ALLEN MUNRO

"The lab has just ordered me a personal computer!" exulted the Whiz Kid. I was a bit surprised. The Kid had a Ph.D. in artificial intelligence and was working on the frontiers of computer science. I reflected smugly that I'd been using a personal computer for almost three years.

"What, an Apple?" I asked, prepared to bestow my wisdom born of experience. The Whiz Kid giggled, not sure that I was serious.

"No, no," he said, "a Lisp personal computer. You know, better performance than a dedicated DEC KA-10, forty to ninety thousand dollars. What good is a personal computer without Lisp?"

I was about to try to tell him what good a personal computer without Lisp could be—about electronic spreadsheets, word processing, mailing list programs, and adventure and arcade games, not to mention the convenience of Basic and the elegance of Pascal—when I realized it would all mean nothing to him. To workers in artificial intelligence, there is only Lisp—and a few even more exotic "object-oriented" languages such as Logo and Smalltalk. The history of artificial intelligence—the science of making machines think, often much like human beings—and the computer programming language Lisp are intimately intertwined.

Lisp was developed by John McCarthy of MIT in the 1950s in order to facilitate the development of programs that exhibit purposive problem-solving behavior and can process natural language. Research projects using Lisp have produced *Eliza,* a program that interacts with the user like a Rogerian or "client-centered" psychotherapist, chess-playing programs, pattern-recognition systems, intelligent robots, and Shrdlu, a system that interprets English commands to move blocks about on a table surface. Artificial intelligence is expected to play an increasingly important role in providing new application domains for the powerful, inexpensive computers of the late 1980s. Lisp is likely to be important in this process.

The Whiz Kid's personal Lisp computer is real. Xerox refers to its $59,719 Dolphin/Interlisp machine as a "medium sized personal computer." This system and its competitors, the Symbolics LM-2 and the Foonly F5, all offer a megabyte or more of main memory, 23-80 megabyte hard disk drives, Lisp optimization, and many programming environment support features.

"How about color graphics?" I asked the Kid.

"An extra cost option," he admitted.

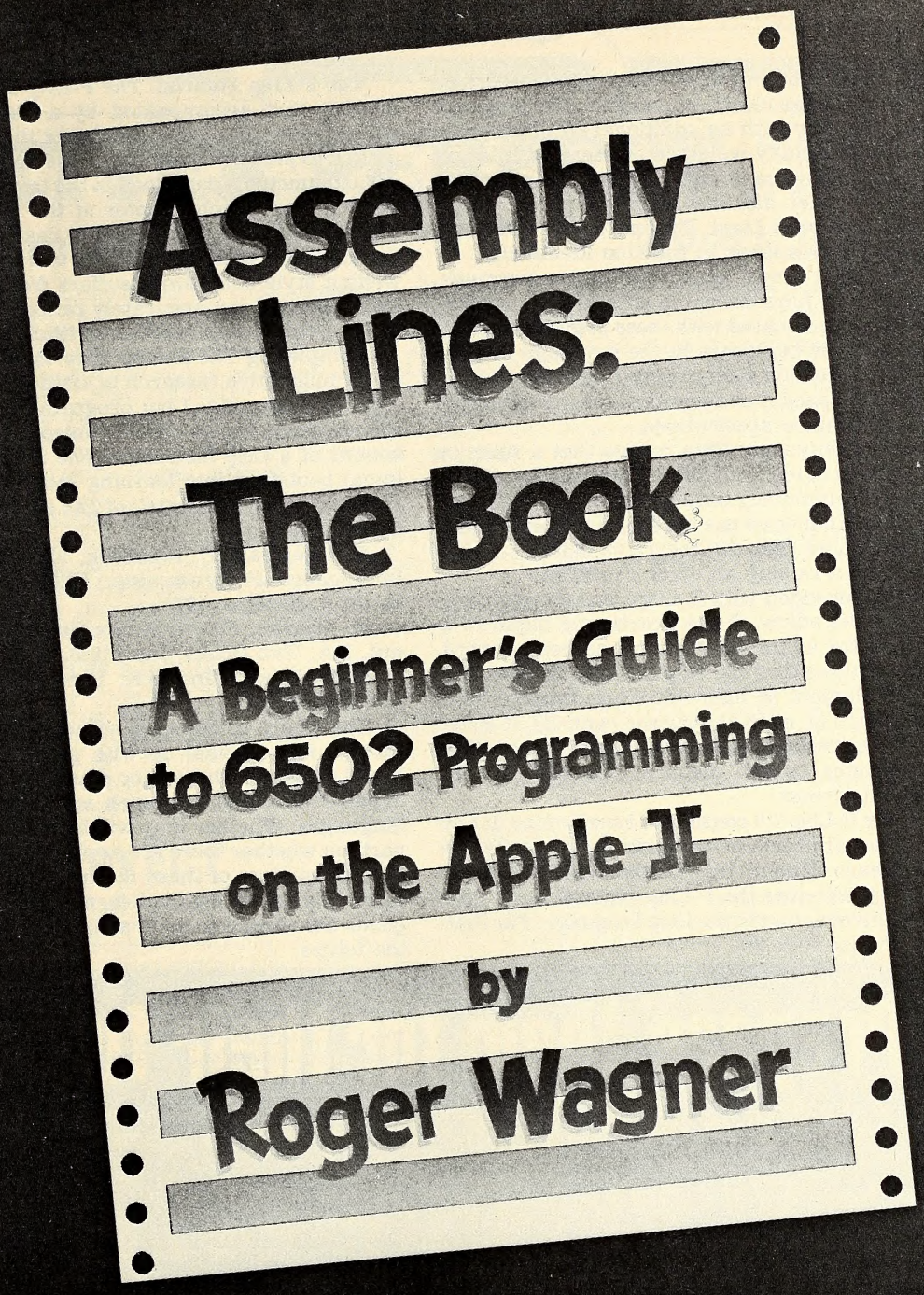
Well, Lisp is now available on the Apple II, and lo-res color graphics is even included at no extra charge. Total system cost is well below the Whiz Kid's system, as well: all you need is a 32K Apple II (48K recommended) and a $99.95 P-Lisp software package from Gnosis (Philadelphia, PA). The package includes two versions of the P-Lisp system, one for 32-48K Apples and one for 64K Apples. In addition, it offers a Lisp editor with a pretty-print function for consistent screen formatting of Lisp functions. Two demonstration programs are included: a version of the *Eliza* therapist program, and a *Pig Latin* program that converts English input to Pig Latin. Of course, the P-Lisp package offers something less in speed, maximum program size, and programming environment features than "personal computers in the $40,000-$90,000 price range." It does provide the user with an excellent opportunity to learn Lisp by using Lisp.

To this end, Gnosis is about to release *The P-Lisp Tutorial,* designed to introduce the reader to Lisp, using the P-Lisp system. The text is $15 and an accompanying diskette is $5. The combination of an inexpensive microcomputer Lisp and a textbook specifically geared to that implementation of the language has the potential to liberate Lisp from the hothouse academic atmosphere that has nurtured it for the past twenty-five years.

P-Lisp. P-Lisp on disk is an Apple DOS-based system (Apple DOS commands such as *bload* and *brun* are available). Fifty-seven Lisp functions are standard in the system, including several for control of Apple lo-res graphics. There are no built-in functions for reading the Apple paddles or controlling the Apple's built-in speaker, but there are *peek* and *poke* functions, which permit the user to write paddle and tone functions. One of the characteristics of Lisp systems is that users tend to add functions to develop a personal version of the language and programming environment.

Two sample programs are included as examples of P-Lisp applications. The sample programs provided will stimulate many users to try to improve them. For example, the *Pig Latin* program recognizes spaces, but not punctuation, as word boundaries. As a result, the transformed words put out by the program can have embedded punctuation marks. The Gnosis version of *Eliza* is entertaining, but evidences somewhat less wit, variety, and "insight" than is revealed in the original *Eliza* dialogues given in Joseph Weizenbaum's *Computers and Human Reason* (San Francisco: Freeman, 1976).

Bringing up either of the P-Lisp systems is quite simple. To use the normal 32-48K Lisp, one can simply boot the Gnosis P-Lisp disk and then type *brun Lisp.* If the system has a 16K memory card in slot 0, such as an Apple Language System card or a Microsoft RamCard, then one has the option of



A SOFTALK BOOK $19.95

When Roger Wagner isn't writing utility programs for the Apple computer, planting corn in his backyard, publishing communications and utility software, or singing on the beaches near San Diego, he contributes a monthly column to Softalk.

Called Assembly Lines, it deals with assembly language programs at a fundamental level and has caused more comment and more mail than any other feature in the magazine. Because of the large number of requests, Softalk is now reprinting all Assembly Lines columns through December 1981 in a compact form, bound with spiral binding to make reference easy.

The columns are fleshed out with additional comments and the book will contain handy reference information that will put the information you need to program in assembly language in one neat package.

If you are an aficionado of Roger's column, or if you've got a driving desire to speak to the Apple in assembly language, there is no better way to start than by reading Assembly Lines: The Book by Roger Wagner.

To receive a copy, send $19.95, plus $1.50 for postage and handling, to Softalk Book, 11021 Magnolia Boulevard, North Hollywood, California 91601.

using the normal P-Lisp system in the manner just described or using the special language card Lisp version. The language card Lisp provides the user with an additional 8K of program space, which will be especially welcome to users with larger applications. To boot the language card version, one must first bring up a DOS 3.3 master diskette, then insert the Gnosis P-Lisp diskette and type *brun Lispl*. Except for the additional memory, the two systems appear to function identically.

Gnosis Lisp, like most Lisps, is an interpretive language. When the user types in a function name with arguments, the function is immediately evaluated with those arguments. This is an excellent feature for beginners in the language and the system—there is an immediate system response to every completed input. There are speed disadvantages for the advanced user with large and complex applications.

Lisp is a recursive language. This means that a function may be partially defined in terms of itself. When the function is used, it will call itself recursively until some terminating condition is reached. The productive use of recursion is somewhat limited in P-Lisp. No more than 128 recursive calls of a function are possible without raising an error condition.

An editor utility is provided with P-Lisp. One simply types *(load edit)* to invoke the editor. Some practice is required to make effective use of the editor, which is not screen-oriented. Instead, the editor requires that one step through the lists that comprise a function element by element. Users familiar with the Apple Pascal Editor or even the cursor controls of Basic are likely to find this a frustrating exercise. The P-Lisp editor includes a pretty-printer, which displays functions on the screen in a consistent format.

Documentation for P-Lisp 2.0 consists of twenty-four densely written pages. Several sheets of addenda for the enhancements of the 2.1 release accompany the diskette. The documentation accurately describes the P-Lisp system, but it does not provide an easy introduction to the Lisp language. For that, Gnosis offers *The P-Lisp Tutorial*.

The P-Lisp Tutorial. *The P-Lisp Tutorial* is a printed introductory text accompanied by a diskette with a P-Lisp file called *book*. The text introduces the reader to Lisp through guided exploration of P-Lisp. The *book* file provides a number of Lisp functions described in the text but not available in standard P-Lisp. Each chapter of the text is short. Only five to twenty minutes are required to read a chapter, work through the example, and try out a few test Lisp lists of your own. The writing style sometimes suffers from labored wit and excessive cuteness, but accurately reflects the lighthearted nomenclature commonly found in artificial intelligence programs.

Completing *The P-Lisp Tutorial* will not prepare the reader for innovative research in artificial intelligence, or even for writing substantial Lisp programs. However, it will prepare the reader to tackle meatier introductions to Lisp without the benefit of a Lisp instructor. *The P-Lisp Tutorial* is an educational bootstrap into learning more about Lisp. Three recent Lisp texts the "graduate" of *The P-Lisp Tutorial* might want to consider are:

E. Charniak, C. Riesbeck, and D. McDermott. *Artificial Intelligence Programming*. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1979.

K. Tracton. *Programmers Guide to LISP*. Blue Ridge Summit, PA: Tab Books, Inc., 1980.

P.H. Winston and B.K. Horn. *LISP*. Reading, MA: Addison-Wesley, 1981.

The future of personal computing is thought by many observers to be bound up with powerful artificial intelligence-based systems. There is no doubt that future hardware will be capable of supporting such sophisticated environments and languages. Whether they will be fully exploited will depend, in part, on whether there are applications programmers who can take advantage of these features. The P-Lisp package offers those who have not had formal training in artificial intelligence the chance to start acquiring the programming skills of the future. ◨

LETTER PERFECT

T.M. LJK

WORD PROCESSING

ATARI 400/800 — APPLE II & II+

EASY TO USE — Letter Perfect is a single load easy to use program. It is a menu driven, character orientated processor with the user in mind. FAST machine language operation, ability to send control codes within the body of the program, mnemonics that make sense, and a full printed page of buffer space for text editing are but a few features. Screen Format allows you to preview printed text. Indented margins are allowed. Data Base Merge with **DATA PERFECT** by LJK, form letters, accounting files and mailing labels only with MAIL MERGE/UTILITY by LJK. **FEATURES** — Proportional/Incremental spacing * Right Justification * File Merging * Block movement * Headers * Footers * Print Multiple Copies * Auto Page Numbering * Scroll forward/backward * Search and Replaces * Full cursor control * Underlining * Boldface * Superscripts * Subscripts * Auto page numbering * Insert character/line * Delete character/line * Centering * Horizontal tabs/changeable * Multifunction format line (line spacing — left margin — page width — lines/page — change fonts — top/bot margin adjust) **MUCH MORE! $149.95**

ATARI VERSION 2.0 #2001

Compatible with Atari DOS. Uses proportional font, right justified with Atari 825/Centronics 737, 739 printers. Uses EPSON MX* Series + Graftrax/italicized font. Can mix type fonts on same page; mix boldface and enhanced font in same line with justification. Can be used with 16K Atari/400.*

"Compared to the price of many other word processors, this package is a steal. It does everything the advertisement claims and more. On top of this the software is very easy to use." **A.N.A.L.O.G. MAGAZINE**

APPLE VERSION 5.0 #1001

DOS 3.3 compatible — Use 40 or 80 column interchangeably (Smarterm — ALS; Videoterm-Videx; Full View 80 — Bit 3 Inc.; Vision 80 — Vista; Sup-R-Term — M&R Ent.) Reconfigurable at any time for different video, printer, or interface. **USE HAYES MICROMODEM II*** LCA necessary if no 80 column board, need at least 24 K of memory. Files saved as either Text or Binary. Shift key modification allowed. Data Base Merge compatible with **DATA PERFECT*** by LJK.

"For $150, Letter Perfect offers the type of software that can provide quality word processing on inexpensive microcomputer systems at a competitive price." **INFOWORLD**

DATA PERFECT T.M. LJK

APPLE & ATARI DATA BASE MANAGEMENT $99.95

Complete Data Base System. User orientated for easy and fast operation. 100% Assembly language. Easy to use. You may create your own screen mask for your needs. Searches and Sorts allowed, Configurable to use with any of the 80 column boards of Letter Perfect word processing, or use 40 column Apple video. Lower case supported in 40 column video. Utility enables user to convert standard files to Data Perfect format. Complete report generation capability. **Much More!**

EDIT 6502 T.M. LJK

This is a coresident — two pass **ASSEMBLER, DISASSEMBLER, TEXT EDITOR, and MACHINE LANGUAGE MONITOR.** Editing is both character and line oriented. Disassemblies create editable source files with ability to use predefined labels. Complete control with 41 commands, 5 disassembly modes, 24 monitor commands including step, trace, and read/write disk. Twenty pseudo opcodes, allows linked assemblies, software stacking (single and multiple page) plus complete printer control, i.e. paganation, titles and tab setting. User can move source, object and symbol table anywhere in memory. Feel as if you never left the environment of **BASIC.** Use any of the 80 column boards as supported by **LETTER PERFECT,** Lower Case optional with **LCG.**

LJK DISK UTILITY APPLE $29.95

This menu driven program allows the user to manipulate a variety of different file types. Binary, Text, and Source files may be easily converted into each other. The program may be used with **APPLESOFT*, VISCALC***, and other programs. These program files may be readily adapted for multiple use including editing with **LETTER PERFECT** word processings.

$29.95 MAIL MERGE/UTILITY APPLE & ATARI

This menu driven program combined with **LETTER PERFECT** allows user to generate form letters and print mailing labels. With the Atari, you may **CONVERT ATARI DOS FILES,** or Visicalc files compatible for editing with **LETTER PERFECT.** Utility creates Data Base files for Letter Perfect.

LOWER CASE CHARACTER GENERATOR $34.95

Lower Case Character Generator for the Rev. 7, Apple II or II+ computers. When installed, this Eprom will generate lower case characters to the video screen. Lower case characters set has two dot true descenders. Installation instruction included. Manual includes listing of software for full support and complete instructions for shift key modification. Compatible with **LETTER PERFECT.**

COMPUTER BASED SOFTWARE — LJK ENTERPRISES INC.™

LJK ENTERPRISES INC.
P.O. Box 10827
St. Louis, MO 63129
(314) 846-6124

DEALER INQUIRES INVITED

*Trademarks of: Apple Computer — Atari Computer — Epson America — Hayes Microcomputers — Personal Software — Videx — Bit 3 Inc. — M&R Ent. — Advanced Logic Systems — Vista Computers

ONTESTWINNERSCONTESTWINNERSCON

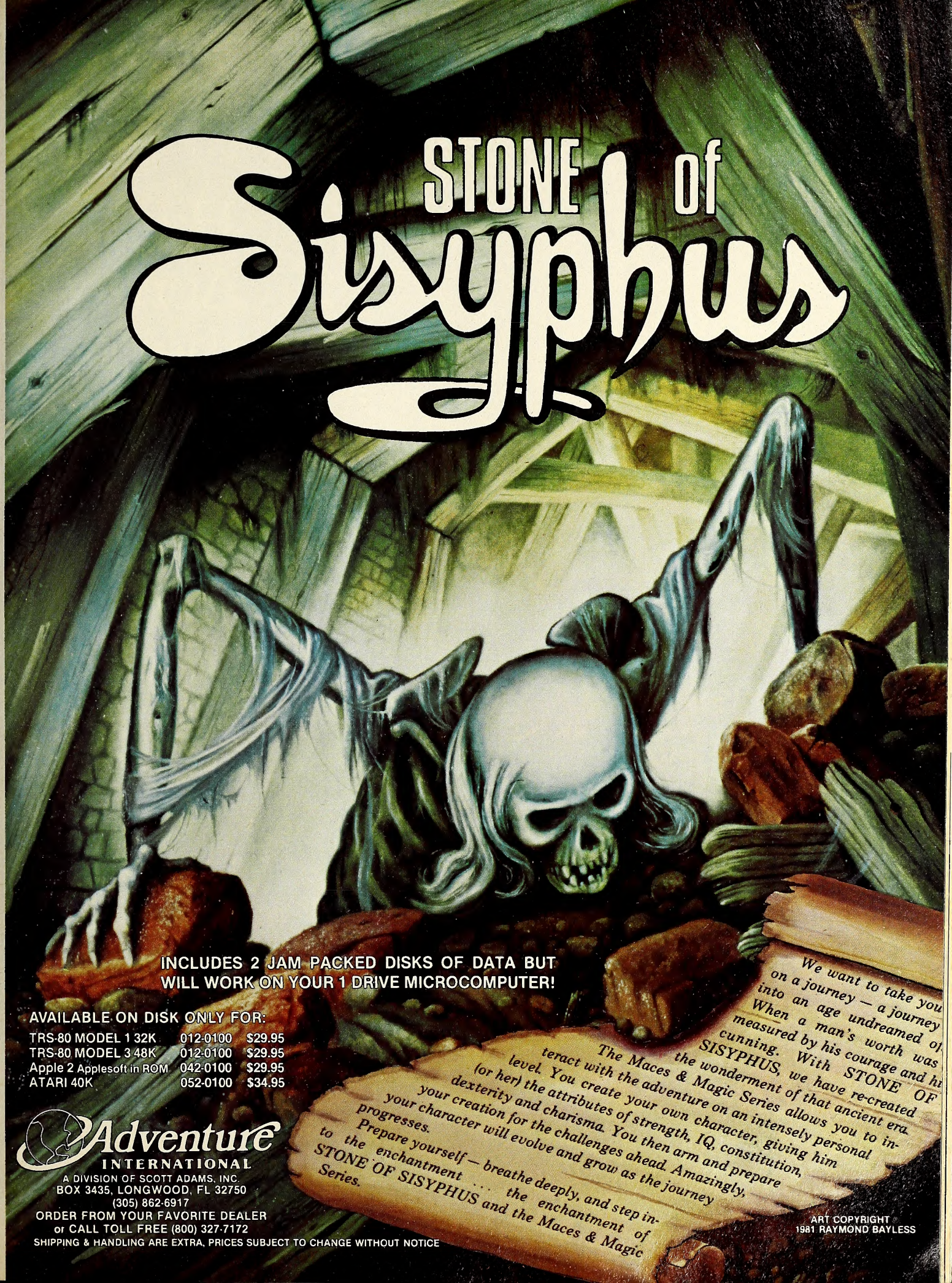
from page 4

word), he said. "Fill the gun" (definition).
17. Stock. Singular (make answer singular—it is ordinarily plural by this clue) punishment too old (definition: old punishment was public confinement in stocks) for Wall Street (association).
20. Programs. *Gramps* role (anagram) is no good on *TV fillers* (definition).
23. TV. *T*own and *v*illage initially (take initials) went for *home entertainment* (definition).
24. Secret. Half a par*sec* (drop *par*) and part of the *ter*rain (drop *rain* and anagram) later (than *sec*), Stepan remained *undercover* (definition).
26. Filing. A short *life* (drop e) is badly spent (anagram *lif*) *putting away papers* (definition and ing).
28. List. The *inventory* (definition) is *still* too much (drop l from *still*) confused (anagram *stil*).
32. Boot. "*Boo*," the (hidden word) ghost said as he sloshed away in his *galoshes* (definition).
33. CIO. Don't *show what you're doing* (functional definition) to the *union* (second definition). *See, I owe* (pun) you one.
35. Chat. A French cat (*chat* is French for "cat") loves to *converse, shortly* (definition).
37. Modem. A *sixties-style* (definition of first part: *mod*) runner, he used the *telephone connection* (definition of whole) to call in *a long dash* (definition of second part: *em* dash).
39. Source. Michener's book (you had only to choose which one) often tells about a *fountainhead* (definition).
40. Roleplaying. Tuesday found *Polly in rage* (anagram) about *acting* (definition).

Down

1. End. M*ending* sessions stop (definition) early (drop *ing*) when headless (drop *m*).
3. Pascal. *Al's cap* (anagram) *blazed* (pun on Blaise, Pascal's name) in the sun, in any *dialect* (definition).
4. Fantasy. A *wily* (with y) *addict* (fan) *sat* (anagram y fan sat), locked in *daydreams* (definition).
5. PROM. *Dance* (definition), and *make memories* (function—what you can do with a PROM; also definition) to last forever (what ROM does as opposed to RAM).
7. Ranch. Trading *you for a* (change *u* to *a*) *churn* (anagram after switch) occurs on a *cattle farm* (definition).
8. RAM. You lose your *mind* (definition; also, whole phrase—what happens if you turn off your computer?) if it's *marred* (backward hidden word) enough.
13. Computers. *Recut mops* (anagram) don't last as long as *microprocessors* (definition).
14. Puzzle. An *enigma* (definition), snoring (zz), makes baby pule (insert zz).
16. Disk (entrants who spelled this disc were counted as correct). *Swerving* (definition of skid) in circles (clue to anagram skid) can cause *spinal parts* (definition) to hurt.
18. Apple. *Pale P*atsy (anagram pale and initial P) insisted upon the *best computer* (definition) in school.
19. Code. Ar*co de*plores (hidden word) *symbolic substitution* (definition).
21. Adventure. Seventeen *true vaned* (anagram) weathercocks went on an *escapade* (definition).
22. Lights. An actress plants *bulbs* (definition) under the glow of *Big Berthas* (second definition).
24. Save. After the *rescue* (definition), he received a huge *vase* (anagram) of daisies.
25. Run. People who *jog* (definition) are partially *nur*ds (drop *ds*) returning (reverse *nur*).
26. Finance. *Put up the money* (definition) or they *can fine* (anagram) you.
27. Inventory. Confused, *Roy* (anagram *roy*, part of charade) uses *everything he owns* (definition of whole) to *think up* (definition, other part of charade) an alibi.
29. Integer. Endless *greetings* (drop end, *gs*, and anagram) confuse *whole numbers* (definition) of people.
30. Command. "*Come and* (pun) get it!" (illustrative definition) the cook shouted when our *order* (definition) was ready.
31. Micros. Every morning, *Mike rose* (pun) early to work on his *Apple and Atari* (definition).
34. Voice. Swirling *ice ov*ercame (anagram) the *soprano* (definition) as the Titanic went down.
36. Range. The new *stove* (definition) can do *everything from* melting butter *to* baking pottery (illustration).
38. Get. A *sage trainer* (hidden word) teaches her dog to *fetch* (definition).

A delightful side-benefit of this contest was that we heard from many readers who hadn't entered any previous *Softalk* contest, and that raises the possibility of readers who'd like to see more of this kind of puzzle, perhaps as a regular feature in addition to the monthly contest. Drop us a line if you'd like to see a regular puns 'n' anagrams crossword in *Softalk;* please note whether you'd be interested in it only as a contest with prizes or if you'd enjoy having it there just for the fun of working it. A postcard will do to *Softalk Puns 'n' Anagrams,* 11021 Magnolia Boulevard, North Hollywood, CA 91601; if you address it that way, all you have to write is *Yes* or *No,* and *as contest only* or *for fun.* ■



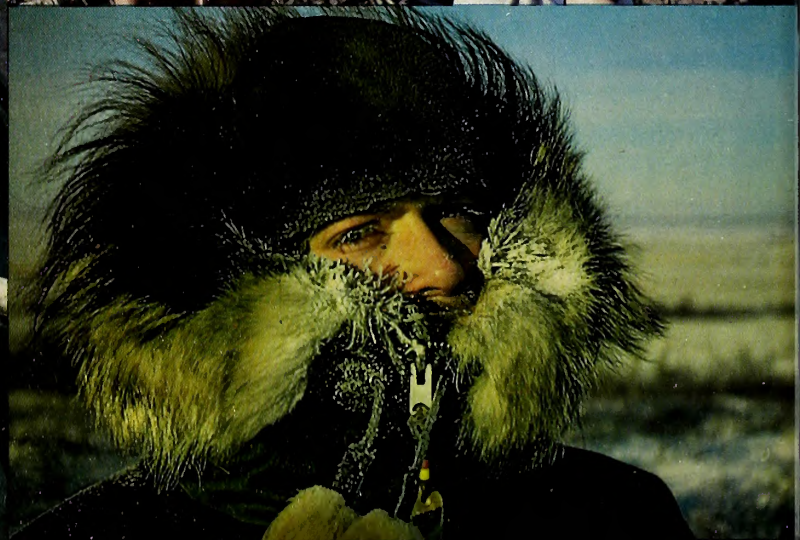
INCLUDES 2 JAM PACKED DISKS OF DATA BUT WILL WORK ON YOUR 1 DRIVE MICROCOMPUTER!

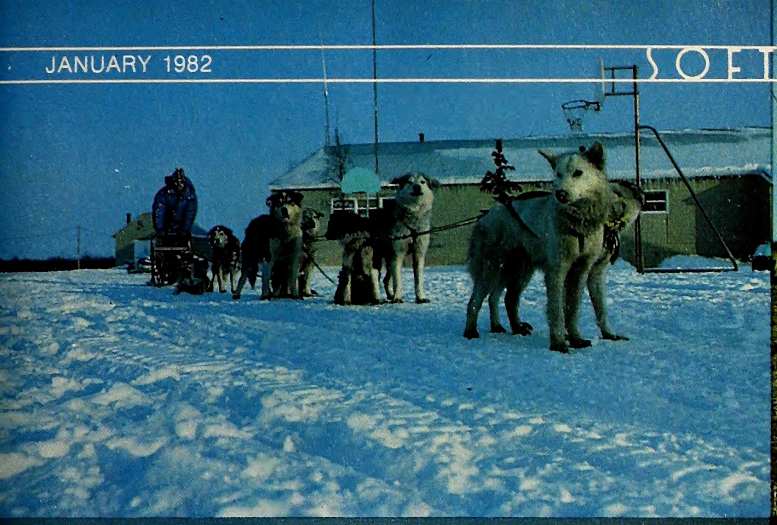
AVAILABLE ON DISK ONLY FOR:

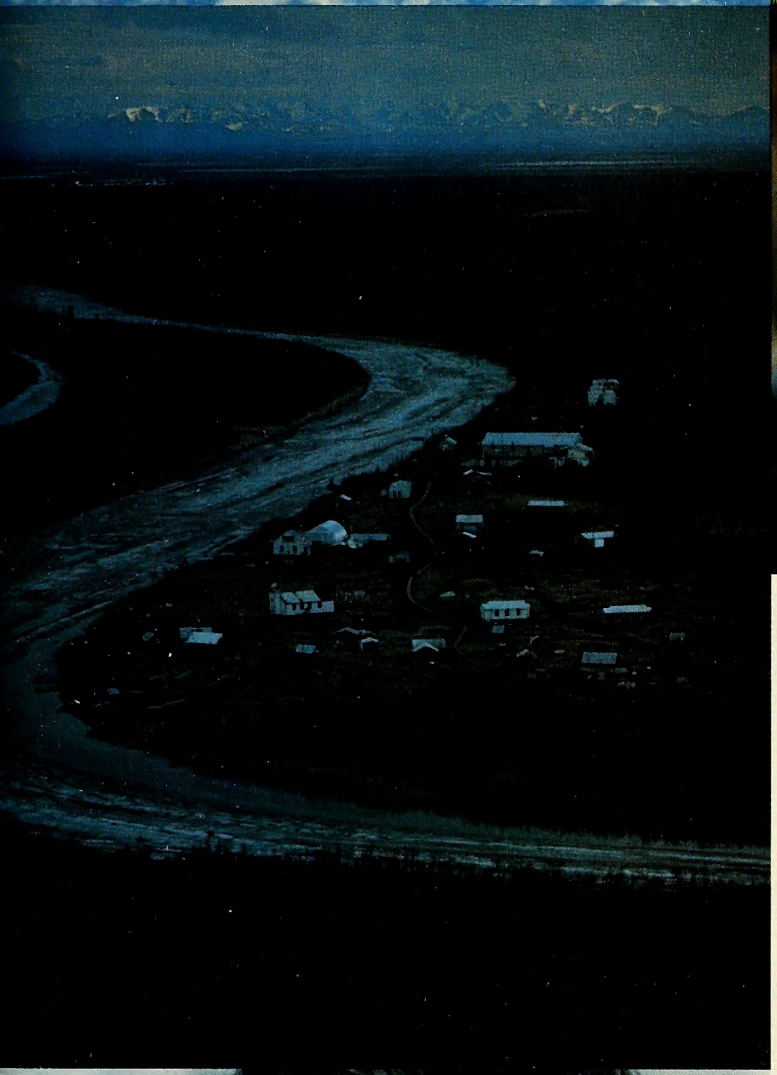
TRS-80 MODEL 1 32K	012-0100	$29.95
TRS-80 MODEL 3 48K	012-0100	$29.95
Apple 2 Applesoft in ROM	042-0100	$29.95
ATARI 40K	052-0100	$34.95

Adventure INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC.
BOX 3435, LONGWOOD, FL 32750
(305) 862-6917
ORDER FROM YOUR FAVORITE DEALER
or CALL TOLL FREE (800) 327-7172
SHIPPING & HANDLING ARE EXTRA, PRICES SUBJECT TO CHANGE WITHOUT NOTICE

Hands Across the Ice







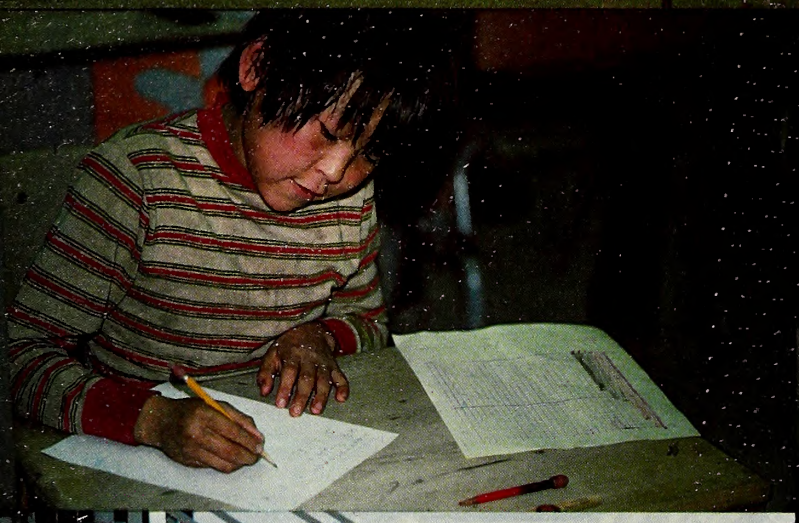
Eskimos and Their Apples

BY MELISSA MILICH

All photos from the collection of Jeff Whittaker, Education Specialist, Bethel School District. Whittaker has been shooting pictures of Alaska for the past ten years.

Among the citizens of Kotzebue, there are Eskimos who still hunt polar bear, who chew sour seal skin, and many who wear muklucks. And there are many who use computers.

As children, Americans envision our two groups of natives in a rather fairy tale way. Quite soon in our growing up, we learn that few American Indians inhabit teepees anymore; but many of us continue to envision our other native Americans, the Eskimos, as living in igloos.



In fact, igloos always were merely temporary dwellings (as you might logically deduce), used by hunters on the trail. An occasional winter wilderness traveller might still put up an igloo; but a stateside tourist would be hard-pressed to find one, even in the remote Eskimo town of Kotzebue.

Several hundred Apples are scattered throughout the schools of Alaska, some in the most remote Eskimo villages, providing one of the more sophisticated educational systems in the country for computer literacy. The parka-clad children who attend know how to set rabbit traps as well as they know how to reset a computer.

And soon a networking system called Project Walrus will link the isolated villages electronically, allowing the adult Eskimos to hold hunting conferences and other information exchanges with neighbors separated from them by miles and miles of snow.

The only thing that's holding up the project is telephone installation; some of the villages don't have such facilities yet.

Not long ago a cheechako (Alaskan for "newcomer") asked Donovan J. Rinker, director of telecommunication of the Northwest Arctic School District, if he didn't think it was slightly unusual that Eskimos were using computers.

Rinker looked as if he had just been asked the stupidest question he had heard all day, but he replied politely, "Most Americans don't know too much about Alaska."

But computers? After all, some parts of Alaska are definitely *rural*, if not primitive. Some Eskimos still haul their water from the river. Some children above the Arctic Circle

Introducing The Grappler™:

The only interface that makes computer graphics easy as Apple pie.

ACTUAL APPLE II™ PRINTOUT USING GRAPPLER™ AND EPSON MX-100

Orange Micro offers the only universal parallel interface card that simplifies high resolution graphics for Apple™ computers. No longer does the user need to load clumsy software routines to dump screen graphics—it's all done by the Grappler's exclusive E-PROM chip. There are versions to accommodate the Anadex, Epson MX-100, MX-80* and MX-70, IDS Paper Tigers, Centronics 739, NEC 8023 and C. Itoh Prowriter, and future graphics printers. The Grappler™ accepts 18 simple software commands accessible through the keyboard or user program, making it the most intelligent Apple Interface available. Order The Grappler™ through Orange Micro direct, or authorized dealers.

*Requires Graftrax-80

FEATURES—User Benefits

GRAPHICS SCREEN DUMP—Choice of Hi-Resolution Graphics page 1 or page 2.
INVERSE GRAPHICS—Provides reverse graphics of black-on-white or white-on-black.
EMPHASIZED GRAPHICS—Allows high density graphics on certain printers.
DOUBLE SIZE PICTURE—Doubles the graphic screen representation vertically & horizontally.
90° *ROTATION*—Rotates the screen picture 90°.
CENTER GRAPHICS—Accomplished through setting left margin thereby centering the graph.
CHART RECORDER MODE—Successive horizontal pictures are combined continuously simulating a chart recorder.
BLOCK GRAPHICS—For printers with block graphics (e.g. Epson MX-80, Okidata M80), the high order bit can be controlled.
BELL—For printers with a bell, bell characters are deleted during user program listings.
MARGINS—Set left and right margins.
SKIP-OVER-PERF—Set page length; printer will automatically skip 6 lines between each page.
VARIABLE LINE LENGTH—For user program listings, sets line length and wraps around with breakpoint at nearest blank.
TEXT SCREEN DUMP—The text from a user report or page of program listing can be dumped directly from the screen.
WORKS WITH PASCAL & CPM

Dealer inquiries welcome to:

Waybern Corporation
(714) 554-4520

CompuCable Corporation
(714) 635-7330

Kal Tronics
(312) 291-1220

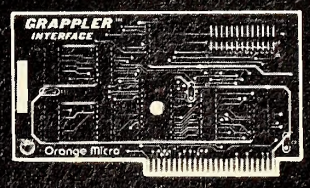
Orange Micro inc.

3150 E. La Palma, Suite G, Anaheim, CA 92806

(800) 854-8275 TOLL FREE

(714) 630-3322 CA, AK, HI

The Grappler™



have never seen a tree, let alone a supermarket, but these same kids know all about a computer.

It makes no sense and it makes perfect sense.

Remember, Alaska is our biggest state. Put the three next largest states together—Texas, California, and Montana—and you have a territory approaching the magnitude of Alaska. Approximately four hundred sixty thousand Alaskans—Caucasian, Eskimos, and Indians, make up the population of the entire state—about as many people as live in Cincinnati. Anchorage and Fairbanks tend to hustle and bustle just like many other American cities, but the villages in the north, within shivering distance of the Arctic Circle, tend to become isolated during the relentless winter. Then the only way out, to other civilizations or neighboring villages, is travel by airplane, snowmobile, or dog sled.

The Communitree, a bulletin board system for access by modem built on a branching tree structure, was Red Boucher's way of breaking the ice. H. A. "Red" Boucher, former lieutenant governor of the state and now head of his own consulting firm, is setting up the networking system throughout Alaska. The Communitree is a communication by modem device that allows users to converse freely back and forth; the system is set up to allow conferences to extend from the central discussion much in the form of a wildly growing tree. Boucher has already set up a Communitree in his Anchorage home, from which he talks about Alaska with other people all over the world.

"It taught me how little we really know about each other," says Boucher, who made the Communitree a present to the isolated villages. "So little is known about Alaska. So little is known about the first inhabitants of America!" And then he relates a lesson on Eskimo history with an eloquent pride reserved for cheechakos.

"Forty thousand years before the birth of Christ, man crossed the Bering sea land-bridge to the new world." He pauses and a picture of early men and women comes alive, the first Americans crunching through the snow from Mongolia to what, when later discovered by Russian explorer Vitus Bering in 1741, became known as Walrussia. Later it was bought by the United States for $7,200,000, approximately two cents an acre. Settlers said it was a mystery—not why Russia sold it, but why America wanted it.

"Here we still have the last vestiges of that first frontier. People still hunt and fish as a way of life; a subsistence lifestyle," declares Boucher with finality.

Eskimos learned to live in the snow and use everything the land had to offer. And maybe that's why they didn't turn their backs on computers when the machine first arrived in the villages.

But the Americanization of the Eskimo culture has many people worried, and one of the most concerned is Boucher.

"Rich heritages and languages are being lost," he says. Paradoxically, the introduction of the computer network may help preserve some of the endangered cultural elements.

"The Communitree will give young children and old people the chance to communicate between villages. I believe there will be a lot of conferencing during whaling and caribou season, and I see no reason why Eskimo children can't learn to program in their native Inuit language."

Boucher has had his own Apple system since June 1980 and started his own user's group which he calls the Sourdough Network. Again, a bit of history. During the gold rush of the 1800s, the prospectors kept a bit of sourdough bread with them to use as starter for another batch. These hardy, self-sufficient pioneers of Alaska were nicknamed sourdoughs. Today, to qualify for sourdough status, a cheechako must wrestle a Kodiak bear, climb Mount McKinley, and spit in the Yukon. To become one of Boucher's sourdoughs is not as difficult.

Boucher met one of his sourdoughs, Jeff Whittaker, while looking at software in a computer store in Anchorage. Whittaker, a teacher, decided his students could use a Communitree in their village.

"It's reasonable that I should run into Red even though we live four hundred miles away from each other," explains Whittaker, an education specialist with the Bethel School District in western Alaska. "Socially it seems as if this is a very small state. People who are active in computers are going to cross paths eventually."

A sourdough in more ways than one, Whittaker has been teaching in the Bureau of Indian Affairs school system for the last ten years and has lived in three different villages. He was trained at the University of Oregon, but he had to learn a lot more when he came to Alaska.

"There's a bilingual and a bicultural hurdle to cross; some of these villages have been extremely isolated. The kids have not seen cities, nor even trees—the things people in the more southern forty-nine states take for granted."

"Some years ago, I actually overheard a mother tell her child 'horses are not real; do not believe them.' "

The B.I.A. schools are small. Usually three or four teachers handle several ages, kindergarten through eighth grade, and they face a wide range of instructional problems. But in this school system there is generally one Apple for every thirty students. The computers provide a wide range of instruction from individual drill and practice to group teaching of computer awareness and literacy. The majority of Whittaker's students are not native speakers of English, but they're becoming fluent in computer languages.

Good students are also rewarded with time on the computer for playing games. Their current favorites are *Raster Blaster*, *Olympic Decathlon*, and *Apple Panic*.

"A child anywhere is going to enjoy the entertaining nature of a computer. But it takes a little longer for adults. It's still new and confusing to the parents."

Whittaker feels as if he's at the meeting of two cultures, but he's not trying to force his culture upon the Eskimos.

"Weather and distance have been barriers for the children here to see other villages. The Communitree will provide a way for children to communicate with each other, make friends in other villages, and discover commonalities. They'll be, in a sense, electronic pen pals."

On request, Whittaker attempted to describe the everyday life of a child from an Eskimo village; he made clear he doesn't like generalizing, but offered this view from ten years experience working with the children.

Imagine a football field with fifty houses on it. Houses are small, but families are large, sometimes eight people in a two-room or three-room house. As a rule, there are no roads, just paths. The entire layout may appear helter-skelter, but the houses have been built in practical places, always close to the water. In the smaller villages, people carry water to their houses, usually from the well or from the river.

"I am generalizing," reminds Whittaker.

Ice skating is everywhere, and although older boys might spend their after-school time checking their rabbit snares, most of the children engage in group games.

"These children are not as competitive about games as most of us are accustomed to seeing. They play for the sake of playing, rather than winning.

"Eskimo children are very good at playing and enjoying life. They're experts at being children. Children in other American cultures tend to be suppressed by adults. That hasn't happened to the Eskimos."

When all the phone lines are in—not even all the schools had telephones—the Communitree will double as a means for teachers to exchange ideas. "We're always looking for ways to find shortcuts to administrative needs," says Whittaker.

Whittaker's district has dubbed their Communitree the Walrus network, an apt acronym that stands for Western ALaska RUral Schools. The villages initially participating are Alakanuk, Chevak, Kasigluk, Kipnuk, Kwethluk, Nunapitchuk, and Tununak. Next fall, Whittaker expects to see the people of Chefornak, Mekoryuk, Napaskiak, Newtok, Scammon Bay, Toksook Bay, Tuluksak, and Tuntutuliak also talking to the Walrus.

But will it work? "Well, we're going to find out," spouted

SVA
MAKING APPLES GROW

256K APP-L-CACHE

TRIPLE DISK SPEED — QUINTUPLE MEMORY SIZE!!

Triple your data retrieval speed!

Imagine the speed of a "solid-state disk" without access delays or moving parts. Fast and reliable.

Quintuple your memory size!

Imagine the power of 304K memory — 48K more than IBM's 256K new personal computer. Big and responsive.

The SVA 256K APP-L-CACHE gives you two highspeed solid state mini disks and all the functions of a 16K memory card conveniently packed inside your Apple!

Consider the capabilities and exponsion potential of the SVA APP-L-CACHE before buying another mini disk or a memory card.

With the APP-L-CACHE you get big system performance at a micro price.

ADVANTAGES

Improve system operation:
- Add 64K to 256K bytes of memory to your Apple
- Gain up to 2 "solid-state mini disk drives" through EPROM emulation of standard DOS disk commands
- Enjoy all the functions of a standard 16K memory card
- Eliminate need to purchase an additional mini disk drive

Improve software performance:
- Enhance high speed data access and file sorting
- Compute on large data bases without mini disk delays
- Reduce disk swapping and handling
- Gain hard disk performance and the simplicity of mini floppy operation.
- Improve program development speed and lower costs
- Copy and verify disks with only one mini drive

SOFTWARE SUPPORT

You have complete compatibility with CP/M*, PASCAL, and the APPLE DOS* operating systems.

Pascal users gain
- Automatic load of all system files to APP-L-CACHE
- Double compilation speed
- Reduced Editor and Filer load time by 3 times
- Single mini disk drive operation

Apple DOS users gain
- Automatic initialization
- Faster program loading and file accessing

CP/M users gain
- Automatic initialization
- User selectable volume number

SVA

SORRENTO VALLEY ASSOCIATES, INC.
11722 Sorrento Valley Road
San Diego, CA 92121-1084
(714) 452-0101

*CP/M Trademark Digital Research, APPLE DOS Trademark of Apple Computer.

Red Boucher. "We're not going to push, we're going to put it there. We want this to grow naturally, as a way to bring in education from other areas.

"The kids will help the old-timers. There's no reason why they can't program in their native tongue."

According to Boucher, the micro allows its users to become information participants in an otherwise sedentary world. "As a nation, we're used to sitting as spectators, watching electronic junk spit out at us from a tube. Out of the twentieth century comes *Charlie's Angels*. The children of the Alaskan villages sit there in wonder, trying to figure out how that all relates to them.

"I believe this network will allow both children and adults to carry on a cultural exchange between villages, to learn from each other."

Donovan J. Rinker of the Northwest Arctic School District is also setting up a Communitree system for eleven surrounding villages. Life in Kotzebue, a town that lies several miles within the Arctic Circle, seems a universe away from Rinker's former life in Los Angeles, although he left it barely two years ago. Outside his Kotzebue office, the winter rages with grey and gloomy skies and a wind chill factor that brings the temperature down to nearly forty degrees below zero. But it's not the weather he notices, it's the absence of freeways.

"The most elaborate car in Kotzebue is a late model Chevy station wagon, and they use it as a cab," reflects Rinker, who used to spend an hour and forty-five minutes commuting to work in southern California—and that on a good day.

"Now it takes me five minutes to walk to work, thirty seconds if I take my motorcycle," he gloats. "You couldn't blow me out of here with a stick of dynamite."

Life in Alaska is a well-kept secret, according to Rinker. "Most people don't think about Alaska in any other terms except that it's cold.

"But of course it's cold. Right now the sun rises about nine-thirty in the morning and sets at two-thirty in the afternoon. We're losing about eighteen minutes of sunlight a day. But it's real nice.

"People are marvelously friendly. It's like everybody's related. The Eskimos don't take anything from nature they can't give back. I wish we could all share their philosophy of life."

There's presently a concerted effort to bring the Eskimo culture back into the schools, evidenced by a new requirement for all the children to learn Yu'pik, the native language. In addition, some of the courses, such as Eskimo history, which seem best expressed in the native language, are being taught in Yu'pik.

"We had an historical situation here where the first purveyors of Christianity told the Eskimos to speak English, that it was heathen for them to speak their native language. The folly of that has been realized, and now we're getting the language back into the schools."

Computer-based learning is "an obvious plus" for all Alaska, notes Rinker. "It makes sense because of the distance between schools. In the northwest Arctic district, there are eleven schools spread throughout 36,000 acres."

Rinker takes great pleasure in destroying the myth about Alaska being a backward state, and he won't let you forget that you inquired about igloos. "Kotzebue won an all-American city award in 1979," he says proudly. "We have a variety of housing, and some of our apartments are as modern as those in L.A. The only difference here is that we don't have swimming pools or hot tubs."

Even though there are no igloos, Rinker agrees that there is a lot of Eskimo heritage and culture that needs to be saved from modernity. And, like Boucher, he believes the Communitrees will help. People who talk together learn things from each other, and the concept of electronic pen pals isn't foolish to Rinker.

"When you're talking about a computer system, anything and everything makes sense."

One of Boucher's most far-out yet sensible suggestions is the electronic potlatch. "What?! Don't tell me you don't know what a potlatch is?" exclaimed Boucher. Again the old sourdough geared himself up for a history lesson.

Potlatches are still very much a part of Eskimo life, although their roots in the culture go back almost as far as the Bering Sea land-bridge. Boucher describes the potlatch as an economic, social, and cultural get-together between villages to pay off all debts and obligations, exchange hunting information, and barter walrus hides in exchange for clothes and food. There is always a big feast where muktuk (whale blubber) is served with Eskimo ice cream for desert. Notes the historian, "You haven't lived until you've tried Eskimo ice cream."

"Cultures tend to preserve themselves by passing on information. No matter what happens to the rest of the country, the people want to go on living the same way as their ancestors."

Historically, potlatches have been determined by the hunting season or by the weather. Boucher says the Communitree can provide an ongoing cultural exchange by allowing people in various villages an electronic potlatch to communicate about their concerns and their needs. The only thing missing will be the ice cream.

So Eskimos will soon have the Communitree to learn about each other, and outsiders anywhere in the world can now learn about Alaska by calling up the sourdough network through their modems. And if microcomputer users make the electronic trip to Alaska, Boucher will send them official certificates naming them as Honorary Sourdoughs. The last paragraph of the Honorary Sourdough certificate allows its bearer to "swap stories about Alaska with anyone who will listen."

Americans are a lot alike, and yet they're different.

In Alaska computer networks are breaking down some barriers. Shy Eskimo children will soon be able to make friends in neighboring villages. The rest of the world can call the sourdough network. Red Boucher will be there, and he'll tell stories to anyone who'll listen. ■

THE NEXT GENERATION OF COMPUTER GAMES IS HERE

FROM THE MAKERS OF THE FIRST ANIMATED ADVENTURE GAMES

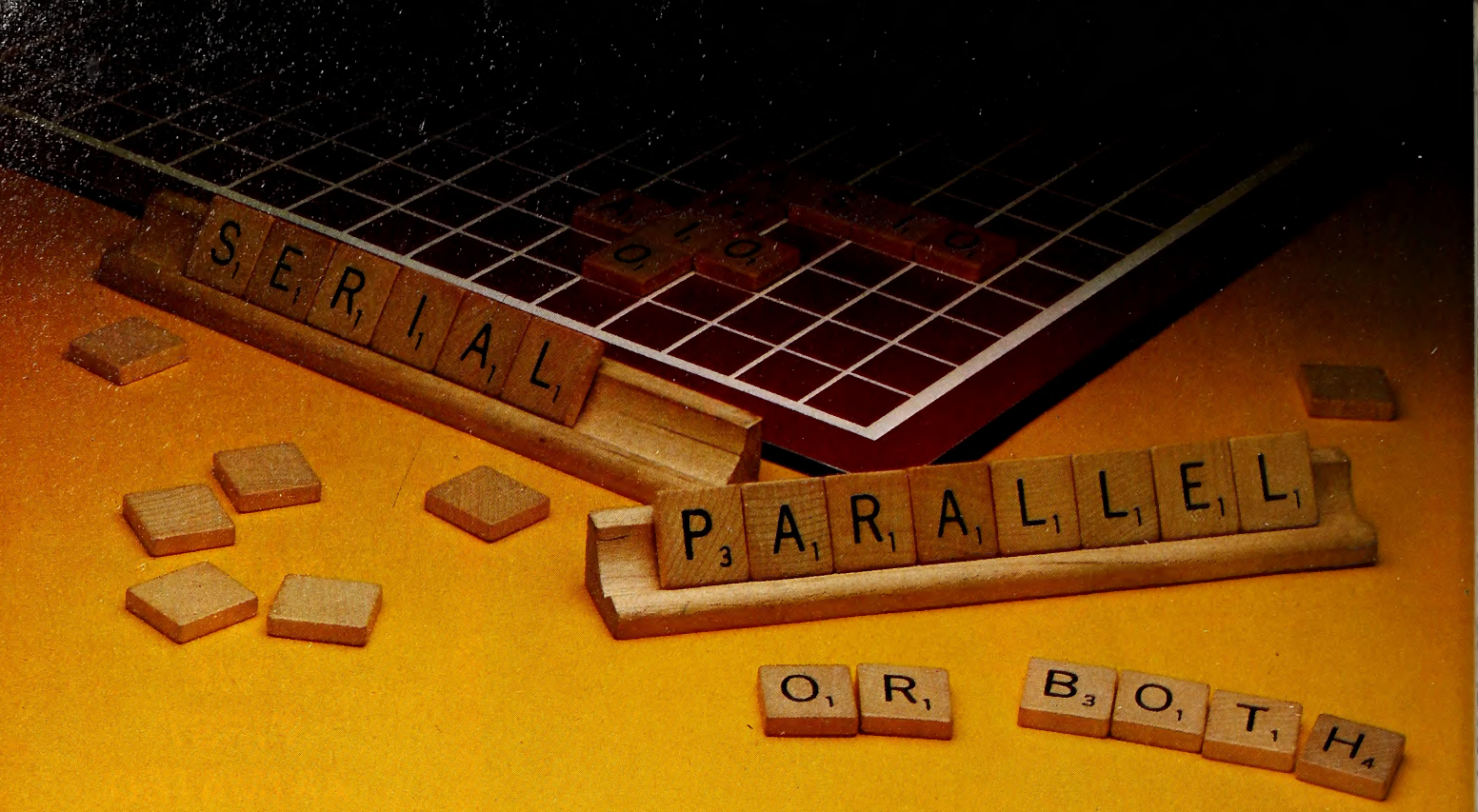
Synergistic Software

- **APVENTURE TO ATLANTIS** is a new fast action apventure game, by Robert Clardy, continuing the Odyssey Apventure series. 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.
- Computer adventuring has been good, Synergistic Software has just made it better. Can you imagine the most exciting adventure game ever composed with more color and sound, highly detailed animation, arcade like action, true role playing, greater challenge, unending variety and endless enjoyment? That's what the Next Generation of computer games is all about. This professionally designed adventure was specifically written to take full advantage of all the Apple's capabilities. Requires careful planning and strategy, quick thinking and reflexes, as well as a 48K Apple II* with DOS 3.3 and paddles.

Only $40.00

*Apple II is a trademark of Apple Computer Inc.

From
Synergistic Software
5221 120TH AVE. S.E.
BELLEVUE, WA 98006
(206) 226-3216



The New AIO-II.

Another winner from the folks who invented multi-function Apple interfaces.

SSM is still ahead of the game when it comes to the Apple II™ Choose from any of four operating modes. And you're assured of flexibility at a price that spells big savings.

Check out why SSM spells success:

THE NEW SERIAL/PARALLEL AIO-II: this new full-function serial and parallel interface for the Apple II gives you true simultaneous operation. It's totally transparent to the user; there's no software to write, and no need to modify your computer. Advanced design techniques overcome the need for "phantom" slot assignments and related software compatibility concerns.

The AIO-II™ appears to application software as either an Apple communications card or parallel printer card. You don't have to worry about special setup requirements. Hassle-free, user-oriented AIO-II design permits easy selection of four functions: serial modem, serial terminal/printer, parallel Centronics printer, and general-purpose parallel. On-board firmware provides all necessary drivers in a single package. Optional cables support Centronics-compatible printers and others.

You can remain in the game, too, with either our Serial ASIO™ or Parallel APIO™ boards—if low cost, single function's your style. Our ASIO connects your Apple to a modem with an optional terminal routine, or to a terminal/printer without modifying the cable. It gives you standard RS-232 interface with complete handshaking. And selectable rates from 110 to 9600 baud. The APIO parallel interface supports general-purpose I/O and all popular Centronics-compatible printers, including Epson, IDS, Okidata, Anadex, and others.

All our game-winning boards are designed to make your Apple more productive. Optional software support is available for Apple Pascal and Microsoft CP/M™

Whether you're an OEM or end user, we can deliver any quantity you specify. Fast. For the latest word on any of these interface boards, or our newly enhanced A488™ (the industry's highest performance IEEE-488 controller), see your local dealer, or call us today.

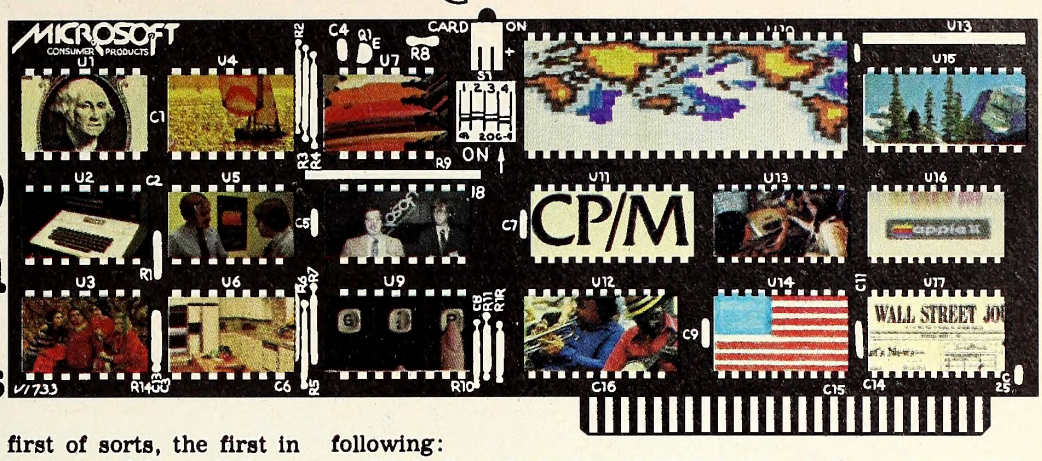
SSM Microcomputer Products, Inc., 2190 Paragon Drive, San Jose, CA 95131, (408) 946-7400, Telex: 171171, TWX: 910-338-2077.

SSM

The Transformation People

SOFTCARD Symposium

by Greg Tibbetts



This month's column will be a first of sorts, the first in which we'll have a chance to respond to some of the mail that's arrived from readers. Because of the time lag between the writing of a column and its appearance in print, specific questions from specific readers should long since have been answered. Nevertheless, we hope that general comments in response to such questions will be of interest to many readers.

Before we get to the mail, however, our main topic of this month—and probably for several months to come—will be the CP/M utilities. Since, we hope, this column is being read by Apple owners without CP/M as well as those already familiar with CP/M, we'll get as basic as possible in our descriptions of the programs. Also, we'll try to take them in the order of most confusing to least confusing, based on general comments of users. With that in mind, *PIP* will be our first target.

PIP stands for *Peripheral Interchange Program* and is one of the transient commands available on all CP/M systems. Apple users traditionally think of the term peripheral as meaning a discrete device connected to the Apple via an interface card of some type. To Digital Research and hence to CP/M, the term is used to describe any module, hardware or software, physical or logical, that interfaces with the operating system. In this light, the keyboard, video display, card reader, a disk file, and the printer are all examples of peripheral devices.

Now the name *Peripheral Interchange Program* begins to make sense, since this is the means by which these peripheral devices exchange data amongst themselves. In a later column covering STAT.COM, we'll have more to say about the structure of devices and the ways they can be arranged or assigned. For now we'll deal with them in their default structure.

Since *PIP* is a transient command, it exists as a COM file on the disk and is invoked simply by typing its name, like so:

PIP

PIP always requires additional input to function. But, like most other Digital Research utilities, if it is invoked by typing only its name, it will prompt the user for additional input with an asterisk. If the user is prepared both to invoke *PIP* and at the same time to give it the input it requires, then it can be invoked as follows:

PIP target=source [opt]

Since, as its name implies, *PIP* is a program that allows the interchange of data, there must always be both a source from which the information is taken and a target for the information to be sent to. There may be more than one source device specified, in which case *PIP* will concatenate the multiple devices and/or files, reading from left to right.

As with all CP/M utilities, the positions of the target and source devices with respect to the equals sign can be remembered by analogy with a Basic assignment statement; the object to the right of the equals sign is placed into the object on the left, as in A=10 or B$="hello".

In addition to the target and source devices, the user may also select one or more special options—indicated by the [opt] in our example—that direct *PIP* to give special handling to particular data.

Target and source devices may be disk files or any of the following:

CON: Console device; if CON: is target, then output will be to the screen. If CON: is source, then input will be from the keyboard.

TTY: UC1:, AND CRT: Unless specially defined by the user, these devices will have the same effect for input and output as CON:, whether they're used as targets or as sources.

RDR: General-purpose reader device. This is typically set up to handle input from an input-only device or from the input portion of a bidirectional device. Unless the proper hardware and software interface has been done, use of RDR: with *PIP* will be undefined. Since this is an input-only device; it can only be used as a source, not as a target.

PTR: UR1, and UR2: These devices are to RDR: what TTY:, UC1:, and CRT: are to CON:, and the results of using them will be same as with RDR: unless special definitions have been made.

PUN: General-purpose output device. This is typically set up to handle output from an output-only device or from the output portion of a bidirectional device. Unless the proper hardware and software interface has been done, use of PUN: with

An exciting new game by the master of Apple animation: NASIR.

**PIGGO, the fearless firefighter,
battles blazes left in the fiery wake of the incredible
FIREBIRD.**

Presented by

GEBELLI SOFTWARE INC.

PIP will be undefined. Since this is an output-only device, it can only be used as a target.

PTP:, UP1:, and UP2:. These devices are to PUN: what TTY:, UC1:, and CRT: are to CON:. The results of using them will be the same as with PUN: unless special definitions have been made.

LST: The system list device, usually a printer. It may also be assigned to the console, via the TTY: device, if the console is used as a teletype producing hard-copy output. Obviously this is an output-only device and therefore can only be a target, not a source.

LPT: and UL1: These devices are to LST: what TTY:, UC1:, and CRT: are to CON:; the results of using them will be identical to those of using LST: unless special definitions have been made.

In addition to the names we've listed so far, the following special names can also be used:

PRN: This special "device" is the same as LST:, except that tabs are expanded at every eighth character position, all lines are numbered, and page ejects (form feeds) are inserted every sixty lines with an initial eject before printing starts. PRN: can be used to create formatted hard copy of files created by ED.COM, for example.

NUL: This special "device" is used with paper tape punches as a way of providing necessary leader and trailer on the tape. Its effect, when encountered, is to send forty null characters (ASCII zeros) to the target device.

EOF: This special "device" is used to insert the CP/M end-of-file character (always an ASCII control-Z) at the end of a transfer. This isn't often used, since all ASCII files transferred via *PIP* get an automatic end-of-file character. Non-ASCII files do not require such a character, since the true end of file—the total number of records specified in the file's directory entry—is used.

INP: and OUT: These special "devices" are only usable if *PIP* itself has been altered by DDT to patch in special driver routines to interface to some nonstandard peripheral device. It is possible, for example, to patch in a routine for getting and sending data from and to cassette with these routines, but this requires knowledge of assembly language programming, as well as of the Apple cassette-handling routines and hardware. See the CP/M interface guide portion of the manual for more information on this facility. If enough interest is shown in this subject, it could be made the focus of a separate column.

Now that we have examined the possible target and source devices, we'll look at the special options that can be invoked. Such options must be enclosed in square brackets—achieved by control-K and shift-M on Apples with unmodified keyboards.

[B]. This option tells *PIP* that the source device will be sending the ASCII x-off character, a control-S, on a regular basis. It also directs *PIP* that when it sees this character it should pause input and write out to disk all the data so far received and stored temporarily in memory, then return for more input. This option is most useful when you're reading from a device that sends a continuous stream of information that would exceed the memory available for buffering it, but which can be programmed to send the x-off periodically and also pause its output when directed.

[Dn]. Truncate all lines transferred at column *n*. This is supposed to allow the transfer of text with long lines to a narrow device. However, it's difficult to imagine a situation where data cut off on the right would not be missed.

[E]. Echo all transfer options to the console as they are performed. This has the effect of typing out the file to the console device during transfer and consequently will slow all but printer transfers.

[F]. Remove any form feeds encountered in the source during transfer.

[Gn]. This option, for use only with disk files, tells *PIP* that the source file resides in USER No. *n*, which must be different from the current user area. Also, you should be aware that wild-card file names like *.* and *.BAS cannot be used with [G].

[H]. This tells *PIP* that the source file is an Intel-format hex

file. Such files follow a very specific format, as can be seen by assembling DUMP.ASM from your master disk and then printing or typing the DUMP.HEX file created. *PIP* will do some editing of the file and will allow retries and corrective action if warranted.

[I]. This option also applies to Intel-format hex files and tells *PIP* to ignore the :00 records in the file. This option automatically sets the H option.

[L]. This option translates all upper-case alphabetic characters to lower case during transfer.

[N]. With [N] specified, each line transferred will have a line number added. The lines will start at one and increment by one with leading zeros suppressed and a colon following the number. You can optionally specify [N2], which has the effect of including leading zeros in the line numbers and inserting a tab character following the number.

[O]. This tells *PIP* that the file to be transferred is an object (non-ASCII) file and therefore that any control-Zs encountered do not signify the end of file as they do in ASCII files. If [O] is selected, *PIP* will use the absolute end of file as indicated in the file's directory entry to tell when to stop transfer. It should be noted *PIP* automatically assumes all COM files to be object files.

[Pn]. Insert a page eject (form feed) every *n* lines. If no number is specified, or if an *n* of one is used, *PIP* will default to sixty. If the [F] option is used first, any existing form feeds will be eliminated in favor of the ones specified with [P].

[Qs control-Z]. This tells *PIP* to stop the transfer when string *s* is encountered. You must terminate the string in the command with a control-Z. See also the [S] option.

[R]. This option informs *PIP* that the file used as source is a system file. Without this option, *PIP* will not recognize system files.

[Ss control-Z]. This tells *PIP* to start transfer when string *s* is encountered. You must terminate the string in the command with a control-Z.

[Tn]. This option tells *PIP* to expand any tab characters that it encounters to every *n*th column during transfer. This is different from simply replacing tab characters with *n* spaces. You should be aware of that difference.

[U]. With this option, *PIP* is directed to translate all lower-case alphabetic characters to upper case during the transfer.

[V]. This option tells *PIP* to perform a verify operation on every disk write that takes place. With the [V] option in effect, the information written to disk is read back and compared against what is in memory before new input is taken from the source. The target in this case must be a disk file. It is a good habit to use the [V] option whenever you transfer data; the decrease in speed is offset by increased reliability.

[W]. With [W] in effect, a target disk file that has been set to read-only (R/O) will be overwritten without *PIP*'s asking for permission. Without [W], *PIP* will not overwrite until it interrogates the console for the user's approval.

[Z]. This option tells *PIP* to zero the parity bit on all incoming ASCII characters during transfer.

Now that we've gotten all the descriptions out of the way, we can talk about *PIP*'s uses. The majority of users treat *PIP* as no more than a simple disk file transfer program. In many cases, for a particular user, this may be its only valid application. However, as can be seen by a study of the material in this column, *PIP* is really much more sophisticated than a simple file transfer program. Ignoring the abilities that apply to such special peripherals as tape readers and the like, it can be used as everything from a text file print formatter to a gross form of editor.

As a disk or file copy program, *PIP* is somewhat slower than *COPY*, but it gives increased control over the files to be transferred. The availability of wild-card file names or ambiguous file names (as the manual refers to them) makes it easy to transfer selected files from a source disk to a formatted target disk.

For example, PIP B:=A:*.BAS will transfer all files with the extension .BAS to the target disk (B: in this case). PIP B:=A:*.* will transfer all files, regardless of name or extension, to the target disk, PIP B:=A:G*.* will transfer all files with names beginning with G and any extension to the target disk. And so on.

This wild-card facility makes incremental backups on often updated files as easy as setting up a system to have all active files share a common name or extension—NEW, for example. A PIP B:=A:*.NEW, and the backup is done. You can also create a SUBMIT file that will automate this process even further.

Copying individual files from disk to disk is only one use of *PIP* in this fashion. Files may also be renamed during the copy process, by specifying a new name on the target side. Even duplicate copies of the same file may be made on the same disk, simply by specifying a different name for the copy. *PIP* will default to the currently logged drive if there's no drive letter specified.

In addition to copying files, *PIP* will also edit files during transfer or duplication. By using the appropriate options, you can have page ejects inserted or ignored, line numbers added, upper-case and lower-case conversion made, and so on. You can concatenate files, by specifying more than one source file name, or you can have only selected data extracted from a file, by using the S for Start-at and Q for Quit-at options. Since options apply only to the file name or device to which they're immediately appended, it is possible to combine several options and several files without conflict.

Finally, anyone who has done much with large files using ED.COM will attest to the difficulty of editing such files when ED line numbers don't show on printed output. By using *PIP* with the target as PRN:, you can get a hard-copy printout with line numbers that follow the default numbering of ED. Such copy will also have all tabs expanded and page ejects inserted for improved readability.

Although programs exist that have many of the same special purpose printing and data-capturing facilities (from CP/M bulletin boards, for example) many users prefer to use *PIP* because it is easy and because one program will do the work of several individual ones. By far the easiest way to learn the use

The Brøderbund

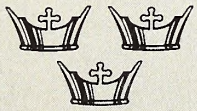
Arcade Machine

by Chris Jochumson and Doug Carlston

☆ ☆ ☆CREATE YOUR OWN ARCADE GAMES☆ ☆ ☆
NO KNOWLEDGE OF PROGRAMMING NECESSARY

- Each time you use the Brøderbund Arcade Machine, you will generate a new, copy-protected disk containing on it an arcade game of your own design!
- Your games will have the following features:
  - Animated full-color monsters and other players as designed and drawn by you! (or select from a medley of creatures on the Arcade Machine disk)
  - Dramatic shoot-'em-up explosions and sound effects
  - Automatic scoring and high score features
  - Fast-loading auto-boot to title page
  - Colorful title page with your title and name in large graphic letters (like other Brøderbund games)
  - One or Two player options
  - Several skill levels
  - Optional gravity effects
  - Menu driven for easy use, regardless of age or ability!
- Your arcade programs are created entirely in machine language for fast and smooth arcade-quality animation.
- $44.95 Requires Apple II or Apple II Plus with 48K and one disk drive.

Ask your dealer or send check or money order to:



Brøderbund Software

2 Vista Wood Way, San Rafael, CA 94901 (415) 456-6424

and versatility of *PIP* is simply to practice with it. Try out the various options, devices, and formats. You'll probably find—as many others have in the past—at least one more very substantial use for this program.

Before we leave the subject of *PIP*, there are a couple of miscellaneous things that should be mentioned.

Concatenated source files must be separated by commas, and options must be enclosed in square brackets with no other punctuation between the source file and the option. Also, options apply only to the source file to which they're appended. *PIP*'s default state assumes that source files are true ASCII files ending with a control-Z; COM files, however, are assumed to be object files. The total length of a single command line must not exceed 255 characters, and all characters in a *PIP* command line, if given at the same time *PIP* is invoked (that is, on the same line), will be translated to upper case. This last fact will seldom cause a problem, since CP/M keeps its file names in upper case. However, if you are using the S and Q options, the strings you are searching for also will be transferred to upper case. To avoid this unwanted conversion, invoke *PIP* by typing only its name, and then enter your command line at the asterisk prompt. *PIP* will not translate the characters in the strings to upper case if you do it this way.

When *PIP* copies or duplicates a file that already exists, it creates a temporary file on the disk with the appropriate filename and an extension of .$$$. The original file is only deleted and the temporary file renamed to the actual name if the transfer operation is successful. Otherwise *PIP* will indicate an error and terminate, leaving the original file intact; the temporary filename, in that case, will remain in the directory, but the temporary file will be empty. Such temporary files should be deleted as soon as possible. Pressing any key on the keyboard will also terminate a transfer operation; *PIP* will report the message ABORTED, with the same result as an error condition.

Now—on to the mail.

Several people have written asking for a reverse of the APDOS utility—one that would go from CP/M to Apple DOS. Such a utility does exist and, in fact, is included on the disk sold as part of the Assembly Language Development System. The program is not available separately as yet, since there doesn't seem to be sufficient interest to warrant making it a product. The program is available in printed form, but unfortunately consists of several pages of hex dump. It can, however, be requested. Perhaps a group effort would reduce the drudgery of typing in all that hex dump. As is the case with all such application notes distributed, this material is considered to be in the public domain.

Several people have written to ask whether the SoftCard works with the Apple Graphics tablet. To the best of our knowledge, no driver currently exists that will allow these two products to work together. If anyone knows of such a program, we would welcome news of it and report it here.

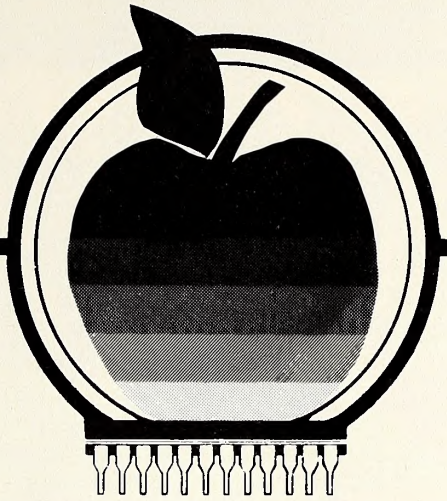
Many of you have written concerning the interfacing of machine language routines to Basic-80 programs. That subject was dealt with briefly during the series on Basic-80, and more esoteric examples will be forthcoming in future columns.

One very good question received was why the common programs *MOVCPM* and *SYSGEN* that one reads about were not included on the SoftCard diskettes. *MOVCPM* is a utility that makes it possible for the user to relocate CP/M to fit the memory size of a particular system—16K, 32K, 48K, 60K, or whatever. Since the SoftCard can only be used with Apples that are either 48K or 64K systems, only two versions of CP/M were possible—44K and 56K. The program *CPM56.COM* fulfills the function of *MOVCPM* in creating the larger size for those who need it. *SYSGEN* is a program that places the operating system on disk. That function is fulfilled with the SoftCard by using *COPY.COM* with the /S option.

We have also received several questions about why the utilities *DDT.COM* and *ASM.COM* are 8080-based programs rather than Z-80 based, since the SoftCard is a Z-80-based product. Both those programs were developed at a time when CP/M was only available for 8080-based machines, and they were standard items included with every copy of CP/M. Later, when the Z-80 chip became popular and CP/M was made available for systems using it, these programs were left as standard, since they were still very functional on the newer machines. Other assemblers and debuggers designed to take advantage of the Z-80 were introduced and sold separately for those people who wished to upgrade, but by far the widest support among users groups (in the exchange of programs and so forth) is still for 8080 code, since, no matter which system you have, 8080 code will execute properly.

Finally, the problem of available software is still very much with us. Most programs written in the CP/M world are utilitarian in nature—compilers, assemblers, disk utilities, and so on. There is, however, a wealth of public domain software out there in the CP/M users groups that can be had for the cost of a telephone call to download it. This software ranges from games to business applications, and most of it is very good. The only real requirement is the use of a modem and connection to Micronet or one of the local CP/M user group bulletin boards. I would recommend to anyone wishing to expand his CP/M library, that he attempt to get up on these boards either personally or through his local Apple club, which probably already has a Micronet account or at least the phone numbers of some existing boards. You may be pleasantly surprised at what's out there.

Until next month . . . ■



YOU GET

WORD PROCESSING POWER

AND

PRINT ... USING

FOR YOUR APPLESOFT* PROGRAMS

WITH THE

FORMAT ROM

WITH THIS POWERFUL ROM IN YOUR MOUNTAIN COMPUTER'S ROMPLUS* OR ANOROMEOA'S ROMBOARD* YOU WILL MAKE APPLESOFT* BASIC ONE STEP MORE POWERFUL THAN THE MOST AOVANCEO LANGUAGE AVAILABLE. AND - BECAUSE REAO ONLY MEMORY NEVER FORGETS, THESE NEW COMMANDS BECOME A PERMANENT PART OF APPLESOFT'S* BASIC LANGUAGE.

PRINT STATEMENT FORMATTING - Can you imagine formatting a letter with Applesoft print statements? Monumental to say the least - you would have to agree. Well - with the print statement formatting capabilities of the FORMAT ROM all you have to worry about is what you want your print statements to say and leave the rest to the FORMAT ROM. You tell the FORMAT ROM, right in your Applesoft program, how you want your letter, manuscript, etc. to look and best of all, formatting commands can be changed at any point in your program with simple basic commands. FORMAT ROM gives you the same powerful commands as a dedicated word processor costing hundreds of dollars. Here are some of the powerful formatting instructions which you will have with the format rom: **DEFINE PRINTING MARGINS - OEFINE THE LENGTH OF A PRINTED LINE - RIGHT HANO JUSTIFICATION - NO WORO WRAP AROUNO JUSTIFICATION - AUTOMATIC INOENTING OR OUTOENTING OF PARAGRAPHS - SKIPPING OF LINES AFTER EACH PRINTEO LINE** (Double space/Triple space etc.) - **PAUSE AFTER PRINTING SO MANY LINES - CENTER THE NEXT LINE OF TEXT** (FORMAT ROM FIGURES IT OUT FOR YOU) - **USER OEFINEO CHARACTER SUBSTITUTION** - FORMAT ROM will even put two spaces after each period it finds, even if you forget and only put one space.

PRINT...USING - Will format and tabulate the output of alpha/numerical data that has been predefined by you without having to go through the basic programming steps to get the format requirements. Five modes of PRINT...USING are available: Alpha mode - will tabulate and right justify strings of any length. Alpha/Numeric mode - will tabulate any predefined string to the left of any predefined numerical data which can be formatted in any of the following output modes; Floating Point or Integer with any number of digits, Fixed output with any number of decimal places with round off capabilities of positive or negative numbers, Fixed output with commas inserted every third digit from the left of the decimal point. All PRINT...USING routines will tabulate, right justify, line up all decimal points, pad the right and left side of a number with any predefined character and can be used within formulas or equations which will then format the mathematical result. Overflow messages or symbols, defined by you, can be printed if a number overflows the limits specified by you. PRINT...USING can be used in immediate or deferred and is compatible with all Applesoft basic commands.

FORMAT ROM will support all printers, 80 column boards, lower case adapters, and requires 48K, FP in ROM, DOS 3.2 or 3.3, M.C's ROMPLUS* or Andromeda's ROMboard* **$49.95**

OTHER ROMS AVAILABLE: All ROMS are compatible with MC's Romplus or Andromeda's ROM Board.

- **DUAL DDS RDMS** - Switch from one DOS (3.2 or 3.3) to the other without booting **$54.95**
- **FP RENUMBER/MERGE ROM** - Apple Computer's infamous renumber program **$39.95**
- **BASICS RDM** - Will boot standard, special, and dedicated 13 sector disks **$39.95**
- **FP EDITRDM** - Global search, change, and remove. (Works jointly with PLE) **$39.95**
- **COMMAND ROM** - Catalog Command Menu and Disk Map **$39.95**
- **DISK COPY/SPACE RDM** - Duplicates 13 or 16 Sector Disks **$39.95**
- **FP UTILITY RDM** - Auto Line Numbering, LIST CTRL, RESURRECT KILLED PROG. Alphabetize DISK CAT., Expung DOS **$39.95**

(Check or M.O.) Visa or MasterCard Accepted

SOFT CTRL SYSTEMS, BOX 599, WEST MILFORD, NJ 07480 **201-728-8750**

*REGISTERED TRADEMARK
ALL FIRMWARE IS COPYRIGHTED

The New Spellers

BY JEAN VARVEN

Word processing programs have done much to simplify the lives of microcomputer users. Now you can draft, rewrite, revise, update, and otherwise modify your written communication to your heart's content—no need to start all over again at the trusty typewriter or dog-eared legal pad. For many of us, our favorite word processing program for the Apple qualifies as a small miracle, holding the place of honor reserved previously for erasable typing paper and later for self-correcting typewriters.

A Significant Stone Unturned. But as useful as word processors are, they leave one important task undone—they do nothing to ensure that the words in that professional-looking letter or document are spelled correctly and that no typographical errors have crept in during the fever of inspiration or the flurry of revision. Not only don't word processors know whether *accommodate* has one *m* or two, they won't even alert you to simple errors—missing letters, transpositions, and the like. A good spell-checking program that's compatible with the word processor you use holds promise of filling the gap.

We looked at four spell-checking programs for the Apple: *Goodspell*, by Henry G. Brown (Synapse Computer Services, Apple Computer/Special Delivery Software); *Magic Words*, by Bill Graves and Bill Depew (Artsci); *Apple Speller* (Sensible Software); and *The Dictionary*, by Thomas G. Cain and James E. Linley (On-Line Systems). *Goodspell* bears a 1980 copyright; the other programs are quite recent releases. Each requires 48K and works best with a printer, although only *Goodspell* and *Magic Words* require one. *Goodspell* and *Magic Words* are single-disk drive programs, *Apple Speller*'s current version requires two drives, and *The Dictionary* accommodates either one or two.

To be useful, a spell-checking program must be compatible with the files your word processor generates. *Goodspell* is designed specifically to work with 3.3 *Apple Writer* files. *Magic Words* is particularly suited to files created on Artsci's *Magic Window* word processor but will also process files created on other word processors that generate standard 3.3 text or binary files. Although the prerelease version of *Magic Words* had not been endowed with *Apple Writer* compatibility, the package you'll find in your store is, according to Artsci. Sensible Software's *Apple Speller* is compatible with standard 3.3 DOS text or binary files, including those generated by *Apple Pie, Executive Secretary, Letter Perfect, Magic Window, Text Editor, Superscribe II, Word Power*, and *Write-On*, in addition to *Apple Writer*. The current version of *The Dictionary* is compatible with *Apple Pie*, *Apple Writer*, and On-Line's own *Superscribe II*.

Seeing the Trees for the Forest. Now that so many spell-checking programs are hitting the market, several questions arise. What is a spell checking program supposed to do? How do spell-checking programs work? How do the existing programs differ? Which program is right for you?

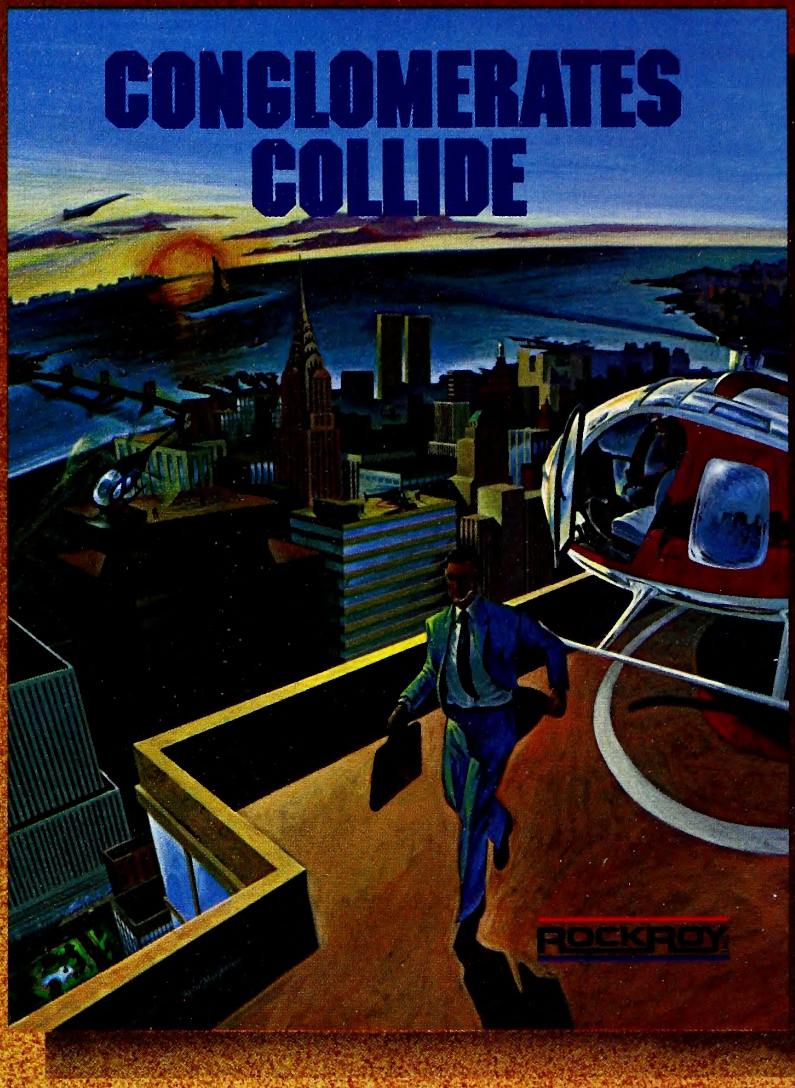
The primary stated purpose of each of the programs evaluated is to help you find and correct spelling and typographic errors in your word processor files. Each program accomplishes this by means of a dictionary or wordbook that's stored on disk and loaded into RAM (and, in some cases, into your RAM card if you have one). After the program and the dictionary have been loaded into memory, your previously generated word processor file is checked (proofread) against the dictionary. Letter combinations the program isn't familiar with are presented to you for consideration.

All four of the programs allow you to indicate that the spellings questioned are correct, ask that they either be marked or printed out as errors, or ask that they be ignored, and all allow you to generate a printed list of questionable spellings or known errors.

How and What Else. Although these programs can rightfully be called spell-checkers, they differ in how they accomplish this task and in the other features they offer.

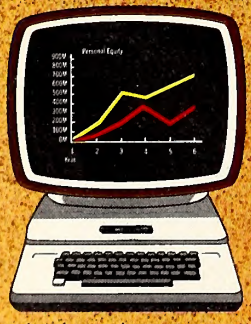
Goodspell checks any *Apple Writer* text file you specify against its dictionary and prints out a list of suspected errors. After *Goodspell* is up and running, you insert your *Apple Writer* text file disk in the drive and select one file to be checked. Your *Apple Writer* disk is catalogued, just in case you can't recall the exact file names. In attended operation, *Goodspell* will ignore any words you ask it to and will print the others out in a sixty-four-character display showing the word in its context, followed by a message showing which word wasn't found. Your word and its context will also be displayed on the screen. If your Apple has a lower-case chip, certain characters (apostrophes and periods, for example) will be shown incorrectly on the screen, but will be fine in your printout. When you choose the unattended mode, *Goodspell* will print out in context all words it does not recognize, again followed by the word-not-found indicator.

In addition to attended/unattended modes for spell-checking files, *Magic Words* gives you a printed error list showing the page and line numbers at which errors occurred. You can vary the error context length, as well as specify such things as the width of the column in which the printed-out error line is displayed. *Magic Words* also allows you to create a marked file, using whatever character you designate to indicate the occurrence of errors. This makes it easy to locate your errors



Lead your own company in a game of corporate survival to become "Chairman of the Board".

Conglomerates Collide™ as you battle Multi-National corporations for world dominance. Land atop your own corporate headquarters and prepare to acquire growth companies, handle bankers and deal in world markets to build the value of your company. Pay close attention to rapidly changing interest rates, earnings and the P/E ratio. Global news events that effect trade conditions will be reported to World HQ and challenge even the most clever of Presidents. At Decision Central you are on-line to 5 corporate data banks for ready access to vital information.



Each player's progress is instantly charted with RockRoy's exclusive color line performance graph (illustrated). Easy to learn rules and 3 different game play options: solitaire, 2-4 players, or multiple computer opponents. For high scoring "Chairmen of the Board" each game disk comes with 3 entries to the RockRoy prize competition.

For your Apple® (48K, Disk with Applesoft in ROM). $39.95 includes shipping and handling. Visit your local dealer today. To order by mail send your check to RockRoy. VISA and M/C holders order by calling toll-free 800-528-2361. 15 day money back guarantee.

ROCKROY INC.

7721 E. Gray Road
Suite 103
Scottsdale, Arizona 85260
Toll-Free 800-528-2361

later when you use your word processor to correct them. It's also possible to correct your errors as you go along and to create a new, completely corrected file on a separate disk without rebooting your word processor. *Magic Words* also allows you to add words to its dictionary during the correction process or to specify that a particular spelling should be accepted as correct but *not* added to the dictionary.

Apple Speller offers attended and unattended operations and an attractively formatted printed error list that shows your word in context. It also allows you to catalog your word processor disk if you need help remembering the name of a file. It gives you the option of creating marked files so that your errors will be easy to spot when you correct your files through your word processor. This program also allows you to add words to its dictionary during the process of reviewing your file.

The Dictionary offers attended or unattended modes and allows you to add words to a temporary dictionary during file processing. It provides the options of a printed error list or a marked file but not the option to have both. The printed error list does not give you the context of a questionable spelling; what it does do, and shouldn't, is print out the screen prompts that guide you through the disk-swapping process. This detracts somewhat from the appearance of the printed output. The presence of a RAM card will be recognized by *The Dictionary* and will speed up its spell-checking process.

Time of Decision. Which program is right for you depends on a number of factors, including what word processor you use, whether you have one or two disk drives, whether you have a RAM card you'd like to make use of, and how much money you plan to spend. Processing speed and frequency of disk swapping may also be significant considerations.

Other pertinent variables include dictionary size; dictionary add/delete capabilities; program start-up time, organization, and ease of use; error-handling; adequacy of screen prompts; and clarity of written documentation.

When Size Counts—and When It Doesn't. Dictionary sizes in the four evaluated programs are: 14,000 words in *Goodspell* and *Magic Words;* 25,000 words in *The Dictionary*; and more than 31,000 words in *Apple Speller*. Although many people simply assume that a larger dictionary means a better program, the issues of dictionary size and adequacy have been the subjects of considerable discussion.

Since spell-checking dictionaries are not context sensitive, the point can be made that a larger spell-checking dictionary may actually increase the possibility that errors will slip by undetected. The more combinations of letters a dictionary recognizes as words, the greater the probability that it will accept some combination as correct even though it doesn't spell the word you intended. If you accidentally type the word *spear* when you meant to type the word *speak*, for example, the dictionary will not alert you to your error, since your error itself is a word the dictionary recognizes.

Focusing entirely on dictionary size obscures another important concern—dictionary quality. Given that the English language has been estimated to consist of some six hundred thousand words, doesn't the usefulness of a dictionary depend in large measure on the appropriateness of the words selected?

The matter of dictionary size also brings to mind the sticky crocomputer owner an average person? *Goodspell*'s 14,000-word dictionary was designed to meet the needs of the statistically defined average person. That turned out to be less than adequate to our purposes. *Goodspell* queried a great many words we knew were spelled correctly but it didn't. This is not meant to imply that *Goodspell* is not a useful program in certain settings, only that it was not an appropriate spell-checker for our text files, which happened to be manuscripts for typical *Softalk* articles.

A more significant limitation to the *Goodspell* program than dictionary size is that its dictionary cannot be augmented or altered. Although words you ask it to ignore while it processes a particular file will store temporarily in RAM, the dictionary itself cannot be permanently updated so that it recognizes words and abbreviations you use often. This means it can't be customized to particular needs. For files such as ours, the checking process using *Goodspell* would necessarily be lengthy and repetive.

While the *Magic Words* dictionary is no larger than *Goodspell*'s it did recognize a higher percentage of the words it encountered in our files. Whether this is indicative of better word choice on the part of *Magic Words* or has more to do with the oddities of our files is difficult to ascertain. Neither the *Goodspell* or *Magic Words* dictionaries will list, so it was not possible to compare them to one another in an attempt to account for this somewhat unexpected difference.

Unlike *Goodspell*'s, the *Magic Words* dictionary is extensible. Applications dictionaries of 400 to 500 words apiece (significantly more if you have a RAM card) can be created to complement the main dictionary. Files can be checked against these applications dictionaries as required, and, if you need to add more than the alloted number of words, you just create volume two of a particular applications dictionary. If you want to add lots of words to the dictionary all at once, you can create a text or binary file listing of new words or treat an existing file as an applications dictionary.

At more than 31,000 words, *Apple Speller*'s dictionary is the largest among the four programs. It is one of the two programs that recognizes contractions rather than treating the apostrophe as a word separator. It is also extensible—as many as five thousand words can be added to its main dictionary and, if you should run out of room, you can create additional volumes of the dictionary on which to store additional words (up to five thousand on each disk onto which you've copied the main dictionary). You can do global adds to the dictionary by creating empty dictionary files for words of your own choosing to reside in; deleting words from the existing dictionary to

create room for more of your own words is also a relatively simple matter.

Apple Speller's dictionary is also listable, in whole or in part. If you want to know, for instance, all the words that begin with the letters *br*, you can ask that they be listed, and they will appear on the screen. If you have game paddles hooked up, you can vary the speed at which the listing scrolls, from slow to lightning quick.

When we used *Apple Speller* as a spell-checking program for some typical in-house files, it performed extremely well. While catching typos such as *overweigt* and *dlightful* and a misspelling of the word *aggression*, it recognized most of the words we used. This speeded up processing time considerably.

On-Line's *Dictionary* also sports a rather large (25,000-word) lexicon that will list and print out. As the accompanying documentation forewarns, listing the contents of the wordbook requires a lot of paper—approximately one hundred thirty sheets—but a printed copy of the main dictionary may come in quite handy when you want to add or delete words. *The Dictionary*'s wordbook is not listable in segments, however, so when you ask it to list, be sure you really mean it. If you change your mind once you've set things in motion, your best bet is to reset and start again.

The add and delete processes on *The Dictionary* are straightforward, and the program offers single-word add and delete as well as addition or deletion of groups of words contained in a particular file. Up to 1,500 words can be added to the main disk before a new one is needed.

Special Attractions/Idiosyncracies. Although your primary reason for purchasing a spell-checking program is to have a way of verifying the correctness of your spelling, various special features of the different programs are likely to interest you and may even sway the vote to the program you eventually decide on.

Goodspell's major pluses are its basic simplicity of operation and the fact that it requires only one disk drive.

Magic Words has various special features to recommend it. Among these are its file formatting capability and variable context length, its dictionary extension via applications dictionaries, and its ability to take advantage of a RAM card if you have one. Its correct-as-you-go capability and the ability to create new corrected files without requiring you to return to your word processor are also noteworthy features. In addition, *Magic Words* provides word counts of documents you check. The fact that *Magic Words* is a single-disk-drive program that doesn't call for a great deal of disk swapping may also rate as a plus, as may the fact that it is intended to work with a variety of word processor files in addition to *Magic Window*.

Among *Apple Speller*'s major pluses are the adequacy of its existing dictionary, its extensibility, and the ease of its add/delete dictionary functions. Also worth noting are the fact that its dictionary is listable, in whole or in part, and the fact that it recognizes contractions. Its documentation, simplicity of operation, and user friendliness are also strong points, as is the fact that it works with a wide variety of word processor files. *Apple Speller* gives you a word count of your file and also allows you to list all the words of your document in "almost alphabetical order." It will also provide counts of how many times each word occurs in your file and tell you how many unique words your file contains. From start to finish, this is an excellent program.

The Dictionary's special features include its compatibility with a variety of word processors, recognition of contractions, a listable and printable dictionary with global and single-word add/delete capabilities, and the fact that it will support either one or two disk drives. Unfortunately, single drive operation requires a lot of disk swapping. Screen prompting through this process is quite good, but the pace is annoyingly slow, especially on a long file. Also significant is its ability to speed processing by taking advantage of a RAM card.

The Components. *Goodspell* package consists of the master disk on which the program and the dictionary are contained and a short, straightforward documentation booklet. You can copy the master to create a backup, although you are not directed to do so. The documentation of the program and the explication of the way spell-checking dictionaries work is informative.

The *Magic Spell* package consists of a master disk and documentation in a three-ring binder. The program is copy-protected, but a card that's included in the package can be returned to Artsci in exchange for a back-up disk. Our impression is that *Magic Window* owners who use this program are likely to become proficient very quickly, but those unfamiliar with *Magic Window* are likely to have a few adjustments to make at first. While the documentation we read (a preliminary version) was clearly written, its organization was a bit difficult to follow. As we worked with the program (also in its prerelease version), there were times we wished for a bit more onscreen prompting. But once we got the hang of it, the logical organization of the program became clear.

The *Apple Speller* package provides two disks—the dictionary disk and the program disk. A card included in the package can be returned along with $7.50 to obtain a backup of the copy-protected program disk; the dictionary disk is copyable. The accompanying documentation, a forty-page softbound booklet, is clearly written and logically organized. It takes you step-by-step through the program and provides illustrations of the screen displays you'll see at various times during program operation. This manual is easy to refer to when you have a question about a particular function or feature.

The *Dictionary* package consists of a wordbook disk which is copyable and a program disk which is not, a backup for the program disk, and documentation. The preliminary version of the documentation we read is certainly adequate, and the illustrations of typical screen displays are useful.

Goodspell sells for $60, *Magic Words* for $69.95, *Apple Speller* for $75, and *The Dictionary* for $99.95. ■

Communications Software from the Utility Specialists

What do you really need in good communications software? We think the most important thing is using the equipment YOU have to transfer data easily and accurately. When Ascii Express made its first appearance in 1979, it was immediately recognized as the finest program of its kind available. Since that time, other programs have appeared with one or two flashy new features, but not a single one of them offers as many choices in terms of hardware compatibility and data transfer modes as the SDS line of communications software.

ASCII EXPRESS II™

If a versatile and accurate data transfer program is what you're after, Ascii Express II is for you. It offers more data transfer modes than any other program of its kind. In fact, we have yet to find a computer system it can't be used with.

In addition, Ascii Express II offers a built-in editor that allows you to produce or modify files before and after transfer, a built-in directory for your most frequently called numbers, and keyboard macros which allow you to send entire sign-on sequences or other host commands with a few simple keystrokes. All these features are designed to help reduce on-line time, and that saves you money.

Z-TERM™

For CP/M users, Z-Term offers all the above, and more . . .

- Support of all 80-column boards and external terminals
- Large copy buffer (41K)
- Unlimited download capacity – no file is too large

For special applications, Z-Term will even allow your Apple to emulate most popular types of terminals for maximum compatibility with your system.

Z-TERM "THE PROFESSIONAL"™

The ultimate communications package. This is the finest program we offer, and it is certain to set the standard for all communications software for years to come.

Z-Term "Pro" provides the solid performance that the truly serious user demands. Many of our users find Z-Term "Pro" to be so valuable that it more than justifies the purchase of the Z-80 Softcard.

Z-Term Pro's special features include:

- Support of the new Novation Apple CAT modem
- Support of specialized transfer protocols, e.g. PAN and Christensen
- Automatic answer and send/receive modes for unattended operation
- Special buffer for parallel printers to ensure no lost characters

We are so convinced that all the SDS programs are the finest available that we've compiled a detailed point-by-point comparison of our programs and those of our leading competitors (for your copy, send a stamped, self-addressed envelope to us at the address below). This comparison of more than 50 items clearly shows what our customers have known all along – that SDS software is the finest. Amen.

SDS
southwestern data systems

P.O. Box 582-S
Santee, CA 92071 Tel. 714-562-3670

CP/M is a trademark of Digital Research.
Z-80 Softcard is a trademark of Microsoft.

TRYING TO MAKE CONTACT ?

THE BASIC Solution

By Wm. V. R. Smith

Here it is, 1982. Only two years away from George Orwell's infamous *1984*. He predicted that, with the help of computers and electronic surveillance, Big Brother would be in control of every person's individual freedoms.

From the perspective of January 1982, it seems quite obvious that the concept of Big Brother is, after all, science fiction. Among other things, Orwell could not foresee the impact that the personal computer would have on the individual.

Nineteen eighty-two also brings Apple's sixth year in the microcomputer market. It seems almost unbelievable that a machine less than six years old could have such an impact on the American home and business.

In that six-year span, very little has changed in the basic Apple II system. Even though the electronic industry is developing new technologies almost daily, the Apple II is still one of the best computers on the market today.

The most curious point about this six-year-old Apple is the fact that even today programmers are still discovering new tricks the Apple can perform. Programs are running faster, sounding better, and, most important, offering visual effects that two years ago were thought impossible.

In this month's Basic Solution we are going to review seven memory locations that control hi-res graphic displays. A number of Basic Solution readers have requested a method to allow the Apple to draw on the hi-res screen without having the user view the action.

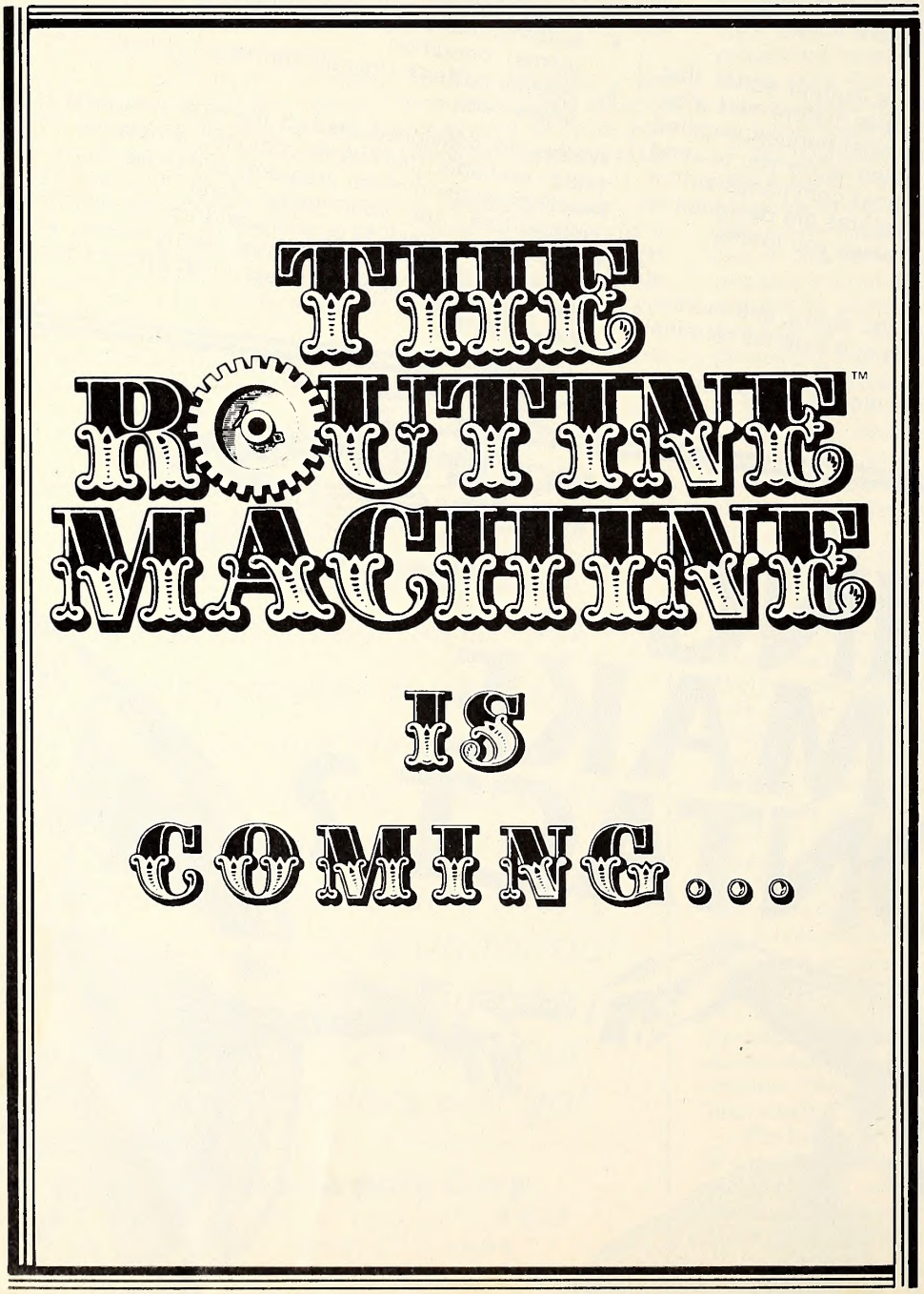
Let's take a look at seven basic commands that can be used to control upon which screen the text, hi-res 1, and hi-res 2 are displayed:

1. POKE −16304,0—Go to grophics
2. POKE −16297,0—Set hi-res grophics
3. POKE −16302,0—Full screen
4. POKE −16301,1—Mixed grophics, four lines text
5. POKE −16300,0—Page 1
6. POKE −16299,0—Page 2
7. POKE −16303,0—Set text mode

Most Basic users use the *hgr* or *hgr2* command to display the appropriate hi-res screen. These commands do, however, clear the screen at the same time they are being viewed. Clearing the screen destroys any graphs, pictures, or other information that may be on the screen. Any pictures at this point must be either drawn or *bloaded* in from the Apple disk. A neat trick following the *hgr* command is using POKE −16303,0. This returns the display to the text screen and still allows the Apple's hi-res drawing commands to operate. After the entire screen is drawn or loaded from the disk, two *pokes*, POKE −16304,0 and POKE −16297,0, redisplay the hi-res screen with whatever information was placed there.

This same system works for the second hi-res page, but the programmer must remember, on returning to text mode or to the hi-res screen, that the display must be set to the proper page. POKE −16299,0 sets the display to page 2. POKE −16300,0 sets the display to page 1. Even the Basic *text* command will not restore display to page 1. It must be manually done.

Many Apple disk owners have had the



opportunity to run the Apple picture disk. This disk, no longer available, contained many hi-res pictures; it cycled the user through them, one at a time, and then repeated the process. The Basic program that controlled this was very simple and used the display control commands we're discussing.

Creating your own picture disk is really very simple. First, a picture is *bloaded* into the first hi-res screen and the *poke* commands are executed to view the screen: POKE −16304,0: POKE −16297,0; POKE −16302,0. Once the user is viewing page 1, the second picture is *bloaded* into the second hi-res screen and the display is switched to the second page, POKE −16299,0. Now, once again, while the user is viewing the picture on the second page another picture is *bloaded* into the first hi-res page and the display is switched back to the first hi-res page with a single command, POKE −16300,0. This process is repeated until all the pictures are shown. On exiting the picture disk program, the programmer must remember to set the display back to page one and to text mode, POKE −16303,0; POKE −16300,0.

By using these few simple *poke* commands, your programs using hi-res graphics can have a much nicer appearance. Don't be afraid to play around with the commands. They won't damage any of your programs and, if you find you're lost, a simple press of the reset key will restore the values to their proper state.

The *Basic Solution* is looking for your comments and ideas on future articles and subroutines. If one of your routines is used in a future Basic Solution, a $10 credit toward your next software purchase will be available at your local computer store. Send your letters to Softalk Basic Solution, 11021 Magnolia Boulevard, North Hollywood, CA 91601.

Who Was That Child? Many readers knew the reference in the November column to be to Karl Friedrich Gauss. One, Fred Abraham of the Blueberry Brain Institute in Waterbury, Vermont, knew another, better documented, and more memorable version of the story, in which Gauss, after his first two years of formal education under the academic tyranny of an old-style Teutonic schoolmaster named Buttner, began his first class in arithmetic at the age of ten.

None of the boys in the class knew what an arithmetical progression was. Buttner took the opportunity to demonstrate his innate superiority by giving the class a lengthy addition problem like 15,486 + 15,659 + 15,832 . . . with one hundred terms and the same step (173) from one number to the next; which he would grandly solve with a simple formula while they all struggled.

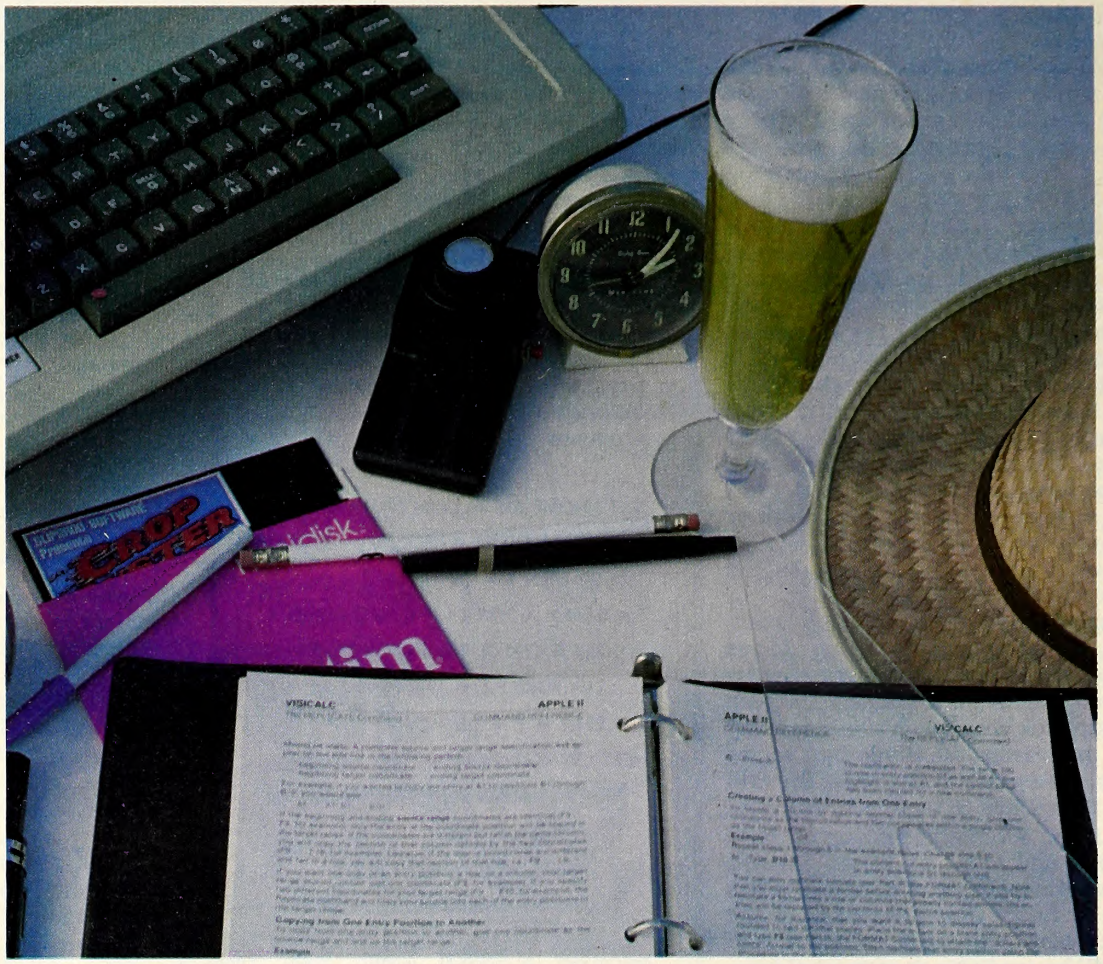
Immediately after Buttner stated the problem, Gauss dropped his slate to his desk, saying "Ligget se"—there it lies. Buttner assumed his youngest pupil was a moron. After an hour, he examined the students' slates, finding Gauss's bare except for the answer.

Buttner was impressed and bought the boy the best arithmetic text available, which Gauss mastered effortlessly. "He is beyond me," Buttner said; "I can teach him nothing more."

Abraham will receive a bonus $10 even though he wasn't first in his time zone, with thanks. Regular winners are: Richard A. Neumark, Saint Paul, Minnesota; G. F. Leeper, Winston-Salem, North Carolina; Richard D. Stratton, Colorado Springs, Colorado; and Dave Brockman, El Cerrito, California.

To Straighten the Record. As several of our readers observed and pointed out to us, there were some typographical inaccuracies in last month's Basic Solution program. We are including here, with apologies, a corrected version of that program:

```
100  TEXT : HOME :D$ = CHR$ (4): PRINT
       D$;"CATALOG":B = PEEK (37) - 2: IF B
       > 22 THEN B = 22
110  T = 0:CH = 4: FOR CV = 0 TO 23: GOSUB
       1000: IF C < > 160 THEN POKE P -
       1,219: POKE P,T + 193: POKE P +
       1,221:T = T + 1:S =CV
120  NEXT CV: VTAB 24:A$ = "TYPE LETTER TO
       RUN, OR LOAD=1 LOCK=2 UNLOCK=3
       DELETE=4 EXIT=5...."
130  B$ = "RUN": HTAB 1: PRINT LEFT$
       (A$,39);:A$ = MID$ (A$,2) + LEFT$
       (A$,1):K = PEEK ( - 16384): IF K < 128
       THEN FOR K = 1 TO 75: NEXT :K = FRE
       (0): GOTO 130
140  POKE - 16383,0:K = K - 176:IF K < 1 OR
       K > 5 THEN 300
200  HTAB 1: CALL - 868: IF K = 5 THEN GET K$:
       NEW
210  PRINT "PRESS /LETTER/ YOU WISH TO ";: IF
       K = 1 THEN B$ = "LOAD"
220  IF K = 2 THEN B$ = "LOCK"
230  IF K = 3 THEN B$ = "UNLOCK"
240  IF K = 4 THEN B$ = "DELETE": FLASH
250  PRINT B$;: GET K$: NORMAL : GET K$:K =
       ASC (K$) - 48
300  IF K < 17 OR K > T + 16 THEN 130
310  CH = 1:CV = S - T + K - 16: GOSUB
       1000: IF C = 194 AND (B$ = "RUN" OR
       B$ = "LOAD") THEN B$ = "B" + B$
320  FOR CH = 6 TO 39: GOSUB 1000:B$ = B$ +
       CHR$ (C): NEXT : HTAB 1: CALL - 868:
       PRINT B$: PRINT D$;B$: GOTO 100
1000 C1 = INT (CV / 8):CZ = CV - C1 * 8:P = 1024
     + 128 * CZ + 40 * C1 + CH:C = PEEK (P): RETURN
```



THE SECOND ANNUAL MOST POPULAR SOFTWARE POLL

MOST POPULAR SOFTWARE OF 1981

OFFICIAL BALLOT

These packages or programs are my favorites, in descending order, of all those released in 1981 with which I'm familiar:

1. _____ 6. _____
2. _____ 7. _____
3. _____ 8. _____
4. _____ 9. _____
5. _____ 10. _____

Name: _____

Address: _____

City/State/Zip: _____

Comments:

Please put a stamp on this ballot and mail it before February 15, 1982. It needs no envelope.

PLACE
POSTAGE
HERE

SOFTALK

11021 Magnolia Boulevard
North Hollywood, California 91601

When *Softalk* introduced the Most Popular Software Award last January, the vote was among all software produced prior to January 1, 1981. It seems a very long time ago when we recall that the winner was *Super Invader*. Few gameplayers remember that disk without fond memories of fun and frustration—but even *Super Invader* is long outdated now by developments in hi-res graphics, animation, speed, smoothness, and color. It's definitely time for a new reigning program.

The idea of the Most Popular Software Poll is to give you, the user, the chance to voice your opinion. The market may be the final polling place, but we believe that, occasionally, a real turkey sells very well either on the basis of its publisher's reputation or of very effective advertising; and sometimes sleepers appear—programs not given much fanfare that pick up speed well after their release, selling steadily but never making large waves in the realm of bestsellerdom.

By the time the votes were counted last year, we realized a problem. Many packages released during the last three months of the year were too new to be considered by the voters. Therefore, the software contending for this year's title includes all that released after October 1, 1980, and before January 1, 1982. Although that means some programs have two shots at the goodies, a program would have to be extremely outstanding to overcome having been released very late in the year. If such a program did win, it would not be eligible to win the following year as well.

The list shows many popular programs released in this timeframe. The list is not exhaustive, and you are welcome to write in other programs. The list is intended only to help define the timeframe and jog your memory.

The ballot has room for your top ten favorite programs. They may be programs of any sort; games, business, utility, whatever. All the programs you list will benefit from your vote. Each will be given a value equivalent to its placement on your list. When all these values from all ballots are totaled up for any program, they will determine its finishing place; so, conceivably, a program could win very few first place votes and still win the award because a great many people placed it second and the vote for first was fragmented.

The author and publisher of the software package you choose as number one will receive an award at the West Coast Computer Faire.

Rules. Only one ballot will be accepted per Apple user; that means that a family-owned Apple might generate four or five ballots—one from each member of the family. Where there are two ballots from one person, only one will be counted if the choices are the same and neither will be counted if the choices differ.

Your ballot must reach *Softalk* by February 15, 1982.

Your vote in a poll such as this helps to encourage better and better software for all of us. And, armed with that worthy purpose as a reason to get out the software you like best, pull up your Apple and start running your favorite programs to decide between them!

Here's a list of some of the programs eligible for this year's awards. This list was compiled from *Softalk*'s bestseller polls; it is not meant to be exhaustive, however, and you may vote for *any* program published between October 1, 1980, and January 1, 1982.

ABM
Akalabeth
Alien Rain (Apple Galaxian)
Alien Typhoon
Apple Adventure
Apple Galaxian (Alien Rain)
Apple Panic
Apple Pie
Apple World
ASCII Express II
Asteroid Field
Asteroids
Autobahn
BPI General Ledger
Castle Wolfenstein
Complete Graphics System
Computer Baseball
Cranston Manor
Creature Venture
Cyber Strike
Data Capture
DB Master
Disk Recovery
Dogfight
DOS 3.3
DOS Boss
DOS Plus
DOS Tool Kit
Dow Jones News & Quotes Reporter
Dow Jones Portfolio Evaluator
Dragon Fire
Enhanced MX-80 Graphics
Enhanced Paper Tiger Graphics
Epoch
Expediter II
E-Z Draw
Falcons
Financial Management System II
Galactic Trader
Gamma Goblins
Ghost Town: Adventure #9
Gobbler
Goodspell
Gorgon
Graphtrix
Hellfire Warriors
Higher Text
Hi-Res Cribbage
Hi-Res Football
Home Money Minder
Information Master
The Inspector
LISA Assembler
Lords of Karma
Magic Window
Market Charter
MasterType
Missile Defense
Mission Asteroid
Multi-Disk Catalog
Olympic Decathlon
Orbitron
Pegasus II
Personal Filing System
Personal Finance Manager
PFS:Report
Phantoms Five
Planet Miners
Planetoids
Pool 1.5
Prisoner
Program Line Editor
Puckman (Snoggle)
Pulsar II
Raster Blaster
Reversal
Robot War
Sabotage
Savage Island: Adventure#10
Sneakers
Snoggle (Puckman)
Space Eggs
Space Quarks
Space Raiders
Space Warrior
Star Cruiser
Star Thief
Super Disk Copy
SuperScribe
SuperText II
TASC
Tax Preparer
Tellstar
Ultima
VisiDex
VisiPlot
VisiTerm
VisiTrend/VisiPlot
Warp Factor
Word Star
World's Greatest Blackjack Program
Wizardry
Zork
3-D Graphics Animation Package

M.D. SOFTWARE

UNIQUE PROGRAMS FOR THE APPLE II*

™

Presents **Disc-O-Doc II** **By Stanley M. Dratler, M.D.**

The original **Disc-O-Doc** has been hailed by educators and programmers alike as the only disk utility program of its kind for the APPLE II. Now, it is even better and more powerful as **Disc-O-Doc II.** With it you will be able to read any diskette - *copyable or not,* repair crashed diskettes, change D.O.S. commands at will and even undelete programs deleted by mistake.

Disc-O-Doc II will save you many hours of work on files that have been ruined, and programs that need to be modified. You can now see how diskettes were created to be uncopyable and even learn how to do it yourself. Finally, the secrets of D.O.S. will be secrets no more. Along with the extensive documentation included you will be able to learn many facets of the workings of D.O.S. Easy to use, too powerful not to!

And now **Disc-O-Doc II** contains many unique new features.

- HALF-TRACKING — now you can even read those diskettes that are half-tracked. Track 1.5, 2.5, etc.
- PRINTER OUTPUT — send all of the information you find out to any standard printer card
- SELF DIAGNOSIS — tells you instantly what part of D.O.S. has been changed on any disk to make it uncopyable.
- AUTO TREATMENT — just tell Disc-O-Doc II what to do and it will change D.O.S. in all the right places.
- SINGLE SECTOR OR WHOLE TRACK — your choice read or write all or part only.

Disc-O-Doc II is supplied on a write protected diskette with complete bound documentation for only **$99.95**

Word Processor II

This is a text editor that outperforms any other on the market. Compare it to any word processor and you'll buy **Word Processor II** without a doubt. Just check the review in Oct. 1981 Call-A.P.P.L.E.

- **Form Letters** - makes individual letters that seem to be personalized with up to 255 different fields.
- **Text Files** - works with any standard sequential text file including most 6502 assemblers.
- **Conditional Text** - checks for a certain condition and inserts necessary text completely formatted.
- **Speed** - ultra fast loading and saving of text. Puts formatter into any 16K card for instant printing.
- **Compatability** - works with most 80 column cards without any modifications - DOS 3.3 or 3.2.
- **True Upper/Lower Case** - both on the screen with adaptors and in the file.
- **Economy** - compare the price and you'll be convinced

WP II only $75.00 Lowercase Rom for Rev. 7.+ **$45.00**

Introducing SIGHT 'n SOUND **By Ray Balbes, Ph.D.**

Enhance any stereo or cassette recording with a fantastic, fully SYNCHRONIZED HI-RES light show. Use basic designs of your own creation - or use those supplied - to enjoy an unlimited number of visual images with the sound. The images are controlled by the sound frequencies in real time.

★ Kaleidoscope light shows ★ Guitar player who sings & plays to the beat ★ Geometric design light shows
★ Fast machine language programs are used to synchronize audio & visual ★ Take your own 'EKG'
★ 23 page, clearly written manual tells:
- How to create new light shows
- How to add light shows to your own programs

★ Watch your voice on the 'oscilloscope'
★ Powerful design/editor for creating light shows
★ Introductory price **$24.95**

For any 48K Apple II or II+ with Applesoft & a disk drive (DOS 3.2 or 3.3) No hardware modifications

MasterCard VISA

FOR CHARGE OR C.O.D. ORDERS CALL TOLL-FREE **800-227-1617** ASK FOR EXT. **372**
IN CALIF. — 800-772-3545

OTHER CALLS AND INFORMATION 314-725-1110

Add $2 postage & handling for each program ordered. Foreign orders add $7 for AIR MAIL POSTAGE

7225 CLAYTON ROAD, ST. LOUIS, MISSOURI 63117

*Apple II is a registered trademark of Apple Computer Inc.

TRADETALK

from page 44

A2M3-139070, A2M3214443, others unknown
1 Diablo 1620 Hytype II printer, CS10008
1 Hitachi 13-inch Amdek color monitor
1 Panacolor 11-inch monitor, Japanese markings, TQF-15626
1 Video-100 Amdek 13-inch b&w monitor
1 Sanyo 9-inch green-screen monitor, 65002370
1 Sanyo 9-inch b&w monitor model VM4509, 60912162
1 Samsung 13-inch color TV, 19588

☐ **Diarmuid McCarthy,** a professor of computer science at the College of Design and Marketing, Dublin, Ireland, has an Applesoft tutorial on disk that he wishes to make available to Apple users everywhere (see *Softalk,* December 1981, page 26). The disk, which requires DOS 3.3 and 48K, can now be obtained from three different sources in the United States. The **International Apple Core** is featuring it as its disk of the month for January and distributing it to Apple user groups worldwide. **Micro Lab,** 2310 Skokie Valley Road, Highland Park, IL 60035, will furnish the program to anyone for a $7 charge that includes costs of disk, duplication, and shipping. And **David Alpert** of **Omega Microware,** 222 S. Riverside Plaza, Chicago, IL 60606, has offered to provide it for anyone who will send him a disk and 56 cents in postage. He requests a self-addressed label and shipping material adequate for a round trip.

☐ **Canoga Data Systems** have expanded their facilities in Canoga Park, California. The change marks a 50 percent increase in facility size adjacent to their previous location and consolidates design and product planning groups under one roof, as well as giving the manufacturing operation additional space needed to support increased sales volume of Canoga's multiplexer and modem product lines.

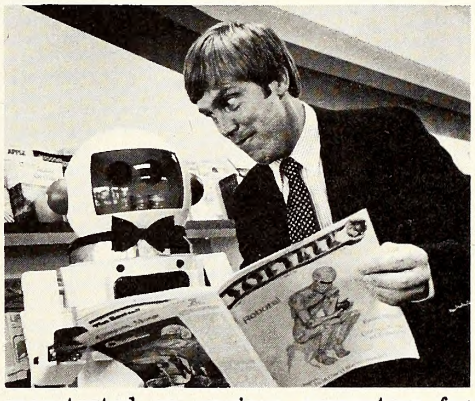
☐ Comdex drew a crowd from all over the world to the jewel of the desert—or was Vegas itself part of the attraction? No one enjoys a good bet any more than a good Australian, and **Harry Harper,** guiding light of **Zofarry Enterprises** just outside Sydney, became a big winner in Las Vegas no matter what he may have dropped at the tables. He called in on his Source account that Friday and found his wife **Zofia** (the Zof part of Zofarry) had given birth to a **Simon Christopher Harper.** Simon's father was here to light a fire under Vista Computing, who've been just about to release Zofarry's Vision 80 eighty-column card since the West Coast Computer Faire last April. The kindling took, and the card is finally available. Just in case you have trouble, though, you can get it through Zofarry Enterprises itself by writing to them at 6A Burwood Road, Burwood, NSW 2134, Australia. Service? Well, Harry gets over here about every six months, so. . . . In fact, according to Harper, the manual has schematics of all the innards so any technician could service your card. If yours couldn't, Zofarry would replace your card with a new one immediately. Harper enjoyed the novelty of spending Thanksgiving here; it's a strictly American holiday. With nothing but football and feasting, he was mulling over ways to get one instituted Down Under.

☐ **Daniel S. Bricklin,** chairman of the board of **Software Arts,** has been named winner of the Grace Murray Hopper Award for his work in the design and development of *VisiCalc*, the most popular microcomputer program ever created. The award has been given annually since 1971 by the Association for Computing Machinery in recognition of major computing achievements made by individuals under thirty years of age. It includes a cash award of $1,000 from Sperry Univac, which Bricklin will donate to the Boston Computer Society.

☐ For the third year in a row, the **Midwest Affiliation of Computer Clubs**, representing about a thousand personal computer enthusiasts in twenty clubs throughout the midwest and southern Canada, will hold its annual exhibition of recent advances in the industry—featuring lectures, demonstrations, contests, and a flea market—at Franklin University in Columbus, Ohio. Potential speakers are urged to contact the MACC as soon as possible before the start of Computerfest 82 on June 18; as are potential exhibitors who wish to reserve booth space—fifty booths available on a first come, first served basis; early subscription rewarded with a discount. Write to Midwest Affiliation of Computer Clubs, c/o Prof. Don Moore, MACC President, 201 South Grant Avenue, Columbus, OH 43215.

☐ **Microcon SoftwareCenters**, New England's first microcomputer software retail/consulting store, has officially opened in Watertown, Massachusetts. **Barry Passen**, president, said that sales and inquiries during the store's first two weeks exceeded his original business plan projections. Strongest customer response has come from small business operators able to discuss their needs with MSC business consultants and check out software on the appropriate hardware in the store's showroom. Passen plans to open two more stores in 1982 in major east coast cities and projects one hundred stores by 1985.

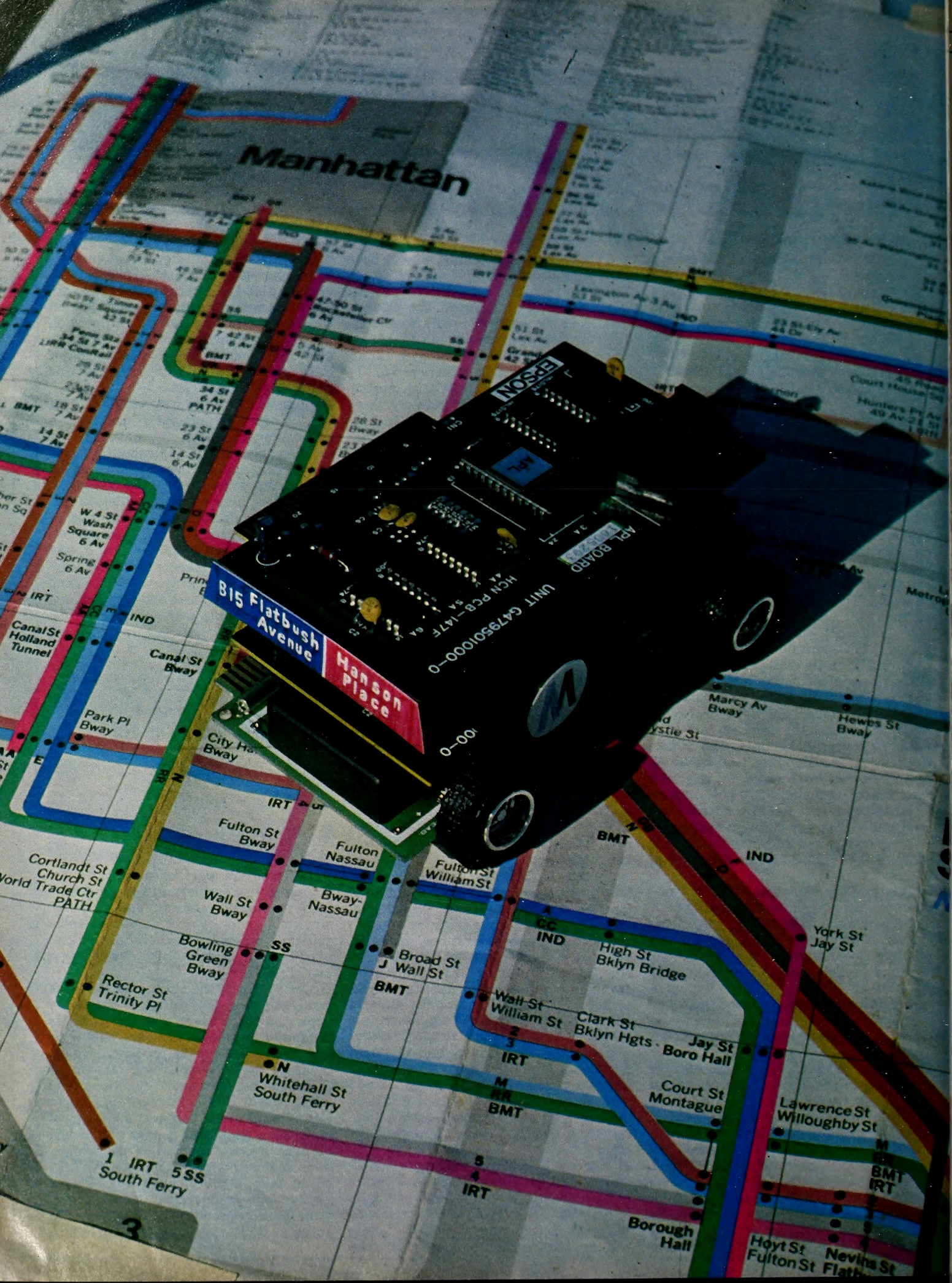
☐ Meanwhile, on the west coast: Calling for "a computer on every desk and a robot in every broom closet," Proto the robot concluded his ribbon-cutting ceremony address to a thousand admiring listeners and officially opened **The Softwaire Store** in Los Angeles. **Glenn Johnson,** the store's founder, wants it to be used "in the same context as record stores and video stores," and has designed it as a supermarketlike central location for computer books and magazines, plus games and personal and business CP/M programs that can be demonstrated on various computers for comparison. Seminars are being scheduled to instruct present and potential small computer owners on how to apply software programs successfully and to explore future needs. A free newsletter is available.

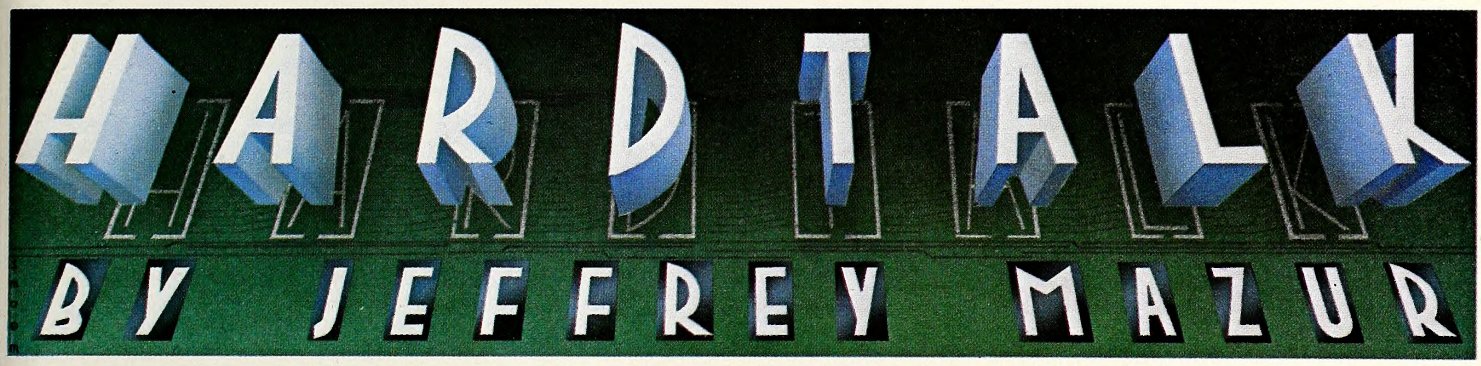


☐ **Christy Lee** is the latest addition to the increased personnel roster in the newly expanded office space of **Small Business Computer Systems** (Lincoln, NE). She will be assisting with product development for the accounting software firm.

☐ Expanding its sales functions and staff, **Olivetti OPE** (Tarrytown, NY) has appointed **Steven Wallace** manager of their new technical support function. Wallace leaves his position as applications engineer with the company to take on the responsibility, which consolidates applications engineering and field service. In further moves to strengthen sales activity, OPE appointed **Lee H. Heller,**

GOTO 151


Manhattan
B15 Flatbush Avenue
Hanson Place
EPSON
APL BOARD
UNIT GA79501000-0
HCN PCB-147F
W 4 St Wash Square
6 Av
Spring 6 Av
Canal St Holland Tunnel
Canal St Bway
Park Pl Bway
City Ha Bway
Fulton St Bway
Fulton Nassau
Fulton St William St
Cortlandt St Church St World Trade Ctr PATH
Wall St Bway
Bway-Nassau
Bowling Green Bway
Rector St Trinity Pl
Broad St J Wall St
Wall St William St
High St Bklyn Bridge
York St Jay St
Clark St Bklyn Hgts
Jay St Boro Hall
Court St Montague
Lawrence St Willoughby St
Whitehall St South Ferry
Borough Hall
Hoyt St Fulton St
Nevins St
Marcy Av Bway
Hewes St Bway
IRT
IND
BMT



Boarding the Apple Bus, Part 1

Welcome to Hardtalk. This new *Softalk* feature will explore the hardware side of the Apple II. Our hope is that this column will teach everyone a little more about how the Apple works, as well as how to connect it to other devices. We will also try to evaluate the various peripherals on the market so you'll find it easier to determine which ones are right for you.

Let's start by taking a good "hard" look at the Apple II. Remove the cover from your computer and gaze inside. There's nothing very exciting in there—just a bunch of little black boxes called *integrated circuits* (also known as *IC*s or *chips*). Now run your favorite game program or watch the computer sort a mailing list of several hundred names. Listen to an Apple synthesize speech or recognize verbal commands. Or, how about programming the computer to calculate pi to 1,000 digits? Now take another look at those little black boxes that make all of these things possible. To most people, such a feat is nothing short of pure magic!

Indeed, the Apple II represents a miraculous achievement. One of the goals of this column will be to strip away some of the magic behind the Apple so you'll understand better what makes the computer tick.

Although you may not become an electronic design engineer, you will learn how to connect almost any type of equipment to the Apple. This will include ready-made peripherals such as printers, storage devices, and so on, and, from time to time, information on how to "roll your own" circuits using *PIA*s, *ACIA*s, and more. Judging by the positive response to Silas Warner's article in October's *Softalk*, "The Controller Even You Can Make," it would appear that a great many Apple owners are interested in this subject.

Before we do anything else, however, let's review the history of computers to see how the Apple came into being.

What Is a Computer? In the world of computers, the Apple II stacks up as a relatively small, slow, and limited machine. It is considered a *personal* computer because it is usually used by only one person at a time, as contrasted to larger time-sharing computers that can handle many users simultaneously.

In most respects, however, the Apple shares the same design as even the most powerful computers. It is a stored-program, digital, electronic machine. This rather lengthy description reflects the entire history of computers.

First of all, the computer is a machine. This means that it does work, or helps accomplish a desired task. A lever is also a machine—it can be used to move objects that would otherwise be too heavy to move. The lever works by transforming a small force over a large distance into a larger force over a smaller distance. In obedience to the laws of physics, *energy* is conserved (neither created or lost), but the desired action occurs: a small force is changed into a larger one. While levers help accomplish physical work, computers aid in the performance of mental work.

A lever is also classified as a mechanical machine and indeed the first "mental machines" were also mechanical devices. In fact, the abacus could be considered one of the first mental machines. There have been many others since then, including slide rules and adding machines. The modern computer, however, is an electronic machine. This means that it uses stationary electric current controlling devices instead of moving mechanical parts. The electronic analogy for the lever is an *amplifier*. An electronic amplifier is nothing more than a device that can use a small current or signal to control a larger one. As with the lever, energy is still conserved because the larger current must be supplied by some other source; but it is the *control* of the currents that becomes important in electronics.

Almost every electronic device is based upon the use of amplifiers. Originally, vacuum tubes were the primary element; now solid state transistors have replaced tubes. Depending upon their associated circuitry, amplifiers can also be used to accomplish other tasks, such as inverting, adding, performing logarithms, or storing electronic signals.

At this point in history, computers were dealing with analog voltage signals much in the same way that a slide rule uses analog lengths to accomplish multiplication and division. Then it was discovered that almost any information could be processed digitally, and this was something that electronic circuits could do exceedingly well. All information to be processed by the computer, whether in the form of numbers, text, or electronic signals, was first converted to a maze of binary ones and zeros. Dealing with only digital signals makes the electronics much simpler. Amplifiers, for example, need worry no longer about nonlinearities or distortion, since they are operating either fully on or fully off.

This development gave rise to some specialized digital circuits called *gates*. A gate can have one or more inputs but only one output. The state of the output (either high or low) is determined by the state of its input(s) according to a truth table defined for its function. For example, the output of an *and* gate will be high only when all of its inputs are high. Similar definitions were worked out for *not* gates (inverters), *or* gates, *nor* gates, and so on. From these primitive gates it is possible to build more complex circuits to accomplish such tasks as the adding of two binary numbers. A particular arrangement of gates worth mentioning is the *flip-flop*. This is a circuit that can hold its output in one state even after the input signal is removed—it is the basic memory element. Thus was born the digital electronic computer.

One major stumbling block still remained, however. These

HARDTALK GLOSSARY

ACIA. Asynchronous Communications Interface Adapter. A large scale integrated circuit that allows bus-oriented systems such as the Apple to communicate serially with other devices. Can be the basis for a serial interface board.

Address bus. The collection of related wires that carries a unique combination of signals (the address) to specify what device the CPU is communicating with. The Apple's CPU has a sixteen-bit address bus, allowing for 65,536 (two to the sixteenth power) different addresses. Every other device in the Apple, including RAM, ROM, and on-board and peripheral I/O, is assigned a particular address or range of addresses.

Analog. A signal or entity that can have an infinite number of states or values. Most characteristics are analog in nature—for example, length, time, and speed. Usually used in contrast to *digital.*

Architecture. The way in which a CPU or computer system is designed. For example, it refers to the number and size of the internal registers in a CPU, how instructions are executed, addressing schemes, and so on.

Binary. The base-two number system. It is the simplest number system, since it has only two digits: 0 and 1. This is very simple to represent with electronic circuits and thus has become the basis for all digital computers.

Bit. Binary digIT. The smallest unit of digital information. Can take on one of two values: 1 (or *high, true,* +5 volts) or 0 (also *low, false,* 0 volts).

Board. Short for printed circuit board. A collection of electronic devices mounted on a suitable material, usually fiberglass, with copper *traces* forming the connections between the various devices. Can refer to any peripheral that plugs into one of the I/O connectors.

Bus. A collection of wires that transmit information (electrical signals) from one circuit to another. Three buses are often referred to in the Apple. Two of these are the 6502 CPU's address and data buses. The eight peripheral connectors at the rear of the computer are all basically tied together and are sometimes called the Apple bus.

Byte Eight bits. The basic unit of data the Apple's CPU can handle. A byte can be used to represent any alphanumerical character or a number between 0 and 255.

Card. See *board.*

Centronics. Actually the name of a company that manufactures printers. However, the interface configuration used by them has become a standard for parallel connection of peripherals (usually printers).

Chip. See *IC.*

Clock. A very rythmic signal used to keep all of the computer's circuits synchronized. Very similar to the coxswain on a scull that keeps the oarsmen in sync.

CP/M. Control Program for Microprocessors. One of the first operating systems available for small computers based on the 8080 (or Z-80) CPU. It has become the standard for software distribution among such computers since it allows a program written on one machine to run without change on any other computer that uses CP/M.

CPU. Central Processing Unit. The "brain" of any computer system. Like the conductor in an orchestra, it is the central figure that directs the performance of the other circuits. Usually one of the large ICs in a computer,

Data bus. The collection of related wires that carries the actual information to be transferred into or out of the CPU. The Apple has an eight-bit data bus which, among other things, allows it to operate on one character at a time.

Digital. A form of electronic circuitry in which all information is processed as binary states (either *on* or *off*).

DMA. Direct Memory Access. A process whereby a peripheral reads or writes directly into memory without going through the CPU.

Eprom. Eraseable PROM. A non-volatile memory element that can be programmed and then erased by exposure to ultraviolet light (this wipes out the entire array, requiring the chip to be completely reprogrammed).

Eeprom, Eaprom. Electrically Erasable (Alterable) PROM. Similar to an EPROM, but can be erased while still in operation, sometimes on a bit-by-bit basis.

Firmware. A program that has been stored in some sort of non-volatile memory device.

Flip-flop. An electronic circuit that can hold a given logic state until directed to change. An R-S (Reset-Set) flip-flop has two inputs; one input sets the output high and holds it there until there is a signal on the other input which brings the output low. A toggle flip-flop has only one input, causing the output to change state each time there is an input signal. Another type of flip-flop is the D (data) latch. This device has a trigger input that is used to grab the state of the input signal and hold it until the next trigger pulse.

Handshaking. The control signals used to coordinate communications between two devices. Makes sure that no data is lost because one device was not ready.

Hertz or Hz. The basic unit of frequency equal to one cycle per second.

early computers were programmed by using wires to connect various circuit elements. Thus, making a simple change in the program could involve a complicated, time-consuming re-wiring. Programs were also limited in size by this technique. The idea of storing the program itself within the machine's memory was the final link to today's computers. Replacing the wires with a bit pattern in memory means that programs can be made more flexible and are limited in size only by the amount of memory available.

Stored-program machines operate by executing instruction cycles. Each cycle begins by having the CPU *fetch* an instruction. This simply means that the CPU looks in a particular area of memory (where the program is stored) and reads a byte of information. This information is then decoded by the CPU to direct its further action. Most of the time, the CPU will need one or two other bytes of *data* to finish the operation. All of this takes place very rapidly—in the Apple, up to one half million instructions can be executed in one second.

The stored-program technique, coupled with advances in digital electronics, has been responsible for the fantastic growth of computer technology since 1950. The last thirty years have brought about dramatic increases in speed and memory size, but the basic concepts of the stored-program, digital, electronic computer have remained the same—whether applied to the latest state-of-the-art mainframe or to the Apple II.

All aboard the Bus! Now that you know what type of computer the Apple is, take a look at figure 1, a block diagram of the Apple II. In the upper-left corner is the reference oscillator and system timing block. A couple of transistors and a quartz crystal form this circuit, which is the heart of the computer. The circuit they comprise, known as an *oscillator*, generates a signal that oscillates, or swings back and forth, at a very precise rate. In the Apple II, this circuit oscillates over fourteen million times per second.

This signal is then divided down or converted into other signals which control all aspects of the computer's operation, including CPU functions, video generation, and memory refresh. Everything is tied to the master oscillator or clock; if it stops, so does the computer. 6502-based computers, such as the Ap-

GOTO 148

IC. Integrated Circuit. A collection of transistors and other related electronic components on one piece of silicon (the material used for almost all active electronic components). Metal leads, or pins, allow connection of the internal components to an external circuit.

Interface. The means by which two things communicate. In particular, the term refers to the electrical configuration that allows two or more devices to pass information.

Interrupt. Not exactly a hardware term, it refers to a process most CPUs can perform. If while the computer is executing a given program, an interrupt signal comes in on one of the CPU's special pins, it temporarily halts the current program, executes a predetermined interrupt routine, and then resumes working on the original program. The term refers to the actual signal, usually generated by one of the peripheral boards in the computer, that is sent to the CPU.

K. Stands for kilo, the metric prefix for 1000. KHz means kilohertz, or 1000 cycles per second. However, when referring to certain binary values such as memory size, K may actually equal 1024.

Latch. See *Flip-flop.*

Logic State. The condition of a digital signal. May be either *high* or *low* (1 or 0, +5 volts or 0 volts) *true* or *false.*

Microprocessor. See *CPU.*

Motherboard. The main circuit board of all complex electronic systems, particularly bus-oriented ones.

Parallel. When refering to communications techniques, a system where data is transferred more than one bit at a time (for example, eight bits together).

PIA. Parallel Interface Adapter. A large scale IC used to connect parallel data signals to a bus-oriented system such as the Apple.

Pin. A small metal connection device such as is found on the outside of ICs or at the "fingers" of a peripheral card that goes into one of the I/O connectors. Since all pins look alike, they are usually numbered in some standard fashion.

PROM. Programmable ROM. A ROM that is programmed after it has been made. Usually requires special programming apparatus.

Pulse. An electronic signal which changes state for a relatively short period of time.

Refresh. The preservation of information held in a dynamic memory element. Such memories store data as a minute charge that rapidly diminishes. By accessing the memory, the charge is replenished; a certain portion of all the memory cells, therefore, must be accessed at least several times per second.

RAM. Random Access Memory. The main working memory of any computer. In most small computers, anything stored in RAM will be lost when the power is turned off.

ROM. Read Only Memory. A memory device in which information is permanently stored as it is being made. Thus, this information can be read out but not changed.

RS-232. A standard specification for serial data transmission. It defines the voltage levels, connector assignments, and signal types for this form of communication.

S-100. One of the first and most popular bus structures for 8080-type computers. Like the Apple bus, it defined what signals would be on what pins, connector sizes, and so on, but it had twice as many connections.

Serial. A form of data communication where information is passed one bit at a time. See RS-232.

TTL. Transistor-Transistor-Logic. The most popular of all logic IC families. These devices are characterized by their operation at 5 volts, with any signal above 2.0 volts considered as *high* and anything below 0.8 volts being *low.*

Volatile. Refers to memory that forgets whenever the power is removed.

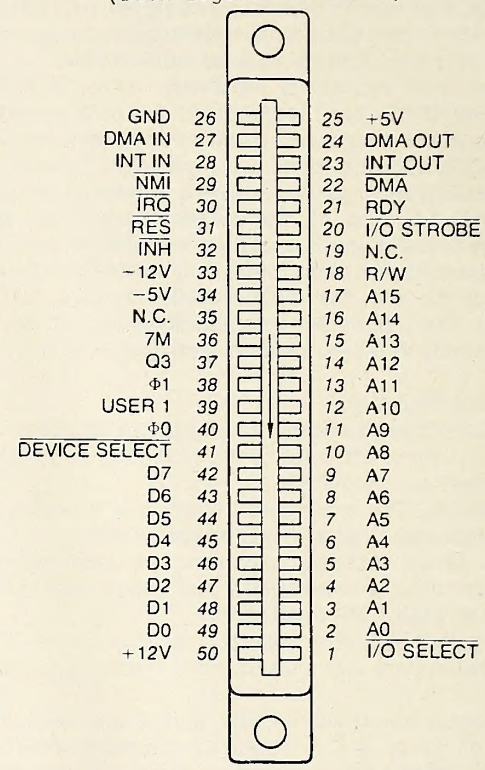
Z-80. An improved version of the 8080 that was completely compatible with all software written for the 8080.

6502. The CPU chip used in the Apple as well as in many other computers.

8080. An older generation CPU chip that helped power the tremendous growth of small computers.

PERIPHERAL CONNECTORS
(EIGHT OF EACH)

TOP VIEW
PINOUT (Back Edge of PC Board)



(Toward Front Edge of PC Board)

LOCATIONS J2 TO J12

Figure 1. Apple II Block Diagram

Chart reprinted from *Apple II Reference Manual* (January 1978) with permission.

from page 146

ple, also use a two-phase clock that is generated by the CPU from the master clock. These are two signals which are complimentary, that is, when one is high, the other is low. Labelled phase one (1) and phase two (2), they help coordinate data transfer in and out of the CPU. For example, when the CPU wants to read data from memory, it will put out the appropriate address on the address bus while the phase one clock is high. Then it will transfer in the information on the data bus at the end of the phase two clock. These clock signals will be used by almost all other devices connected to the CPU.

Next comes the sync counter, which has the job of handling the video display. It makes sure the video memory is read out at the proper rate so that the video signal is compatible with existing TV standards (60 Hz frame rate in the United States, 50 Hz for European Apples). In between each CPU cycle, the sync counter grabs a byte of data from the video RAM and passes it to the video generator. The sync counter also accomplishes the necessary refresh of the dynamic RAMs.

The video generator takes the data received from memory and creates the proper picture display, either text or graphics. Text characters are formed by reading out of a pre-programmed ROM that determines which dots must be turned on within a 5x7 array to form the desired image. The top row of dots for all 40 characters on a line is generated first. Then come the second and third rows and so on. After eight scan lines on the monitor, a complete line of forty characters will have been displayed.

The next block represents the power supply which, as its name denotes, supplies power to all of the circuits in the computer. By the way, this is a switching-regulator-type supply, which makes it much smaller, lighter, and more efficient than a conventional supply.

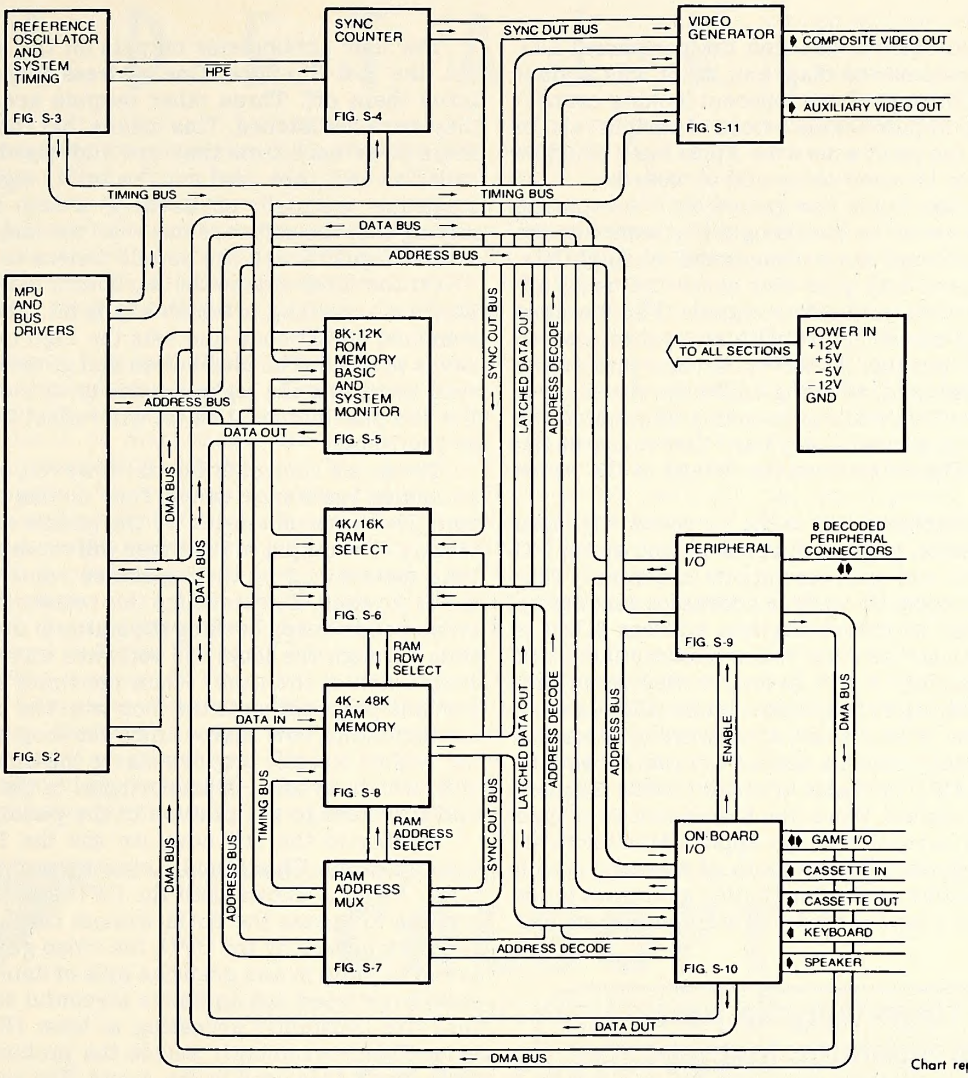


Figure 2.
Apple Peripherol
I/O Connector

Chart reprinted from *Apple II Reference Manual* (January 1978) with permissian.

Jumping to the next square, you find the peripheral I/O. This may be a small step on the diagram, but it was a giant leap for the Apple II. Through these innocent-looking connectors in the back of the computer came a flood of peripherals. In fact, we have reached the point where the Apple itself could be considered a peripheral to some of its add-on boards.

One of the reasons the Apple has gained such acceptance among peripheral makers is the forethought that went into the design of the peripheral connector arrangement, or *Apple bus*. The Apple bus consists of fifty pins that carry the usual address, data, timing, control, and power signals. For the most part, this is a standard bus structure with all connectors tied together, pin for pin, in parallel. However, several pins on the connectors have been isolated, allowing additional decoding to make each slot unique. This additional coding eliminated the need for several chips on almost every card that would be designed for the Apple. Figure 2 shows the details of the Apple bus.

Coupled with the peripheral I/O is the *on-board I/O*. This term refers to the cassette, keyboard, speaker, and game I/O connector. The cassette and speaker outputs are simply flip-flops, or latches, that respond to various addresses sent out by the CPU. For example, whenever the hex address $C020 is placed on the address bus, the state of the cassette latch will change; that is, if it was high it will go low, or vice versa. Address $C030 does the same for the speaker latch. Although one such event would not be of much use, a software loop routine can trigger the latch many times a second to create a square wave signal. Since the CPU operates at about 1 MHz, frequencies up to 100 kHz are possible. When you have an error or type control-G, for example, a routine in the Apple's Monitor firmware generates a 1 kHz tone for one tenth of a second which emanates from the speaker. A similar routine generates tones that can be recorded on a tape recorder to store programs and data.

The four annunciator outputs on the game I/O connector act like R-S flip-flops. One address turns them on, another turns them off. Three other outputs are provided, although they were not latched. This means that they will put out a very short pulse each time they are addressed. These pulses, also called *strobes,* are used as: 1) a utility signal on the game I/O connector (normally not used), 2) a clear keyboard strobe signifying that the software has read the last pressed key, and 3) a start signal to tell the paddle timers to begin counting.

On the input side is the keyboard, which generates seven-bit code according to which key is hit. The keyboard uses the standard ASCII code and sets the high order bit whenever a key is pressed. The pushbutton and game controller inputs are read by testing the highest order bit of their memory location. For the pushbuttons, this bit will reflect the TTL level present on that input.

The game controller inputs, however, are used to represent an analog resistance value. They do this by using a timer circuit. The timer is started by the paddle strobe described previously. The output of the timer will remain high for a period of time determined by the resistance value of the external paddle or joystick. To determine this resistance, a software loop is employed to keep checking the status of the timer output. Each time through the loop, the software increments a counter to keep track of the time. When the timer's output finally goes low, the program exits the loop with the contents of the counter indicating how many times the loop executed. This number will be directly proportional to the time constant of the circuit, which, in turn, is proportional to the external resistance and therefore to the position of the paddle or joystick.

Moving to the left now, we see the RAM address MUX (multiplexor). This circuit takes care of two important functions. First, it allows both the CPU and the video generation circuits to access the RAM without conflict. In between each memory access by the CPU, the video generation circuit is allowed to jump in and grab one byte of data. Since the video circuits must read out an entire screenful sixty times a second, they are constantly accessing at least 1K of memory at this rate. This conveniently solves the problem of refreshing the dynamic RAMs used in the Apple. The second function of the address MUX is to satisfy the memory chips' addressing scheme whereby the twelve-bit address is sent to the chips six bits at a time. Further control is provided by the RAM select circuits (which also used to allow either 4K or 16K RAM chips to be used).

The next block represents the 8K to 12K of ROM that holds the system Monitor and either Integer or Applesoft Basic. This block is what gives the Apple its intelligence when it is first turned on.

Last, but certainly not least significant, is the CPU itself, along with some associated circuits that let it control the rest of the computer. Also shown on the diagram is the DMA (Direct Memory Access) signal which can be used to transfer data into or out of memory without using the CPU. During a DMA request, the CPU's address and data buses would be shut off, allowing the requesting device to take over these buses and access memory directly. Although this is a powerful technique, it is rarely used on the Apple. With the exception of DMA and video generation (which is a form of DMA), all data is transferred via the CPU.

A Preview of What's To Come. Now that we have some background on how the Apple works, we can begin to examine the numerous devices that can be attached to the computer. Next month, we will explore printers and how to hook them up. In the future we will cover such subjects as prototyping, firmware expansion, I/O protocol, and interrupts. Hardware reviews will include paddle/joystick selection, an IEEE GPIB controller, cooling devices, mass storage alternatives, and co-processor boards. If you have questions, topic ideas, or products you wish to see reviewed, please send your comments to Hardtalk, 11021 Magnolia Boulevard, North Hollywood, California 91601.

TRADETALK

from page 143

formerly sales administration manager for Ontel Corporation, head of marketing administration. **Dennis Boland,** formerly with Perkin-Elmer Corporation, has been named mid-Atlantic regional sales manager, and **Jerry W. Korsbon,** late of Perkin-Elmer, head of midwestern regional sales.

☐ **California Computer Systems** (Sunnyvale, CA) has announced the appointment of **Gail R. James** to the position of executive vice-president of marketing. James comes to the new position with the three-year-old company from Qume/ITT, where he was vice-president of marketing. He will be overseeing projects involving new marketing, product, and organizational opportunities.

☐ The second store in a planned national chain has opened in White Plains, New York, under the banner of **Programs Unlimited Computer Centers.** Company president **Richard Taylor** bases the store's approach on giving customers a hands-on opportunity to run software and computers before purchase. Future branches are planned for New York City and north and central New Jersey.

☐ "It's a tough one, and no, we don't have a winner yet," says **Cheryl Stinson,** Marketing Administrator of **Dakin5 Corporation** (Denver, CO). **Level 10,** a division of Dakin5, is offering a seventy-five hundred-dollar reward to the first person who solves their new adventure game, *Alkemstone.* Stinson says the company receives calls every week from players around the country who want clues.

"We listen to their progress in the game, but that's all we do. We can't tell them if they're on the right track or not," says a very secretive Stinson. "But we will announce the winner as soon as we have one. We wouldn't want to leave people hanging, after all."

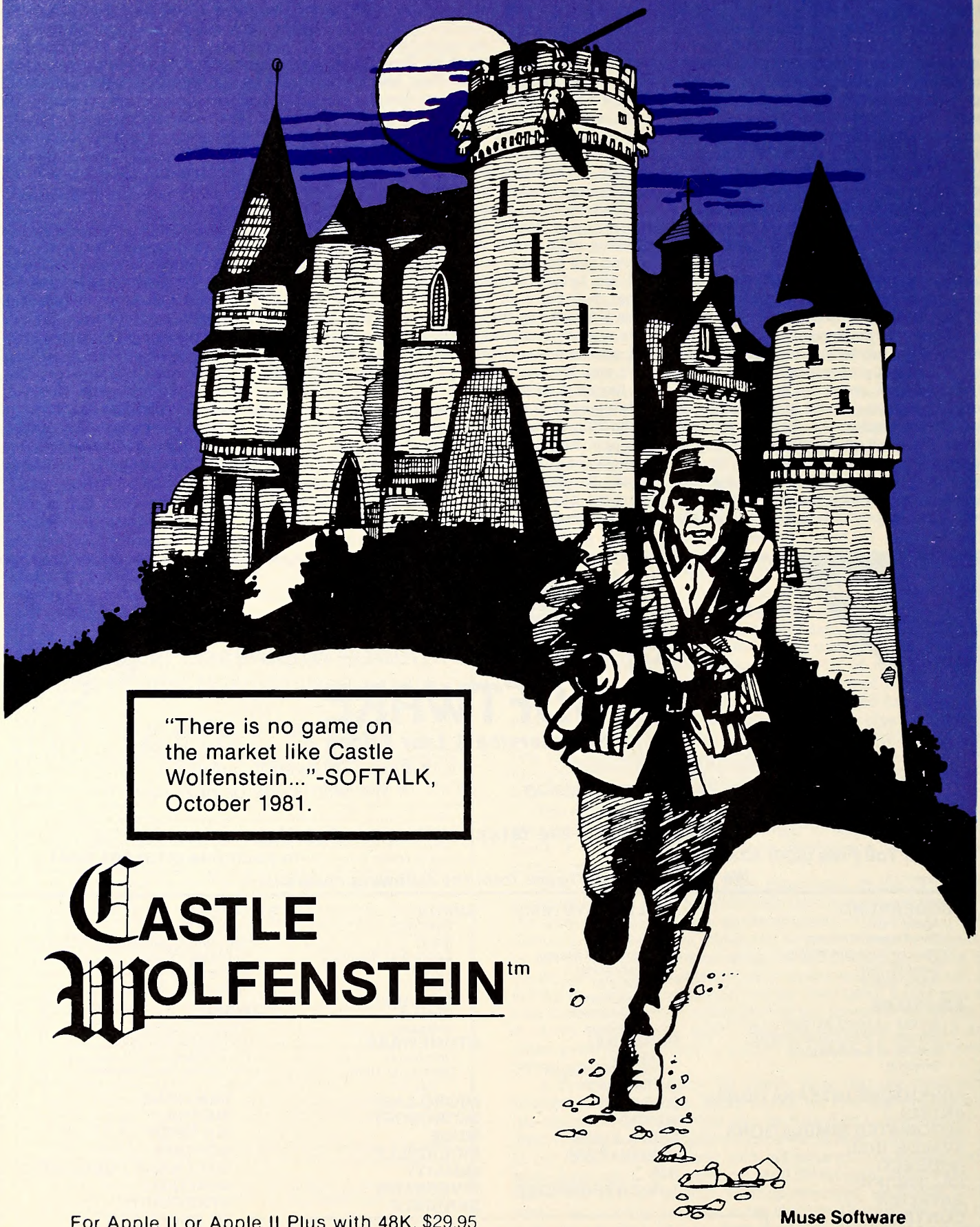
☐ January 15 is the deadline for sign-ups to the 20th Annual Conference of the **Urban and Regional Information Systems Association** to be held August 22-25 at the Hyatt Regency in Minneapolis, MN. Sponsors say URISA is the oldest and largest professional organization concerned with the effective use of information systems technology in the public sector. Individuals are invited to propose a paper, presentation, or panel that describes practical application of computers in government. Microcomputer applications and the implications of anticipated new technology are of particular interest. Abstracts and outlines must be submitted by the January 15 deadline to Thomas M. Palmerlee, 1982 Conference program chairman—URISA, 2033 M Street, N.W., Suite 300, Washington, D.C. 20036. Phone: (202) 466-7406.

☐ **Mitchell Kapor and Associates** (Cambridge, MA), whose *VisiPlot* and *VisiTrend/VisiPlot* made one of the largest sales in software history, has formed **Lotus Development Company,** a research and development organization. Lotus's first product is due for release early this year.

☐ **SSM Microcomputer** (San Jose, CA) has signed an agreement with **Personal Computer Ltd.** for distribution throughout the United Kingdom. The agreement covers all SSM products, including the recently introduced Transend data communications software series for the Apple II.

GOTO 171

MUSE[tm]
SOFTWARE



"There is no game on the market like Castle Wolfenstein..."-SOFTALK, October 1981.

CASTLE WOLFENSTEIN[tm]

For Apple II or Apple II Plus with 48K, $29.95

Muse Software
347 N. Charles Street
Baltimore, MD 21201
(301) 659-7212

THE THIRD BASIC

by

Taylor Pohlman

Exploring Business Basic, Part 5

Last month's column promised the answer to the question, How many bytes of memory are available in a 256K Apple III? As you know, the 256K Apple III has been announced and is beginning to be available, so the answer can now be revealed: 191,484 bytes! That's more than three times the workspace available in any other personal computer Basic. (Aren't you glad you've got an Apple III? Don't you wish everybody did?) We were discussing some sorting techniques for our database that can make good use of that space. This time we'll explore a mixed bag of items, deferring our discussion of the *print using* capabilities of Business Basic until next time.

Our Mixed Bag. The first bagged item this month is the mailbag. Several questions have come my way since this series started in September; the most interesting ones have to do with programming style and philosophy. The most intriguing question concerned why I always use lower case variable names in my programs, especially since the Basic keywords (like *print*) all seem to be in caps. While it would be easy to say that I lack the strength or will to operate the alpha lock key, the real reason has to do with the way Basic itself works.

As you probably know, Business Basic defers its syntax checking (looking for errors) until you actually run the program. Basic does perform some tasks as each statement is entered, however; the process is generally referred to as *tokenizing*. Simply stated, this means that Basic scans each statement and converts each keyword, sometimes called a *reserved word,* into a special internal one-byte code called a *token.* This code not only saves space, but also simplifies error checking and program execution.

Almost all Basic interpreters use the tokenizing technique. One of the consequences of this method is that program statements cannot be listed out without the Basic *list* command converting these tokens back to their English-language equivalents. In converting the tokens, Basic always prints out the upper-case version of the keywords. I type in all Basic statements—both variables and keywords—in lower case so that when I list out a program, I can see what Basic interpreted as keywords. If I misspell *print*, Basic will not recognize it as a keyword, and the fact that it remains lower case makes such an error easy to spot in a listing. In addition, Business Basic requires spaces between keywords and variable names, to allow variables to contain keywords themselves.

Ever try to use a variable like *Orange* in Applesoft, only to discover that *or* is a reserved word (and therefore your variable must be renamed to something like *rnge*)? Typing variable names in lower case will allow you to spot those times you forgot to space and ended up with "fori=1 TO 10" instead of "FOR i=1 TO 10". The first instance will produce an error, since Basic will assume you are trying to assign the value of 1 to the variable *fori* and for some reason put the phrase "TO 10" onto the end of the statement. Some examples will clarify:

Typing:	10 prunt x*53	will result in: 10 pruntx*53
whereos:	10 print x*53	will result in: 10 PRINT x*53
Typing:	10 on xgoto 20, 40,50	will result in: 10 ON xgoto20,40,50
whereos:	10 on x goto 20, 40,50	will result in: 10 ON x GOTO 20,40,50

See how much easier it is to catch the error when it's displayed visually?

As with every rule, there are exceptions. Any variable that starts with the letters "FN" will be assumed to be a function name. Again, typing all lower case will help you spot the problem:

Typing:	10 xval=aval* fnumber	will result in: 10 xval=aval* FNumber

and you'll immediately know something's wrong (assuming that you really wanted to use *fnumber* as a variable name).

There's another little quirk in Basic that this technique helped me spot. As you may know, we've used the on *eof#* statement quite a bit to take action if a program tries to read past the end of file. According to the manual, the part following *eof#n* can be any executable statement. So far, we've generally used *goto* or *gosub* statements to take action.

Consider the following:

Typing: 10 on eof#1 goto 20

will result in: 10 ON EOF#1 GOTO 20

as you'd expect. But:

Typing: 10 on eof#1 xvol=20

will result in: 10 ON EOF#xvol=20

For some reason Basic treats the whole thing as one variable. The solution involves dredging up a bit of Basic folklore. Remember in your first class in Basic when they told you that all assignment statements started with the keyword *let*? Most Basic dialects have long since made the *let* keyword optional, and most people have quit using it altogether. An example of the use of *let* is:

10 LET x=45 which is usuolly written simply: 10 x=45

If there's any ambiguity to the way a statement can be interpreted, *let* can be used to clear it up. With our new version of the *eof* statement:

typing: 10 on eof#1 let xval=20

will result in: 10 ON EOF#1 LET xvol=20

and everything works fine. The fact that Basic failed to upshift the reserved word *eof* in the example above is very important to an understanding of the problem.

The technique of entering everything in lower case has saved me countless hours of debugging my errors. I recommend it.

Bag Item Number Two. Last month's list of new goodies in Business Basic 1.1 completely overlooked one item which, while it may seem minor, has important consequences. The change is an extension to the standard *get* statement. Normally, as is the case in Applesoft and some other Basics, *get* allows reading the keyboard one character at a time, including all special control characters and delimiters. This means you can bypass control-C and return, read commas, and so on.

Business Basic 1.1 extends *get* to allow *get#n.* This means you can read any SOS file one character at a time, without respect to what kind of file it is. This can be very handy for reading all characters from the communications port (via the .RS232 driver) or for reading other character streams from special devices. One of its most interesting traits, however, is the fact that it can be used on disk files as well. Remember that one file is just like another in the SOS environment, so if we open a text file on disk, *get#* will allow us to read one character at a time from it.

This means that there's now an easy way to read text files that contain more than 255 characters without a return character. Normally a string overflow error results if you attempt to read such text files with the Basic *input* statement. Even more interesting is the fact that we can also open and read from the Basic data file. Remember that I described the data file as having special tags, called *type bytes,* that enable Basic to determine what data type is stored next in the file. Remember also that numeric data is stored in a data file in its binary form. *Get#* allows reading this binary information, one byte at a time. One example is worth a thousand explanations:

```
5     INPUT"File to dump: ";a$
10      IF a$="" THEN 100
15      OPEN#1,a$
20      ON EOF#1 GOTO 100
25      cr$=CHR$(13)
30      GET#1;a$
40      IF a$=cr$ THEN PRINT
50      PRINT a$;
70      GOTO 30
100       CLOSE
110       END
```

This simple example will dump any text file to the screen, no matter how long the intervals between carriage returns. A good example of a text file with arbitrarily long strings is the file I'm creating now, using *Applewriter III.* Return characters are inserted only at the end of paragraphs which, as you'll notice, tend to run on indefinitely.

Note that this program looks for return characters by loading the variable *cr$* with a return (decimal 13) and then testing for it before printing. If you wanted to reconstruct strings from the file, you could do so by using a string variable to accumulate characters, stopping when a return was encountered. You'd need to test to be sure you hadn't overflowed the 255 character limit.

This program has one serious deficiency, however. Printing arbitrary characters from a file (especially a data file) can have weird consequences when the output device is the console, as it is in the example program. The console uses lots of different control sequences to perform functions, including setting windows and changing from black and white to color text modes. Also, a byte can contain 256 different characters, and the ASCII character set defines only 128. Clearly, we need a safe and consistent way to display any byte readable from a file. So, like most programs that start out short and simple, this last one's about to get complex:

```
5     INPUT"File to dump: ";a$
10      IF a$="" THEN 95
15      OPEN#1,a$
20      INPUT"File for output: ";a$
25      OPEN#2,a$
30      ON EOF#1 LET eof.occurred=1:GOTO 80
35      bytes=0:eof.occurred=0
40      line$=""
45      PRINT#2;HEX$(bytes);"-";HEX$(bytes+31);" ";
50      FOR i=1 TO 32
55        GET#1;a$
57        val=ASC(a$):IF val>127 THEN
          val=val-128
60        IF val<32 THEN line$=line$+"
          .":ELSE:line$=line$+""+CHR$(val)
65        outhex$=HEX$(ASC(a$))
70        PRINT#2;MID$(outhex$,3,2);
75        NEXT i
80      PRINT#2:PRINT#2,"        ";line$
85      bytes=bytes+32
90      IF eof.occurred=0 THEN 40
95      CLOSE
120       END
```

As you scan through the program, note that in addition to opening the file to be dumped, we open a second file to which the output is written. This gives us more flexibility, and still allows us to use .console to see the output on the screen. Line 30 sets up our end-of-file condition, using the *let* statement to get around the problem we described earlier, and demonstrates one other handy thing. We can embed periods in variable names to improve readability. It's obvious that *eof.occurred* is easier to interpret than *eofoccurred,* and this is especially true for more complex variable names (remember that Business Basic permits 64-character names).

Lines 35 and 40 initialize variables. We will be using the *line$* string to accumulate the characters read from the file for later printing. After each line of *print* we will reinitialize the string. Since we'll be printing thirty-two characters at a time from the file, line 45 uses the *hex$* function to set up the labels for each line.

A note about hex is appropriate here. Hex stands for hexadecimal, or base-16, arithmetic. Since any hex digit can be represented by four binary bits and a byte can be exactly represented by two hex digits, it is convenient to use hexadecimal numbering in many aspects of computing. It is preferred over decimal and octal notation and is, of course, much more compact than binary. What usually throws people is that to represent all values between 0 and 15 with a single digit, hex uses the numerals 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, respectively. F thus is equivalent to decimal 15, and 1F to decimal 31 (the 1 is in the sixteens place).

We won't try to offer an in-depth explanation of hexadecimal notation here. If you aren't familiar with it, any beginning text on computers usually covers the subject thoroughly, and readers of Roger Wagner's column in this magazine have been inundated with help on hex. Suffice it to say that the *hex$* function will convert any reasonable numeric quantity into a four-byte string of hex digits.

Getting back to our program, the loop from line 50 to line 75 is the main one where we dump thirty-two bytes at a time in hex format, while providing character representations for those within the displayable range (hex 20 to 7F, decimal 32 to 127). The back of your Basic manual contains an ASCII code chart that will help you follow along with the decoding. Line 57 in the program sets the vari-

```
0000-001F 2E636A0D5420482045202054204820492042044202042204120532049204300
          .cj.T H E   T H I R D   B A S I C
0020-003F 0D6279205461796C6F7220506F686C6D616E0D0D0D2E6C6A0D4578706C6F7269
          .by Taylor Pohlman....lj.Explori
0040-005F 6E6720427573696E65737320426173696320506172742066697665500D0D4C
          ng Business Basic - Part five..L
0060-007F 617374207469D65204920647270706564207365766572616C2062726F6164
          ast time I dropped several broad
0080-009F 2068696E747320616275757420657720736F667477617265206164206861
           hints about new software and ha
00A0-00Bf 726477617265206861707065E696E6773206F6E207468652041707065202F
          rdware happenings on the Apple /
00C0-00DF 2F2F2E2020486F706566756C6C792062792062E6F7720796F7520686176652068
          //.  Hopefully by now you have h
00E0-00FF 6164206120636861E6365207046F20676F20646F776E207046F20796F75722064
          ad a chance to go down to your d
```

Figure 1.

able *val* to the ASCII value of the byte just read, and then an *if* statement checks to see if the value is in the 128 to 255 range. If so, 128 is subtracted from the original value to bring it within the normal ASCII range.

Line 60 checks to see if the resulting character is a control character, and if so, represents it as a period in *line$* to signify that it is unprintable. Otherwise, the character representation is stored. The characters are right justified in each two-byte cell, because they'll be printed below the hex values. Next, the hex value of the original character is assigned to *out-hex$* in line 65, and printed to the output file in line 70. Since we want only the rightmost two hex digits, the *mid$* function is used. After the loop prints out the thirty-two values, lines 80–90 print the ASCII equivalents stored in *line$*, bump the byte count, check for *eof* condition, and repeat the sequence.

Figure 1 shows how the output from this little jewel looks when run against the file for the first draft of this article.

Messy, huh? Let's look more closely at the output to see if it makes sense. The first line tells us we are looking at bytes 00 through 1F (0 to 31 decimal), and the top line is the hex representation of the characters, two digits per character. The first character in the file is 2E in hex, which happens to be a period. Notice that 2E is the character printed below on the next line. The next two characters in the file are 63 and 6A which correspond to the ASCII characters c and j. This is understandable, since *Applewriter III* uses the print format command .cj for center-justify, which is what I wanted done with the title. The next character is OD which translates to decimal 13, or a return character. Note that a period is substituted for this character on the print line, since return is in the control character range. And so on, and so on. Practice on a few text files of your own and get a feel for reading the notation.

It really gets interesting when we begin reading files whose exact format is normally pretty obscure. Data files are an excellent example since, although the *read#* statement can get data out, things like the type bytes and string-length bytes are normally inaccessible. To see how our dump program would work on a data file, we need a way to generate an interesting file at which to look. The following simple program will do the trick. When we get serious later on about sorting techniques, we'll need such a program, so I'll introduce it now:

```
 5   OPEN#1,"junkfile",30
 6   INPUT"Number of records to create: ";n
 10    FOR i=i TO n
 12      i%=RND(1)*10000
 13      WRITE#1,i;i%:PRINT i%,
 15      a$=""
 20      FOR j=1 TO 5
 30        a$=a$+CHR$(65+INT(6*RND(1)))
 35        NEXT j
 41      FOR K=1 TO 4
 42        a$=a$+CHR$(48+INT(10*RND(1)))
 43        NEXT k
 45      WRITE#1;a$:PRINT a$,
 48      val=RND(1)*1E10:WRITE#1;val:PRINT val,
 49      i&=CONV&(RND(1)*1E15)
 50      WRITE#1;i&:PRINT i&
 55      NEXT i
 60    CLOSE
 70    END
```

This program will create a random access data file of arbitrary length containing an integer, a string, a real, and a long integer in each record. What's noteworthy here are the two small loops that build the string value. They're set up in such a way as to insure that the first five characters are upper-case alpha and the next four are decimal digits. Type the program in now and run it to create a small file, say five records. Although each run will differ, the output should look something like this:

```
2092 CEEBE4542   7.72055E+09  930904428626944
7107 CDCAD1031   6.87212E+09  971614244086784
9206 DDADE8239   6.94853E+08  839965717072896
3038 ADBAC4450   6.09472E+09  397952126404096
3814 AABED9057   2.27867E+09  768212125296640
```

```
0000-001F 00000000000000000000000000000000000000000000000000000000120B
          . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
0020-003F 2C2109434545424534353432 14A1661722180003 4EA713C9C400000121BC321
          , ! . C E E B E 4 5 4 2 . ! f . " . . . N ' . I D . . . . . C !
0040-005F 0943444341443130333114A14CCE1D18000373AD91E0A40000001223F62109 44
          . C D C A D 1 0 3 1 . ! K N . . . . s - . ' $ . . . . # v : . D
0060-007F 4441444538323339149E25AA7B180002FBF1C2308C000000120BDE2109414442
          D A D E 8 2 3 9 . . % *  . . .  q B 0 . . . . . ! . A D B
0080-009F 4143343435301 4A135A30818000169EF7321C60000001 20EE62109414142454 4
          A C 4 4 5 0 . ! 5 # . . . . i o s ! F . . . . . f ! . A A B E D
00A0-00BF 393035371 4A007D1D0180002BAAF52D728000000
          9 0 5 7 .  . Q P . . . : / R W ( . . .
```

Figure 2.

Now for the fun. When you run your dump program against the file this program creates, the output should look something like figure 2.

Well, nobody said computer science was for the faint of heart! By the way, the term generally used to refer to this type of listing of file contents is *formatted dump*. Formatted, because we have organized the information in the printout, and dump, because it is a nonselective output of the exact contents of the file.

Now the fun begins. The first thing to notice is that almost the entire first line is composed of zeros. Remember that although our dump program starts at the beginning of the file, the program we used to create this file began at record 1. Since the record size was thirty bytes, we would expect to find an empty record of thirty bytes at the beginning, and that's exactly what the dump shows. This means that the hex 12 in byte 31 of the file must be at the beginning of record 1. Now something that was mentioned earlier about type bytes in data files becomes important. Remember that the general format of a data file is:

Type byte	Data bytes (2,4 or 8)

for numeric values (integer, real and long integer), and:

Type byte	Length byte	Data bytes (0 to 255)

for strings. This information should enable us to decode the information in this dump.

Since the first value in the record was an integer, the hex code 12 must be the type byte for integer data. Following our format, this means the next two bytes (hex codes 08 and 2C) must be the binary integer value. Evaluating the hex value 082C yields decimal value 2092, exactly the value our printout led us to expect.

The next value in the file is a string, which contained CEEBE4542. Referring again to our format for strings in data files, we'd expect the next file byte to be the type byte. That's the hex code 21. Next is the length byte, which, since the string is nine characters long, should be equal to 9. That's hex code 09, one of those lucky hex numbers that is the same as its decimal equivalent. After that, our format line shows that indeed, the string value is CEEBE4542.

The next value in the record was real. Since the next byte after the string should be the type byte for reals, we can conclude that the hex 14 found in position 2C (44 in decimal) is the floating-point type byte. Floating-point numbers are stored in a thirty-two-bit internal format in Business Basic, so we would expect the next four bytes to contain the binary value. Proving that this value (hex A1661722) is equal to 7.72055E+09 is a considerably more complex task and will be left to the numerically inclined reader. That phrase "left to the numerically inclined reader" is this author's equivalent to the famous line found in all math texts—"it can easily be shown that . . ."—and is just as big a cop-out.

The last value in the record is a long integer, and the type byte in position 31 (decimal 49) has the value of hex 18. Long integers are stored as eight byte quantities, therefore the next sixteen hex digits should represent the number. Since that hex value is 00034EA713C9C400, it follows that converting this value should yield the decimal value originally printed out: 930904428626944.

As a little added bonus in this article, let me offer a program that demonstrates the truth of the preceding statement. This program will convert any reasonable hex value into decimal and print it out rather quickly, using the long-integer data type and Business Basic's conversion functions. Forthwith, it is:

```
5     sixteen&=16
10      INPUT"hex value: ";a$
15      IF a$="" THEN 100
20      cum&=0
25      mult&=1
30      FOR i=LEN(a$) TO STEP-1
35          val&=CONV&(TEN(MID$(a$;1,)))
40          digit&=mult&*val&
45          cum&=cum&+digit&
50          mult&=mult&*sixteen&
55          NEXT i
60      PRINT cum&
65      GOTO 10
100       END
```

The program simply brute forces the problem, one digit at a time, but since the long integer arithmetic is very fast, program performance is quite reasonable. This program knows nothing about sign bits, though, so it will fail to convert negative integers expressed as hex constants. A fix for this limitation would be to check for the high-order bit and negate the final result, but the program would then lose its general nature. Anyway, it's free.

Well, that got us completely off track. Going back for a second to the formatted dump, we are now at position 3A hex (58 decimal), which is really position 28 decimal in this record. The remaining two bytes of the record (remember that we declared the record to be thirty bytes long) should be empty; sure enough, they show up here as zeros. This gets us to position 3C, the beginning of the next record, and there we find the integer type byte 12, signaling that we can start the whole process again. I leave that to you if you want to try your hand at decoding. Some of what we have learned can be summarized in the following table:

Data type name	TYP() function value	Internal hex	file code decimal
Integer	2	12	18
Real	1	14	20
Long Integer	3	18	24
String	4	21	33

Don't forget that the *get#* statement can be used in lots of other interesting ways and that its primary function is to process console input effectively without those characters being first processed by Basic. I just thought the examples above would give us a chance to explore several interesting topics at once.

Final Thoughts (Bottom of the Bag). I'd fully intended to explore one more topic that had previously generated questions, but this tome grows overlong. The topic I had in mind was the use of the *request* invokable module. Those of you who are writing programs that do lots of reading and writing of numeric arrays to disk should tune in next time when we show how to get at least twenty times the performance improvement over using *for—next* loops to accomplish the same task. That, combined with the huge memory space available for arrays, provides some significant capability to the person interested in data analysis and sophisticated file indexing.

I also promise to get to my thesis on *print using,* especially since Business Basic allows some tricks not available in most other Basics. One of these days we'll get to graphics as well, and discuss how to use *bgraf* and *download* to create some really interesting stuff.

Until then, just one last note. I looked back over this article and decided that the word *hex* was mentioned so many times that we have left the era of "voodoo economics" and entered a new era of "voodoo Basic". Oh well. Maybe if I wore garlic while typing . . . .

ARTSCI explains why some word processing systems are better than others.

Let's begin with an easy to understand explanation of what a word processor is and how ARTSCI has created a professional system.

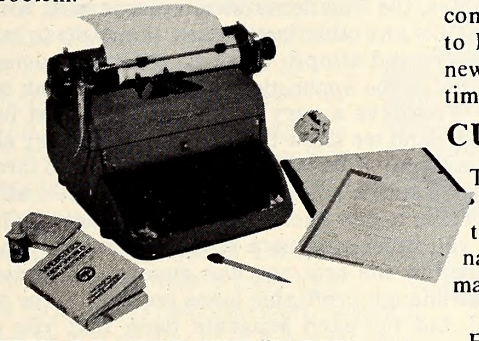
A word processing system is simply an easier, faster and less expensive way to type. With a modern word processor, documents are entered on a video screen instead of paper.

You can enter your first rough draft without concern about errors or spelling. Simply go back and insert letters, delete words and even move paragraphs with a few keystrokes. No document will ever have to be retyped.

WORD PROCESSING AND THE APPLE II

The APPLE II is the most expandable, inexpensive micro-computer available today. It can perform almost any task, including word processing.

The standard APPLE II however, uses a 40 column video display. This display causes a serious word processing problem: How do you display a full sized 80 column letter? Most word processing programs available today do not solve this problem.

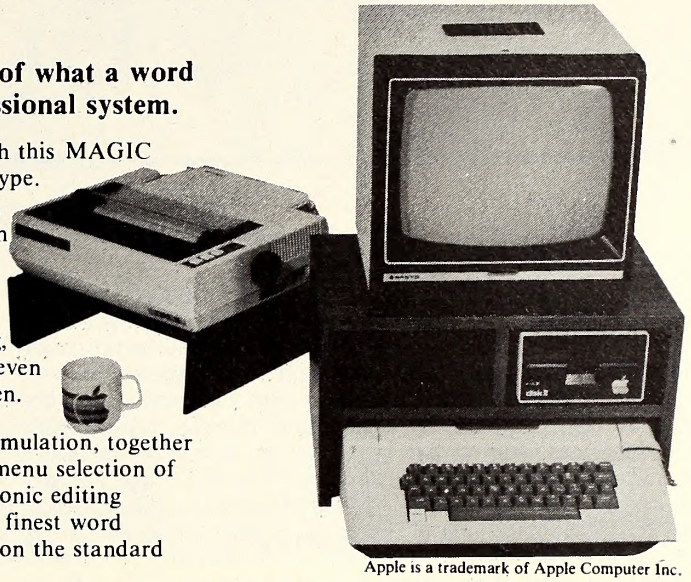


THE MAGIC WINDOW

ARTSCI has developed the MAGIC WINDOW word processing system that incorporates the full power of a professional word processor and solves the APPLE'S display problem without expensive hardware.

The first feature of a professional word processing system is the ability to enter and edit data in a fast and friendly manner. The MAGIC WINDOW operates just like a standard typewriter. The electronic paper moves to the left across the video screen as you type. Almost any size document can be represented on the video screen. You can see the edges of the paper through this MAGIC WINDOW as you type.

The rule is: What you see on the screen is what you'll get in print. However, if you print using proportional spacing, the result will look even better than the screen.



Apple is a trademark of Apple Computer Inc.

This typewriter simulation, together with simple to use menu selection of functions and electronic editing abilities, creates the finest word processor available on the standard APPLE II.

MAGIC SPELL

The second feature of an advanced word processor is the ability to find and correct mistakes. The most common mistakes in most documents is the misspelled word.

ARTSCI's MAGIC SPELL program will take any document you can create and find spelling errors. Over 14,000 commonly misspelled words are known to MAGIC SPELL. You can also add new words to the vocabulary at any time.

CUSTOM LETTERS

The third feature of a professional word processing system is the ability to alter a document by replacing names and other related data from mailing lists.

Form letters, invoices, and almost any document can be individualized by replacing names, addresses or any other personal data anywhere in the document using ARTSCI'S BASIC MAILER.

By using the BASIC MAILER you can take any mailing list and sort through the list by different criteria and print personalized letters with a few simple keystrokes.

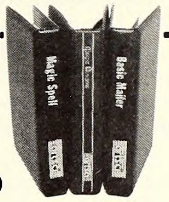
A COMPLETE SYSTEM

These three programs, THE MAGIC WINDOW, MAGIC SPELL AND THE BASIC MAILER, together form the only complete and professional word processing system available on the standard APPLE II computer system.

ARTSCI TAKES THE WORK OUT OF WORD PROCESSING

For a more thorough explanation of the ARTSCI word processing system send for our free booklet.

artsci INC



Mail to:

ARTSCI INC.
10432 Burbank Blvd.
North Hollywood,
California 91601-9990
(213) 985-2922

Name _____

Company _____

Address _____

City _____ State _____ Zip ____

Phone _____

ST PE MI CC BY CA

VENTURES WITH VISICALC

BY CHRISTOPHER U. LIGHT

Eight years isn't a very long time in most industries. Steel mills still use the same type of blast furnaces they had eight years ago. Banks still have tellers to conduct personal transactions, and these tellers still have long lines at their windows. But, as you know, eight years is a couple of generations in the computer business. In this field, eight months can produce revolutions and eight weeks an awful lot of change. In fact, I find that if I don't stop in at my local computer store every eight days or so, I may miss a piece of software I need. And I know I'll miss a new game my teenage boy would like to try out.

Eight years ago I was hired to design and program a financial modeling package for a large bank holding company that would help answer the sort of "What if" questions that are now so easily analyzed with *VisiCalc*. Of course I didn't come up with a catchy name like *VisiCalc* (I wish I had) but, with the help of a number of the bank's vice presidents, its entire research department, a dozen keypunch operators, systems analysts, computer operators, and innumerable cups of black coffee, I finally produced the "Southern National Bank" Financial Forecasting Model (Version 97.3 or something like that).

It was a nightmare. But it ran. And it only took about half an hour of my time—generally around midnight—plus maybe a quarter of an hour of I.B.M.'s biggest and best number-cruncher *each* time the bank wanted to look at one possible policy change, in other words each time we changed the number in one cell.

"Southern Bank Holding Company" (SBHC, for short) was the parent holding company that owned the "Southern National Bank" (both names are disgulsed). Its top management had had a very critical policy decision to make back in 1974. It was located in a state that permitted an almost unlimited number of bank branches within a bank's home community (the joke was that the state had a bank on every corner that didn't already have a gas station) and, using a multibank holding company to own other complete banks, allowed banks to expand into other communities across the state. In 1974 there was a branching war in the state with the twenty or so largest banks starting new banks, taking over smaller ones in mergers and building new branches just as fast as they could to prevent their competitors from getting those key corners that didn't already have gas stations on them. It was eat or be eaten, and the presidents of all but the largest banks, which were busy taking over or moving in on the smaller banks, were running scared. Later, when OPEC-financed Eurodollars began flooding in, heads rolled at a few of even the biggest banks as the new Saudi and Kuwaiti owners hired their own management teams. Perhaps it wasn't the worst of times to be a banker, but it certainly wasn't the best.

In its quest to put one of its own banks in every town in its region of the state and a branch on every vacant corner of that town, SBHC had borrowed several millions of dollars from one of the giants of the banking industry, dollars which it was using as capital to start up its new banks and branches. Interest on this loan had to be paid each year, and the loan itself was to be repaid, presumably out of the profits of the new banks, at the end of ten years.

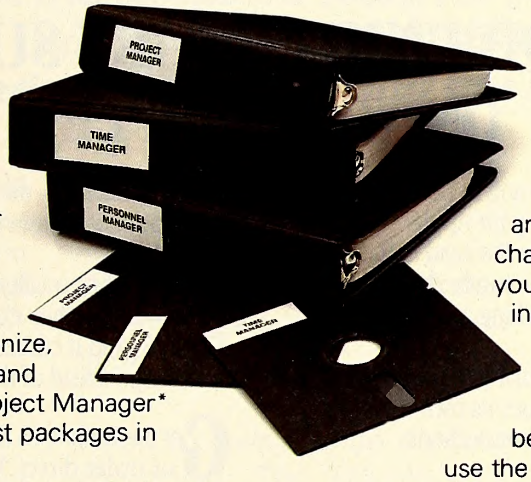
The bank's announced goal was to start or acquire by merger three complete banks a year and about as many branches as it could find vacant lots to put them on. Its board of directors approved, and its stockholders wanted the growth. But a brand new bank is an expensive business to create—costing in the millions—and even a new branch can require an investment in the hundreds of thousands of dollars. Furthermore, banking is a business that is tightly regulated by the federal and state governments. Because no one wants to chance another banking system collapse like the one that triggered the Great Depression of the 1930s, the government makes sure that banks aren't allowed to take risks so great that they might go under—taking their depositors' money with them.

And so, each time that SBHC would apply to the Federal Reserve System for a permit to start another bank, which it was then doing every few months, the Fed would ask, "How will you pay back that multimillion dollar loan?" And SBHC would reply, "Out of the profits from these new banks." For a while the Fed went along with this off-the-cuff forecast. Then, when SBHC had seven banks and a few nonbanking financial service companies, the Fed demanded a closer look. Knowing that new banks, like any other businesses, take time to build up a list of customers and simply won't be profitable overnight, the Fed responded to the application for the eighth bank by saying, in effect, "You have a very large loan that must be paid back in ten years, and we know that you will be in very serious trouble if you can't pay it back then. Give us detailed forecasts showing how your entire holding company complex will generate enough profits to pay back that loan. Because you have a ten-year loan, you must give us a ten-year forecast; in addition, we will need to see how you can generate sufficient deposits and make enough profitable loans not only at the parent company level, but for each separate bank that you own." In other words, somebody at the Fed had realized from SBHC's financial statements that it was expanding so fast that it might run out of money before the new profits came in, and wanted someone on the bank to do all the "What if" analysis necessary to show just how many new banks and branches could be added a year without spending the company into bankruptcy.

Today, the cry would go out, *"VisiCalc* to the Rescue," and a member of the bank's research department would turn to his Apple and begin asking those questions. But only eight years ago not only was there no *VisiCalc*, but Apples still grew on trees. Instead SBHC had to hire someone to design a *VisiCalc*-type of program that would run on the large, mainframe I.B.M. equipment it used for its own accounting. The choice was very clear: either hire a consultant to design a financial planning model at the bank's own expense or stop expanding.

It was a cushy assignment. I was given an office in the bank's tower with a view overlooking half the city, an office that had just been vacated by one of the bank's directors. I reported directly to the parent company's executive vice president. I was given all the secretarial support I needed and, be-

The Manager Series* from Microsoft™ turns a personal computer into an executive toolbox.



Better management tools. The Manager Series from Microsoft turns an inexpensive personal computer into an executive's toolbox. Not a computer programmer's toolbox. An executive toolbox. Computerized management tools for non-computer people.

Time, people, projects. The Series is a system of software tools that work together to help you plan, organize, schedule and record your business and personal affairs. Time Manager,* Project Manager* and Personnel Manager* are the first packages in the Series.

Write it once. All programs in the Manager Series allow you to transfer information between programs. That means you can enter information in one program and transfer it for management by another.

Time Manager. The key. Time Manager helps you manage your personal time, appointments and priorities. It can also help you manage expenses, costs and job schedules. Or, keep a running tally of costs and hours by day, week, month or year. And Time Manager can act as an "executive" to manage other programs in the Series.

Project Manager. Describe the components of a project to Project Manager. It will create timing, task and resource charts to help you focus on critical tasks. Change one piece of information and Project Manager will recalculate the entire project. Project Manager even flags overcommitted personnel resources.

Personnel Manager. Manage information about people, companies, customers or prospects. From names and addresses to skills, position, and characteristics. Personnel Manager lets you enter any kind of people-related information. Then, organize and retrieve it almost any way you want.

Management software. Even if you've never used a computer before, you should be able to productively use the Manager Series in a very short time. And, when you've learned to use one in the Series, you've virtually learned them all.

Seeing is believing. Ask your local computer store for a demonstration of the Manager Series. It's a series of management tools that could be your best reason to own a personal computer.

*Trademarks of The Image Producers, Inc.

MICROSOFT
CONSUMER PRODUCTS

A Division of Microsoft, Inc.
10700 Northup Way • Bellevue, WA 98004

lieve it or not, unlimited computer time. Today, when you can buy a fully equipped computer for your exclusive use for less than half the cost of a plain-Jane automobile, this last may not seem like much. But, back in the mid-1970s, unlimited computer time, which was billed out at a rate as high as $600 per hour, was something you might aspire to as your reward in heaven after an absolutely sin-free life. And, best of all, SBHC agreed to give me a two-month training program in top level bank management. I had been educated in model building and financial theory but knew almost nothing about banking, and the knowledge I gained has proved invaluable both in business and in the classroom. There was, of course, a deadline: six months. At the time it seemed more than ample, and it did turn out to be just adequate. I've often wondered how long it took the designers of *VisiCalc* to come up with their package, which is much more complex and versatile.

The training program consisted of moving from department to department observing how a large bank operates and, especially, meeting the people who had the information I would later need. I spent three days with the controller, perhaps four with the research staff, a coffee break with the president, a week in the computer center, and so on. In the evenings I read textbooks on bank accounting methods. The time went fast. At the end I didn't know enough to be able to run a bank, but I knew who to go to to ask how one should be run. And it was knowledge I needed. The hardest part of forecasting isn't figuring which equations to use to connect the cells in a *VisiCalc* worksheet. It's in the numbers that go into those equations. And the best source of some of those numbers will be the very executives responsible for them. but they'll give you honest estimates only after they've come to know and trust you.

So, with four months to go, it was time to begin building a model. One of the first steps, as is true even if you already have a *VisiCalc* and don't have to create your own from scratch, is to formulate very clearly the questions you want the model to answer and to pin down the sources of the data you will plug into your model, because the form of the model itself depends critically on these.

For example, suppose I were a complete outsider with no access to inside information, hired to answer the question, "Should I buy stock in SBHC?" Instead of estimating how much cash newly formed subsidiary banks will throw off over the next ten years, what I would like to do would be to forecast the stock price for my client. I can't forecast stock prices (and no other forecaster who's honest with himself will claim he can either). So I would suggest that we might try as a proxy question, "What are SBHC's earnings likely to be over the next decade?" If we can talk to management and find out its plans, great. If we can't, one approach that I might suggest would be to look at the overall economic outlook and try to relate this company's performance to it. Depending on how much my client wanted to spend on the forecast, that is, how much he could gain from it, such an approach could involve anything up to multi-equation econometric models of the global, national, and state economies or at least the use of those that were already available.

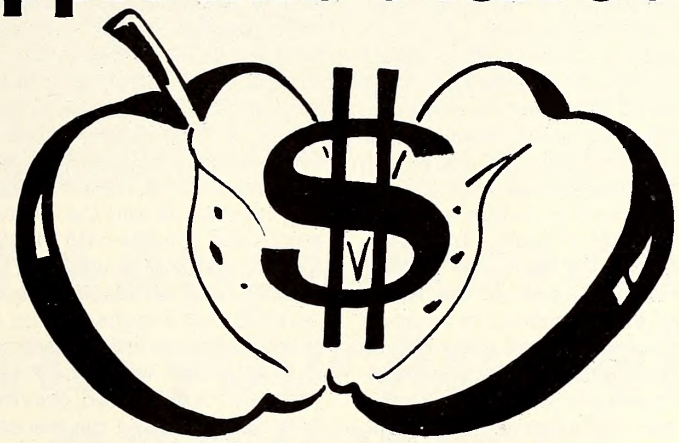
Clearly SBHC's cost and time constraints precluded the econometric approach. Besides, it would be like using a howitzer to kill a fly on the wall of your house; after the dust has cleared, all you can see is a fly buzzing around trying to decide which piece of rubble to land on. Clearly, too, a simulation model was the answer, and *VisiCalc*-type models shine in situations like the one I was in, where I could go to a particular vice president and say, "Let's sit down and figure out what number I should use for this year's home mortgage loans, and what rate of increase I should assume each year hereafter."

The only question was whether or not to make the model "stochastic," that is, whether or not we ought to allow for future uncertainty (and there's a whale of a lot of it when you're trying to go out ten years) by incorporating probability distributions for key variables or not. For example, we might believe that the interest rate on automobile loans, currently 18 percent a year, has a 50 percent probability of remaining un-

micro lab presents a

TAXSAVER

Buy the TAX·MANAGER and turn your apple into a tax deduction.



Let the Tax Laws work for you - buy the TAX·MANAGER and deduct the Apple and the Program

Use The Tax-Manager to help you prepare your taxes. Then include the cost of this program and part of the cost of your computer as a deduction.

You don't have to be an accountant to understand it. Simply enter your tax information into this easy-to-use program. Then sit back and relax while it quickly computes the information and prints most of your federal income tax schedules. Get The Tax-Manager now and start the new year with your taxes under control.

When tax laws change, don't be concerned. The Tax-Manager will never be outdated. With Micro Lab's Extended Warranty Plan ($30 a year) you can update to the current version at no additional cost. You will not have to reinvest more money in a new program. The Tax-Manager, which is the first of the Micro Lab Tax Saver group, will keep you current.

THE TAX-MANAGER IS NOW AVAILABLE AT THE INTRODUCTORY PRICE OF $150.

micro lab

systems that work

2310 Skokie Valley Road
Highland Park, IL 60035 • 312-433-7550

changed next year, a 10 percent probability of going to 20 percent, and a 40 percent chance of falling to 15 percent. We then roll the dice or consult a random number table to determine which figure to put into the model, the idea being that if we run the model many times under different assumed conditions, we will have used the correct interest rate on the average.

It's called Monte Carlo simulation, after the famous European gambling casino, and business schools love to teach it. It's also one of the reasons that Applesoft and the other versions of Basic have the *rnd* function; if you're doing Monte Carlo simulation, you can generate all the random numbers you need inside your program. The problem is that you must come up with a probability distribution for each of the significant variables; you must be able to say, for example, that there is a 20 percent probability that X will happen, a 40 percent probability that Y will happen, a 10 percent probability that it will be Z, and a 30 percent chance that what happens will be something entirely different.

My two-month training program had taught me that there was no way I could reliably estimate the probability distributions governing future interest rates, growth rates of the community, tellers' wages, the bank president's salary, and so on. In fact, a couple of years later I attended a seminar in Monte Carlo simulation. After the main speaker had spent half a day showing how to build a model, I asked him how he comes up with the actual probability distributions he needs whenever he wants to use his model on a real world problem. He smiled and said, "Oh, I leave that to my graduate students."

So, with my boss's approval, I decided to design a simple "What if" model in which we would insert figures that common sense said were reasonable. By varying them and running the model again and again, we reasoned, we could see which management decisions would be the most sensitive in terms of profits and cash available to repay that loan. Then we could examine more closely the assumptions behind the more sensitive decisions and, just using experience and judgment, have a pretty good idea of which runs of the model were likely to be the most realistic. It's a tried and true method that often gives as good results as the most sophisticated models, and it's interesting to note that, despite its theoretical usefulness and the love that professors of business have for it, the designers of *VisiCalc* also chose to eschew the Monte Carlo method. I am sure that it would have been possible to include the *rnd(x)* function in the list of *VisiCalc* functions, although at the cost of some memory, but it must not have seemed worth it. I think I would have made the same decision.

Another key consideration in the choice of model is the computer it's going to run on. Just imagine *VisiCalc* on a 4K machine. That didn't seem to be a problem in the bank's case, although it turned out to be one. The bank's computer center had three I.B.M. 360s, two model 50s and one model 40, and each had 256K of internal memory plus disk drives, tape drives, and, of course, those 132-column printers that gobble up a couple of reams of green-striped paper an hour. All input would have to be on cards punched laboriously on I.B.M. keypunches, but that also didn't seem like a burden at the time. Eight years ago I still thought the only use for a TV screen was to watch reruns of "Bonanza." The only on-line systems I'd ever seen produced teletype output printed on newsprint with Samuel F. B. Morse's own original ink ribbon.

The hardware available seemed fine. Then came the question of which language to use. The bank's computer programmers only knew Cobol, which is the language typically used for accounting. I only knew Fortran, both Versions II and IV, and a variant called Watfiv. Because the final printouts had to go to top executives and the board of directors, they had to be neat, organized reports with proper labels for each row and column. That constraint suggested using Cobol. My boss thought for maybe three seconds about keeping me on the payroll while I learned that language, and said, "We'll do it in *Fortran,* even though there won't be anybody who can help you."

I said that would be fine; it was only later that I realized just how much I had previously depended for help on having colleagues around me whose Fortran was better than mine. In its handling of mathematics, Fortran is very much like Basic, which was modeled on the older language. It uses the same arithmetic operators except that exponentiation is ** instead of the upward arrow. Its commands to print labels are complex in Version II, but not too dissimilar to Basic in Version IV. It cannot handle strings, but, unlike Applesoft, it allows its user to place his output anywhere on the page he wants, and it is this ability that allows one to produce good-looking business reports even though the language was designed for scientific analysis.

Fortran has a *format* statement that allows the programmer to take the entire 132 columns of I.B.M.'s printers and treat them as anything from a single field of 132 columns to 132 fields of one column each. I wish I could do this with Applesoft. The statement goes like this, for example:

```
10  FORMAT I3, 2X, 5(F4.2, 2X)
    WRITE (3) 10, X, Y, (Z(I), I = 1, 5)
```

This means *print* on printer number 3 the values of X, Y and Z(1) through Z(5) in the format shown in statement 10. This format is: an integer of three columns (I3), two blank spaces (2X) and five identical fields consisting of a four-column floating point number followed by two spaces (5(F4.2,2X)); the figure 4.2 means a number using four columns in total including the decimal point and rounded to two places after the decimal point—1.23. Labels can be included in the *format* statement in quotes, as they are in Applesoft.

It's a cumbersome system at best and is very hard to learn because a misplaced comma in a Fortran statement will cause a syntax error, and it might take many tries before a program will produce any output at all. Nevertheless, it does allow professional-appearing reports, which I wish I could do with Basic. *VisiCalc*, of course, provides menu-driven control of its format, which is easiest of all.

The error message caused by a misplaced comma might take five or ten minutes to figure out. In I.B.M.'s Fortran, the

TASC™ The Applesoft* Compiler. It turns your Apple into a power tool.

Step up to speed. TASC, the Applesoft Compiler, converts a standard Applesoft BASIC program into super-fast machine code. By increasing program execution speed up to 20 times, Microsoft gives you a power tool for Applesoft BASIC programming.

Highest capacity available. TASC will compile and run larger programs than any other Applesoft Compiler. As a disk-based system, it doesn't require the simultaneous presence of compiler and program in memory. The memory you save allows you to compile significantly bigger programs.

Power without bulk. Code expansion of up to 100% severely restricts other compilers. TASC's special code compression schemes typically limit code expansion to only 25%. You'll really appreciate that with complex programs or programs that utilize Apple's hi-res graphic pages.

More BASIC power. TASC's powerful new commands increase Applesoft BASIC programming capability. Chain with COMMON allows compiled programs to share variables, so a main menu supports several programs in a single runtime environment. TASC's True Integer Arithmetic and Integer FOR... NEXT capabilities maximize the execution speed of compiled programs. TASC's near total compatibility with Applesoft speeds compilation of existing programs with little or no modification.

What about mistakes? You perfect your programs interactively with Applesoft. If something does slip by, TASC recovers from errors discovered in compilation and traps all runtime errors. It even permits graceful interruptions during compilation.

See for yourself. Ask for a demonstration of TASC at your Microsoft dealer. Discover the software package that turns your Apple into a power tool.

*Applesoft is a trademark of Apple Computer, Inc.

MICROSOFT
CONSUMER PRODUCTS

A Division of Microsoft Inc.
10700 Northup Way • Bellevue, WA 98004

message would look something like, "IT332." You would then go to I.B.M.'s error message manual, which is about the size of a medium-sized city's phone book, look up the error message code and find it translated as "Syntax Error." There is a wonderful dialect of Fortran called Watfiv that makes program debugging almost a joy. Instead of printing SYNTAX ERROR and refusing to run, it lists all the lines in which errors occur and puts a symbol over the error itself. Unfortunately, SBHC's computer center didn't have a Watfiv compiler. Even more unfortunately, one hasn't been developed for Applesoft. Wouldn't it be loverly. . . ?

My daughter was recently in my office doing her high school computer course assignment on my Apple. She just sat at the desk and typed out a forty or fifty line program from some notes she had made, flipped on the printer, listed it, ran it, and printed it all at once. I said, "Don't you want to run it every few lines to check for errors?" She replied, "Dad, I don't make errors." She was right. It ran correctly the first time. But you can't do that in Fortran.

In addition, as I mentioned, Fortran was designed when TVs were black and white instead of green and carried mostly reruns of "Bonanza." Since a user couldn't interact directly with a computer anyway, there was no need for instant conversion of Fortran code into machine language. So, instead of interpreting a program line into machine language and running it immediately, all Fortran programs have to be compiled first. As those of you who have seen the ads for the new Applesoft compilers know, a compiled program runs at several times the speed of one that has to be converted line by line as it's running. But they're terrible to debug because you must compile, correct the first error, recompile, correct the second error, compile again, and so on.

Also the batch processing on an I.B.M. 360 requires cumbersome control cards for the input of instructions that are simply automatic on an Apple. To begin with, if the previous program failed to contain an End Of Job card (I believe the symbol was /*), the computer would treat your program as part of that one, and both would bomb. To ensure against that, some of us would stick a handful of End Of Job cards at the beginning of our programs and let the accountants worry later about the number of jobs that had actually run. Instead of just turning on a switch, we had to have as our first card a Job card that handled the accounting and contained our name, account number, and some authorization code that told the computer we had permission to use it. Instead of having Applesoft ready and waiting when that switch was turned on, we had to put in another control card that told the 360 to load its Fortran compiler (usually from disk or tape, although one installation that I used kept its language compilers on cards—one "fold, spindle or mutilate" and your entire system is down). The next card contained the instructions to load the program. Then we inserted the deck of cards containing the program. This was followed by another card that announced that the next cards would be data cards followed by a data deck. Somewhere or other there would be a control card telling the computer to run the program it had compiled. These control cards, which had to be absolutely correct in each column, looked something like //FORTG.GO,EXECFORT, or whatever. Unfortunately, the control codes vary from computer center to computer center depending on what version of DOS is being used. I long ago bought an Apple and burned my I.B.M. 360 manuals and don't remember exactly what the codes were, but you get the point. We didn't just type "RUN HELLO" and have everything happen magically.

So, using what was then the workhorse, mainframe computer of the world, I was ready to begin developing for the Southern Bank Holding Company its very own "*VisiCalc*." It seemed like a piece of cake. I was young and naive.

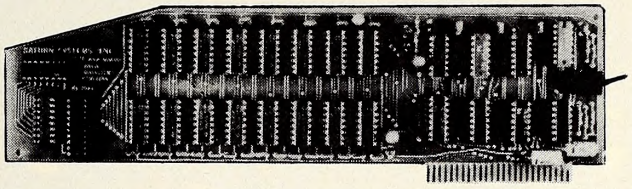
A typical day began when I got to my office in the tower about 8:30 or 9 a.m., spent a minute or two admiring the view and then turned to the SBHC's internal management reports. While a theoretically elegant forecasting system can be designed from scratch, it's simpler to use one that was built around the type of data it will have to use. So, working alternately with a Hewlett-Packard HP-80 calculator to get a rough idea of what the results might be like and some Fortran coding forms I had brought from home, I began working on the individual bank portion of the model. To save memory, it was decided early to do only one subprogram for all banks. This would read the data for Bank 1, perform the calculations, print it, sum it with the other banks in an accumulator and go on to Bank 2, and so forth. Because I had two spare columns on a key data card, the program could handle ninety-nine banks. If there had been only one column available, it would have been limited to nine.

After a week or so, I decided I'd better see whether or not what I had written would run. I made the first of what would turn out to be perhaps one hundred trips to SBHC's computer center out in the suburbs and handed my coding sheets to a keypunch operator who had been assigned to help me. The first thing I discovered was that the slash, /, goes through the zero, not through the letter O. I had come from a university research environment, where the computer center had to handle tremendous amounts of numerical data and very little alphabetical. To avoid writing a slash through one out of every ten characters, everyone there was taught that the letter got the slash. Well, a statement number that has the letter O in it bombs the program in Fortran just as it does in Applesoft, and when you're using batch processing with cards, you don't even get that message, "IT332," or "SYNTAX ERROR," or "TILT," or whatever your machine uses, until the next day.

The next day, still cheerful, my keypunch operator had to repunch every card she'd done for me the day before. She didn't complain the first time, or the second, or even the third. Then I ran out of the coding forms I had saved since my college days and discovered that SBHC had none. That was OK; I often used ordinary notebook paper anyway. Unfortunately my keypunch operator didn't know that each I.B.M. Fortran card must have the statement number in columns one through five,

SATURN SYSTEMS 32K RAM BOARD FOR APPLE

32K of buffered read/write memory on a plug-in card



Compatible with: Apple II®, Apple II+®, Microsoft's Z80 Softcard®, DOS 3.2, DOS 3.3, INTEGER Basic®, Applesoft®, PASCAL, FORTRAN, LISA®, Personal Software's VISICALC®

Software included:

1. Relocation of DOS into SATURN 32K board (recovers appproximately 10K of main board RAM).
2. Utility package for saving and loading Applesoft® and INTEGER® programs and data on the 32K RAM board; overlaying, chaining.
3. PSEUDO-DISK: Modifies DOS 3.3 to allow use of SATURN 32K RAM board(s) like another disk drive.

COMPREHENSIVE DOCUMENTATION • 1 YEAR WARRANTY

ALL FOR ONLY $239.00

NEW!

MEMORY EXPANSION SYSTEM FOR VISICALC®

Now you can expand the memory available to Personal Software's 16 sector VISICALC® using the SATURN 32K RAM BOARD!

With VC-EXPAND™ and one or more SATURN 32K RAM BOARDS the memory available to VISICALC® is increased from 18K to:

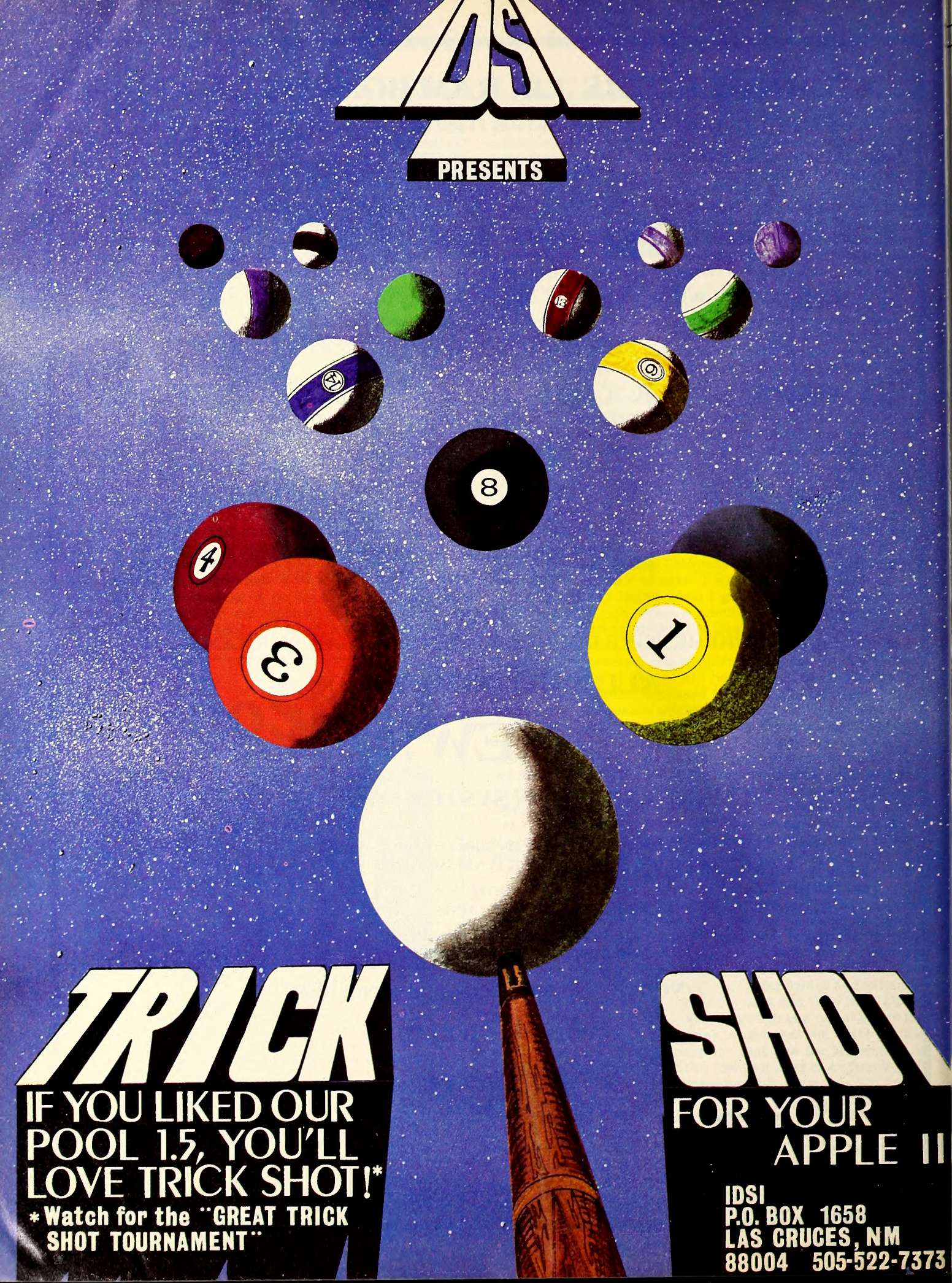
50K with 1 SATURN 32K BOARD
82K with 2 SATURN 32K BOARDS

VC-EXPAND supplied on 16 sector disk

ONLY $100.00

SATURN SYSTEMS

SATURN SYSTEMS, INC.
P.O. BOX 8050 • ANN ARBOR, MI 48107 • (313)665-6416


IDS
PRESENTS
TRICK
SHOT
IF YOU LIKED OUR POOL 1.5, YOU'LL LOVE TRICK SHOT!*
*Watch for the "GREAT TRICK SHOT TOURNAMENT"
FOR YOUR APPLE II
IDSI
P.O. BOX 1658
LAS CRUCES, NM
88004 505-522-7373

must reserve column six for a "C" (Comment = rem), if any, and can't have the statement's command begin until column seven. So she redid all the cards again the next day.

Neither she nor the other keypunch operators ever complained to me about the strange coding they were handling and about the corrections that they would have to make the next morning because the 360 had discovered, for example, an odd number of parentheses. But after a while I realized that, when I needed some work done, the supervisor of the keypunch department was letting it sit on her desk until she had some free time and was then doing it herself. I got the message and began punching the corrections myself.

The next problem took a couple of weeks to solve. Fortran II and Fortran IV are different versions of the same language and require different compilers. Fortran IV is the more advanced and has the Boolean *if* tests—greater than, less than, or equal to—that we're used to in Applesoft. It also allowed labels to be printed by putting quote marks around them. Version II accomplishes these by some very roundabout means. Because I'd been told that SBHC's compiler was Version IV, I had used both the simplified labeling system and some Boolean operators. My program bombed.

The people at the computer center kept assuring me that I was indeed using Fortran IV, but my program kept bombing. I then wrote a couple of very short and essentially identical programs, one in Version II and one in IV. The first ran, but not the second. I shook in fear at the thought of having to ask the keypunch operators to repunch everything again, and decided, since all else had failed, to read the instructions. Unfortunately SBHC's computer center had no I.B.M. Fortran manuals. The local I.B.M. office ordered some, which arrived a week or so later. After only a day and a half or so of trying to read something written by a computer engineer, I realized the problem. There are two (at least) versions of Fortran IV. the complete version is available for systems using I.B.M.'s operating system, OS. Under DOS, the disk operating system used by SBHC, only a truncated version of Fortran IV, without the Boolean operations, was available. Fortunately it did use quote marks for printing labels, so I only had to retype a few lines. But I had lost a couple of weeks.

The next major problem came when my program exceeded the 360's memory—256K. I was incredulous and spent a day trying to find where the program was attempting to divide by zero and producing such a large number that it overflowed memory. Fortran will do that. I couldn't find it. Then I discovered that Fortran is very inefficient in its storage of words and other nonnumerical symbols. So, "Automobile Loans" became "Auto Loans." As the model grew, this problem happened time and time again. At first I abbreviated labels. Later I had to go through and consolidate groups of lines: "Instalment Loans" for the combined automobile, home mortgage, and personal loans, and so on. This required renaming many of the variables and being very careful to make sure that any which were no longer used were erased everywhere.

So it went, and the weeks passed. When we had perhaps six weeks to go, I began spending my evenings at the computer center. I had found that short jobs were processed from about 3 p.m. to midnight, when the bank's large accounting run would tie up all the equipment for the rest of the night. If I simply stayed there and kept making corrections and resubmitting the job, I could sometimes run it four or five times in an evening, each time adding a few lines. The computer center had vending machines: horrible coffee and worse sandwiches, but the lunch room at least gave me a comfortable chair to wait in.

And so it went. Each day the model got a little longer and a little closer to mirroring reality. I don't remember how big it was finally. The worksheet had ten columns of figures because ten years were needed, with everything rounded to the nearest million dollars, plus a label column. There must have been somewhere between one hundred and one hundred fifty rows in the worksheet itself (about seven hundred or eight hundred in the final printout because the program successive-

ly loaded, saved, and reloaded data into the worksheet as needed). Whatever the final number of cells, they strained the capacity of what was at the time the biggest I.B.M. machine in general usage, a machine that still cost in the neighborhood of a million bucks. The truly giant number-cruncher, the 370, was out, but was still very rare.

Anyway, summer was over, the leaves were turning, and the computer center's coffee was even begining to taste familiar. The model was done, and it was time to use it. Most of the data had been worked out simultaneously with the model itself, so it was already in useable form. The first run simply examined where SBHC might find itself if nothing changed. The next run answered one of the many "What if" questions my boss wanted asked. All that was necessary was changing the numbers in one cell, just as with *VisiCalc*.

Unfortunately, this couldn't be done at a keyboard. Somebody (me, since no one else had any idea how the model operated) had to take a deck of cards that was now a couple of feet thick, remove one card (they were numbered), repunch that card with one digit changed and no errors made in the columns that were to remain the same, reinsert the card at exactly the right place in the deck and resubmit the deck to the computer operator. I had a couple of back-up decks, but the keypunch operated by the computer itself, which had copied the original deck, couldn't also type at the top of the cards what was on them, so I couldn't tell which card was which in the duplicate decks. I was very careful with the original deck.

And I was lucky. The card reader never chewed up more than a few cards at a time. No one dropped the deck without its rubber bands. And only once did the person in whose locked filing cabinet it was stored go on vacation for two weeks.

Finally the big day arrived. My boss and I picked what seemed to be the six most representative runs and presented them to the chairman of the board to show him his options, and he presented his report to the full board itself. I collected my check and moved away from that office with a great view to a higher office in the tower of another building in another city. It had been a good assignment, and I'd learned a lot. But, in retrospect, I admit that an Apple with *VisiCalc* would have saved the SBHC an awful lot of time and money.

If *VisiCalc* and Apples had been invented eight years ago, I wouldn't have been hired for this job, of course. For a fraction of the cost of training me in banking, the SBHC could have purchased its 48K Apple, two disk drives, a monitor, *VisiCalc*, and a daisy wheel printer, given a member of the research department time to learn how to use the equipment (a fine investment anyway) and, within a couple of months, have had its report. As I said at the beginning, eight years is long enough for a couple of generations in the microcomputer business.

And did the SBHC act on the recommendations of the model? Yes it did. No matter what assumptions we made, the model showed that the holding company simply could not expand at anything like the rate it was going and also repay that multimillion dollar loan at the end of ten years. In fact, the most likely assumptions showed that in only two or three years, the holding company would have to borrow many more millions instead of paying back what it had already borrowed. It was simply growing faster than it could afford to grow.

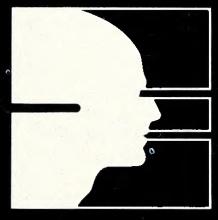
So, as much as it pained the SBHC's top management to stop growing, the holding company decided it had to stop for a while. It cancelled its applications for all its new banks and most of its new branches and planned to grow slowly from its profits. It also voluntarily reduced the assets it then had by three percent and saw profits rise by ten percent.

I don't like ten year forecasts. That's just too far out. But three years after my assignment was finished, I compared the data in that year's annual report with the "What if" forecast that had been based on the company following the recommendation it did, in fact, follow. It was gratifying. The forecasts of the two key variables, earnings, and ability to repay the loan, were off by only about one percent. That was just chance, of course. That margin was too close to claim credit for. But still it was gratifying. *VisiCalc* couldn't have done any better. But it certainly would have been faster and less painful. ◨

JUST LAUNCHED!
VOYAGE OF THE VALKYRIE
BY LEO CHRISTOPHERSON

The ancient Norse lords bravely sailed their ships across new waters in search of adventure. No obstacle was too great to overcome nor any force powerful enough to prevent their mighty advance.■ Now you're in command. You are the pilot of the attack ship Valkyrie. Your mission: to wrestle the island Fugloy from the grasp of an army of laser-wielding war birds and secure the golden treasures hidden within the walls of the island castles. Success of the mission depends solely upon you. The castle Drage appears on the horizon when suddenly the air is alive with the flutter of giant wings. Your prize is in sight . . . do you dare accept the challenge and go on?■ You can put yourself at the controls of the Valkyrie thanks to the genius of Leo Christopherson. As absolutely flickerless birds sail smoothly across the screen, strains of Wagner's "Ride of the Valkyries" are heard. The game offers 10 levels of difficulty which provide a challenge for players of all ages and skills. The game also provides one of the finest examples of computer cartoon-graphics presently available. Voyage of the Valkyrie is available for *TRS-80 and †Apple Microcomputers. Brave old worlds and battle new adversaries when you accept the challenge of the Valkyrie.

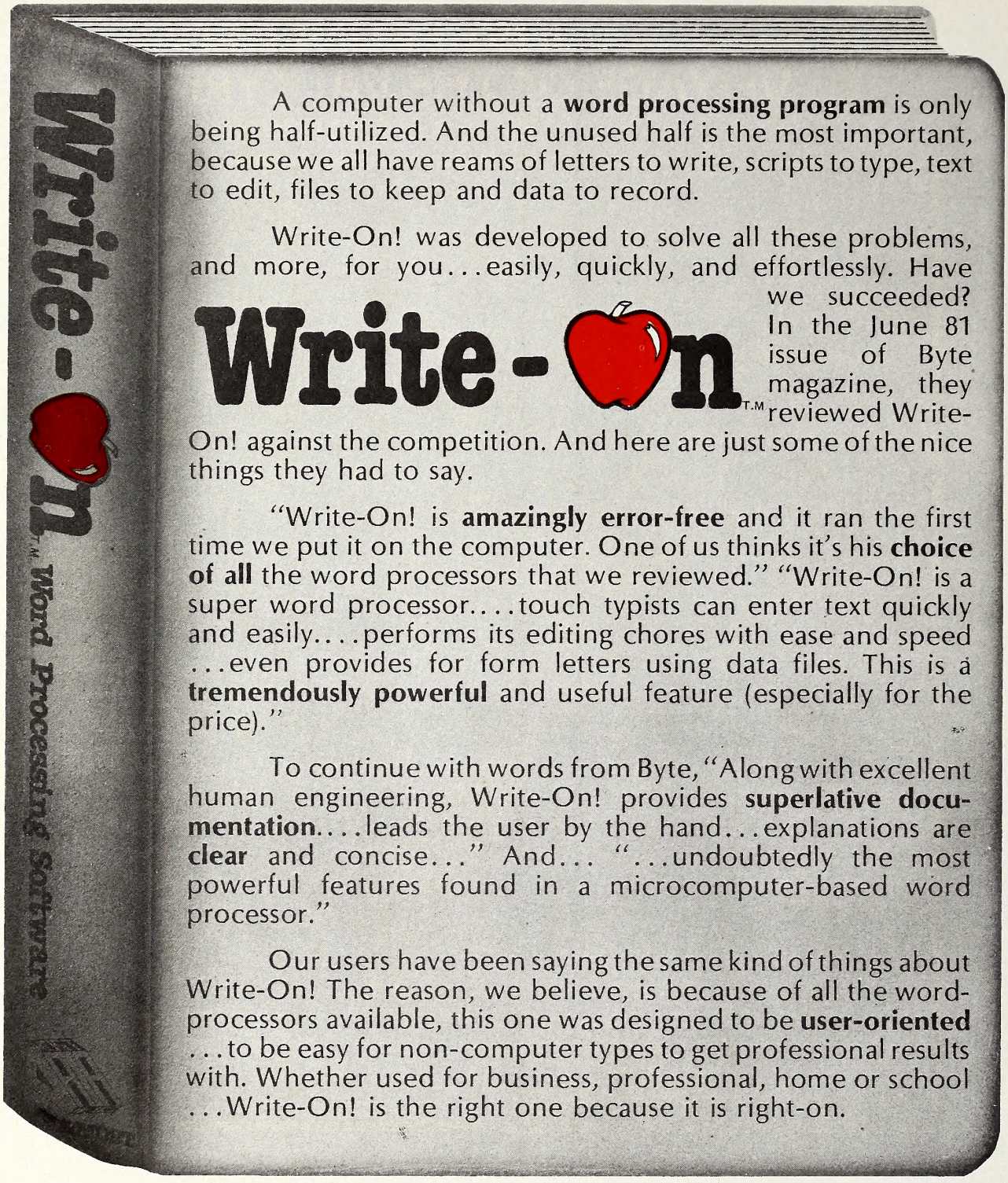
*Trademark of Tandy Corp.
†Trademark of Apple Computer, Inc.



ADVANCED OPERATING SYSTEMS

450 St. John Road
Michigan City,
Indiana 46360

Available now at leading software retailers or call 1-800-348-8558 to order. (Indiana residents call 1-219-879-4693) Disk version: $39.95, Tape: $34.95. VISA and MasterCard accepted.



A computer without a **word processing program** is only being half-utilized. And the unused half is the most important, because we all have reams of letters to write, scripts to type, text to edit, files to keep and data to record.

Write-On! was developed to solve all these problems, and more, for you...easily, quickly, and effortlessly. Have we succeeded? In the June 81 issue of Byte magazine, they reviewed Write-On! against the competition. And here are just some of the nice things they had to say.

Write-On™

"Write-On! is **amazingly error-free** and it ran the first time we put it on the computer. One of us thinks it's his **choice of all** the word processors that we reviewed." "Write-On! is a super word processor....touch typists can enter text quickly and easily....performs its editing chores with ease and speed ...even provides for form letters using data files. This is a **tremendously powerful** and useful feature (especially for the price)."

To continue with words from Byte, "Along with excellent human engineering, Write-On! provides **superlative documentation**....leads the user by the hand...explanations are **clear** and concise..." And... "...undoubtedly the most powerful features found in a microcomputer-based word processor."

Our users have been saying the same kind of things about Write-On! The reason, we believe, is because of all the word-processors available, this one was designed to be **user-oriented** ...to be easy for non-computer types to get professional results with. Whether used for business, professional, home or school ...Write-On! is the right one because it is right-on.

For the Apple III*? Of course.

Our Write-On! word processing program is available for the Apple III* as well as the Apple II*. Comes on disk, with its easy reading, simply explained, step-by-step documentation (marvelously clear!). Write-On! II is $149.95. Write On! III, $249.95. At your computer store or from:

DATAMOST
19273 Kenya St.
Northridge, Ca 91326
(213) 366-7160

VISA/MASTERCHARGE accepted.
$1.00 shipping/handling charge.
(California residents add 6% tax)

*Apple II & Apple III are trademarks of Apple Computer, Inc.

TRADETALK

from page 151

☐ **Avant-Garde Creations** (Eugene, OR) seems to be moving into the micro-familiar supergrowth stage. New distribution agreements with both **Softsel** (Inglewood, CA) and **Micro Distributors** (Fountain Valley, CA) may or may not be responsible for a four-person increase in staff. New Avant-Garde creators are **Carol Ogren, Joyce MacMillan, Gary Hill,** and **Becky O'Malley.**

☐ **Inmac** (Santa Clara, CA) has opened a New England sales and distribution center located at 28 Hampshire, Hudson, New Hampshire 03051. President **Ken Eldred** says doubling of Inmac's sales over the past year has required this expansion. "New England is a hive of computer activity in its own right, and we are pleased to offer broad product selection to the area."

☐ **William G. Hankins** has joined the executive staff of **Hayes Microcomputer Products** (Norcross, GA) as director of manufacturing and engineering. Hankins' twenty-five years' management experience in these two areas focussed most recently on startups of manufacturing facilities for Control Data, Memorex, and Ampex. A graduate of Wayne State University in mechanical engineering, Hankins hails from Detroit.

William G. Hankins

☐ Turnaround time: **Dennis Morrissey** is the new general manager of the Inmac facility. **Lifeboat Associates** (New York, NY) announces the appointment of **Edward H. Currie** to the post of marketing and business development director. In addition to the duties of his new job, Currie is finding time to work on an historical account of the microcomputer industry and a technical text. **John Stec** has been named vice president, finance, of **Whittaker Medicus** (Evanston, IL). Company specializes in the application of management science principles and computer technology to health care institutions and offers its clients consulting services, computer products, and computer software. ◨

DISCOVER THE UNIVERSE

PROUDLY ANNOUNCING

SPACE ADVENTURE

A REAL-TIME SPACE FLIGHT SIMULATOR

Welcome aboard the Callisto.

The Callisto offers you an excellent view of the vast universe before you.

You take the helm and enter the coordinates: quadrant 56, 58, 53; sector 14, 62, 12. Instantly the autopilot takes control of the ship. The viewport reappears as your course automatically changes and you begin your hyperwarp journey. Upon arrival, the message indicator flashes. You take the message and ask the computer for a tactical plot of the situation.

SPACE ADVENTURE is this and much more!

SPACE ADVENTURE offers you real-time instruments, real-time graphics, and an onboard computer that will amaze you.

SPACE ADVENTURE gives you the chance to pilot a space craft through true 3-D space for an adventure you will never forget. **SPACE ADVENTURE** . . . for any Apple II or II+ with 48K and disk. Suggested retail price is $29.95.

Sierra Software

EPISODE TWO COMING SOON
DEALER INQUIRIES INVITED

SIERRA SOFTWARE • 536 E. SAHARA AVENUE • LAS VEGAS, NEVADA 89104

Softalk Presents The Bestsellers

The results of Christmas marketing were felt in November's software sales figures, but the results were not at all what might be expected.

VisiCalc remained the leading program in the Apple market, but the distance between it and second-place *DB Master* is beginning to close. Many more retailers are now reporting that, as they do with *VisiCalc*, they're selling a *DB Master* with every business system out the door. The fact of *DB Master*'s maining in second position indicates the continuing matura-

Strategy 5

This Month	Last Month	
1.	1.	**Castle Wolfenstein,** Silas Warner, Muse
2.	2.	**Flight Simulator,** Bruce Artwick, SubLogic
3.	3.	**Robot War,** Silas Warner, Muse
4.	4.	**Sargon II,** Dan and Kathe Spracklen, Hayden
5.	5.	**The Battle of Shiloh,** David A. Landry and Charles T. Kroegel, Jr., Strategic Simulations

Adventure 5

This Month	Last Month	
1.	1.	**Hi-Res Adventure #3: Cranston Manor,** Harold DeWitz and Ken Williams, On-Line Systems
2.	—	**Hi-Res Adventure #4:** Ulysses and the Golden Fleece, Bob Davis and Ken Williams, On-Line Systems
3.	2.	**Hi-Res Adventure #2: The Wizard and the Princess,** Roberta and Ken Williams, On-Lne Systems
4.	—	**Zork II,** Mark S. Blank, Timothy Anderson, Bruce Daniels, P. D. Leblins, Scott Cutler, and Joel Berez, Infocom
5.	—	**Cyborg,** Michael Berlyn, Sentient Software

Fantasy 5

This Month	Last Month	
1.	2.	**Wizardry,** Andrew Greenberg and Robert Woodhead, Sirtech
2.	1.	**Ultima,** Lord British, California Pacific
3.	—	**Crush, Crumble and Chomp,** Automated Simulations
4.	—	**Sword Thrust,** Donald Brown, CE Software
5.	4.	**Alkemstone,** Level-10, Dakin5

Business 10

This Month	Last Month	
1.	1.	**VisiCalc,** Software Arts/Dan Bricklin and Robert Frankston, Personal Software
2.	2.	**DB Master,** Alpine Software/Stanley Crane and Jerry Macon; and Barney Stone, Stoneware
3.	4.	**Personal Filing System,** John Page, Software Publishing Corporation
4.	3.	**VisiTrend/VisiPlot,** Micro Finance Systems/Mitch Kapor, Personal Software
5.	8.	**VisiFile,** Creative Computer Applications/Colin Jameson and Ben Herrman, Personal Software
6.	9.	**BPI General Ledger,** John Moss and Ken Debower, Apple Computer
7.	5.	**VisiDex,** Peter Jennings, Personal Software
8.	6.	**PFS: Report,** John Page, Software Publishing Corporation
9.	10.	**Data Factory,** Bill Passauer, Micro Lab
10.	—	**Accounting Plus II General Ledger,** Systems Plus

APPLE PANIC

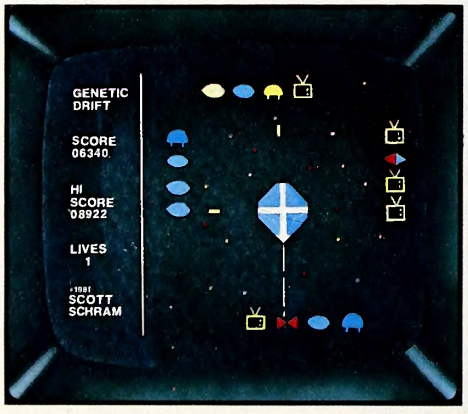
Ladders take you from level to level in this hi-res, fast action Japanese import. Dig holes in the bricks to catch a medley of monsters chasing you about. For all ages. $29.95

SPACE★QUARKS

The deadly, intricate dances of the space quarks will fascinate you in this hi-res arcade thriller. Careful study is necessary if you are to survive to upper levels. $29.95

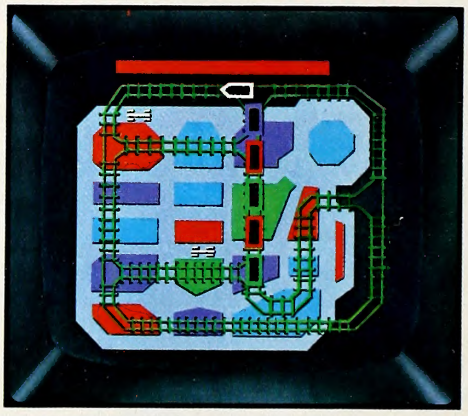
BRØDERBUND SOFTWARE

GENETIC DRIFT



This finger-pounding arcade game requires you to mutate hostile aliens into a benign life form even as they attack from all sides. But watch it! One slip and the friendly ones become deadly once again. $29.95

TRACK ATTACK



Fast cars and fast trains make this gold heist arcade game hard to beat. To steal the gold on the train while avoiding other cars you need great timing and good peripheral vision. $29.95

All products require 48K Apple II or Apple II Plus and one disk drive, 13 or 16 sector controller.

*Apple & Apple II Plus are trademarks of Apple Computer Co.

ASK YOUR DEALER OR SEND CHECK OR MONEY ORDER TO:

Brøderbund Software • 2 VISTA WOOD WAY, SAN RAFAEL, CA 94901 • (415) 456-6424

tion of the market toward business applications.

One could almost draw the same conclusion from the third-place finisher, *Personal Filing System;* but it, *Olympic Decathlon,* and *Personal Finance Manager* all benefitted from a different phenomenon. All were packaged in Apple's special Family Plan package that was merchandised in the final two months of 1981.

Instead of cutting into the individual purchase of the programs included, the Family Plan actually fueled sales because the programs were being aggressively marketed as part of the package deal.

PFS jumped from thirteenth to third, *Personal Finance Manager* leaped to seventh from twenty-fifth, and *Olympic Decathlon* rose to tenth from twenty-seventh.

There were special circumstances behind the fourth-place finish of *Gorgon* as well. Many retailers had been unable to get the product in late October and early November. Rumors that it might be removed from the marketplace because of licensing difficulties heated demand to the point where it was the hottest selling individual package in the last week of November.

This poll probably marks the last mention of *Snoggle,* twelfth in November, which was removed from the marketplace because of a licensing difficulty.

The only program that made the jump from off the Top Thirty into the top ten was *VisiFile,* Personal Software's new

Home 10

This Month	Last Month	
1.	4.	**Personal Finance Manager,** Jeffrey Gold, Special Delivery Software, Apple Computer
2.	2.	**Graphtrix,** Steve Boker, Data Transforms
3.	1.	**Typing Tutor,** Image Producers, Microsoft
4.	3.	**Home Money Minder,** Bob Schoenburg and Steve Pollack, Continental Software
5.	6.	**Data Capture 4.0,** David Hughes and George McClelland, Southeastern Software
6.	7.	**ASCII Express,** Bill Blue, Southwestern Data Systems
7.	10.	**Tax Preparer,** James Howard, Howard Software
8.	9.	**Financial Management System II,** D. R. Jarvis, D. R. Jarvis Computing
9.	—	**Mastertype,** Bruce Zweig, Lightning Software
10.	—	**Real Estate Analyzer,** James Howard, Howard Software

Hobby 10

This Month	Last Month	
1.	2.	**DOS 3.3,** Apple Computer
2.	1.	**DOS Tool Kit,** Apple Computer
3.	—	**A2-3D1 Graphics Package,** Bruce Artwick, SubLogic
4.	—	**DOS Boss,** Bert Kersey and Jack Cassidy, Beagle Brothers
5.	3.	**Enhanced MX-80 Graphics,** David Hudson, Computer Station
6.	—	**LISA,** Randy Hyde, On-Line Systems
7.	—	**Utility City,** Bert Kersey and Jack Cassidy, Beagle Brothers
8.	5.	**Super Dick Copy III,** Charles Hartley, Sensible Software
9.	6.	**Complete Graphics System,** Mark Pelczarski, Penguin Software
10.	—	**Fortran 80,** Microsoft

Word Processors 5

This Month	Last Month	
1.	1.	**Apple Writer,** Apple Computer
2.	5.	**Word Star,** Micro Pro
3.	4.	**Superscribe II,** David Kidwell, On-Line Systems
4.	2.	**Magic Window,** Gary Shannon and Bill Depew, Artsci
5.	3.	**Easy Writer,** John Draper, Information Unlimited Software

data base program. Its presence in the top ten along with *DB Master* and *PFS* is further indication that uses outside of entertainment are becoming more prevalent for the Apple.

Other software packages making the Top Thirty for the first time were *Word Star*, the CP/M-based word processor that runs on the Apple in conjunction with Microsoft's SoftCard; *Snack Attack*, which was the highest ranked new game; *Threshold*; *Superscribe II*; and *Bug Attack*. In addition, *BPI General Ledger* regained the Top Thirty.

The Strategy 5 entertainment list was exactly the same as in October, but the Fantasy 5 list underwent wholesale revisions. *Wizardry* knocked off *Ultima* for first place. *Crush, Crumble and Chomp* regained the list. *Sword Thrust* master disk from CE Software made the list for the first time and *Alkemstone* dropped to fifth from fourth.

Ken Williams shared in each of the top three programs in the Adventure 5 list. He had *Cranston Manor* with Harold DeWitz, *Ulysses and the Golden Fleece* with Bob Davis, and *The Wizard and the Princess* with wife Roberta. *Ulysses* was a new entry as were fourth and fifth placers, *Zork II* and *Cyborg*.

The Word Processor 5 list remains an interesting one. *Apple Writer* maintained first with *Word Star* jumping to second, *Superscribe II* rising to third, *Magic Window* dropping to fourth, and *Easy Writer* dropping to fifth.

However, *Supertext II* and *Letter Perfect* were in a virtual tie with *Easy Writer* for fifth. In addition, the November poll unearthed the first mention of any software for the Apple III—*Word Juggler* from Quark.

Most of the entries in the Home 10 list remained the same, although there was considerable shifting of order, with *Personal Finance Manager* not only overtaking *Home Money Minder* for first in home finance packages but also leading all home packages in November.

New Home 10 entries were *Mastertype* from Lightning Software and *Real Estate Analyzer* from Howardsoft.

The Hobby 10 underwent considerable revision. Beagle Brothers put two programs on the list: *DOS Boss* and *Utility City*. SubLogic's *A2-3D1 Graphics Package*, fueled by the requirement to have it in order to run *Saturn Navigator*, became the leading graphics package. *LISA*, Randy Hyde's assembler, rejoined the list from a new publisher. *Fortran 80* squeaked into the last position, another evidence of the various uses to which the Apple is put.

The only new entry in the Business 10 category is a notable one. *Accounting Plus II General Ledger* from Systems Plus appears ready to make a real run at BPI. Interest and sales of the product have been growing each month.

The microcomputer market in general appears to be unreasonably healthy in the face of the nationwide recession. IBM and Osborne are back-ordered and neither appear to have made a significant impact on Apple sales. ◼

Apple-franchised retail stores representing approximately 9.7 percent of all sales of Apples and Apple-related products volunteered to participate in the poll.

Respondents were contacted early in December to ascertain their sales leaders for the month of November.

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 December 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.

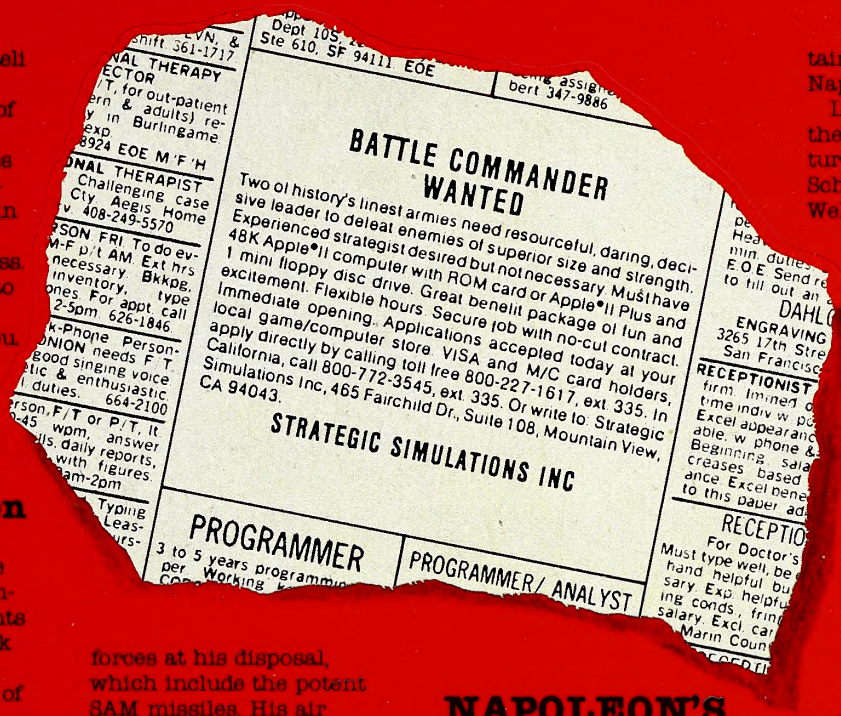
Probability of statistical error is plus-or-minus 4.1 percent, which translates roughly into the theoretical possibility of a change of 2.89 points, plus or minus, in any index number.

The Top Thirty

This Month	Last Month	Index Number	
1.	1.	97.76	**VisiCalc**, Software Arts/Dan Bricklin and Robert Frankston, Personal Software
2.	2.	70.57	**DB Master**, Alpine Software/Stanley Crane and Jerry Macon; and Barney Stone, Stoneware
3.	13.	64.74	**Personal Filing System**, John Page, Software Publishing Corporation
4.	17.	62.47	**Gorgon**, Nasir, Sirius Software
5.	5.	62.15	**Apple Panic**, Ben Serki, Broderbund Software
6.	12.	55.35	**VisiTrend/VisiPlot**, Micro Finance Systems/Mitch Kapor, Personal Software
7.	25.	46.61	**Personal Finance Manager**, John Page, Special Delivery Software, Apple Computer
8.	5.	44.35	**Sneakers**, Mark Turmell, Sirius Software
9.	—	42.73	**VisiFile**, Creative Computer Applications/Colin Jameson and Ben Herrman, Personal Software
10.	27.	42.40	**Olympic Decathlon**, Tim Smith, Microsoft
11.	3.	42.08	**Raster Blaster**, Bill Budge, BudgeCo
12.	7.	41.43	**Snoggle**, Jun Wada, Broderbund Software
13.	8.	38.20	**Apple Writer**, Apple Computer
14.	—	37.55	**Word Star**, Micro Pro
15.	—	37.23	**Snack Attack**, Dan Illowsky, Datamost
	23.	37.23	**Graftrix**, Steve Boker, Data Transforms
17.	22.	36.58	**Typing Tutor**, Image Producers, Microsoft
18.	—	36.25	**BPI General Ledger**, John Moss and Ken Debower, Apple Computer
19.	11.	35.61	**Wizardry**, Andrew Greenberg and Robert Woodhead, Sirtech
20.	14.	35.28	**Castle Wolfenstein**, Silas Warner, Muse
21.	18.	31.08	**DOS 3.3**, Apple Computer
22.	—	29.13	**Threshold**, Warren Schwader and Ken Williams, On-Line Systems
23.	—	27.84	**Superscribe II**, David Kidwell, On-Line Systems
24.	—	26.87	**Bug Attack**, Jim Nitchals, Cavalier Software
25.	19.	25.90	**Flight Simulator**, Bruce Artwick, SubLogic
26.	9.	24.92	**Ultima**, Lord British, California Pacific
27.	10.	24.60	**DOS Tool Kit**, Apple Computer
	4.	24.60	**Epoch**, Larry Miller, Sirius Software
	19.	24.60	**Pegasus II**, Olaf Lubeck, On-Line Systems
30.	16.	23.63	**Hi-Res Adventure #3: Cranston Manor**, Harold DeWitz and Ken Williams, On-Line Systems

HELP WANTED

Man for man, the Israeli army and Napoleon's Grande Armée are two of history's finest fighting forces. SSI's latest games recreate both in meticulous detail — complete in every facet except one: We've left them leaderless. Now we need someone to take the helm of command – someone like you.



SOUTHERN COMMAND™
Job Description

Superb color Hi-Res graphics reproduces the setting for this battalion-level game which recounts the Israeli counterattack to cross the Suez Canal during the October War of 1973 against Egypt.

The 28-by-39 hex-grid map of the Sinai can be viewed as one strategic screen or in twelve

$39.95

separate screens using scrolling.

As the Israeli commander, you have to smash past enemy strongholds, cross the Suez, and establish a bridgehead. In order to accomplish this, your armor, artillery and infantry units — along with your airstrikes — must successfully protect the slow-moving bridging units as they push towards the Canal.

The Egyptian commander's goal is to stop your advance using the forces at his disposal, which include the potent SAM missiles. His air force can be called upon to negate your aerial threat.

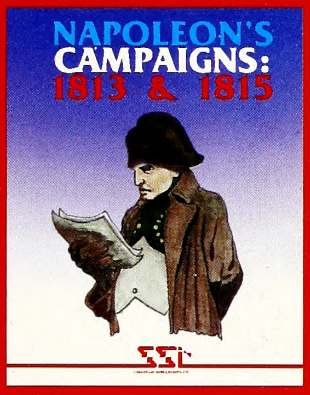
To reflect history accurately, Egyptian and Israeli forces differ in efficiency level and strength points. A unique "delayed move" feature allows for ambushes with infantry and artillery.

Thanks to machine-language programming, the computer can rapidly and efficiently calculate, display, and implement combat results to give you a fun and fast-moving game.

You never need to worry about being out of action for lack of a playing partner. The computer can direct the Egyptians and play you at any one of four levels of difficulty. So whether you're a novice or veteran gamer, you're guaranteed a challenging match.

If you feel you have what it takes (and don't wish to relocate in the Middle East), get SOUTHERN COMMAND today and the job is yours!

NAPOLEON'S CAMPAIGNS: 1813 & 1815™
Job Description



$59.95

This advanced-level, board-assisted computer simulation presents a leadership opportunity where experience in strategy gaming really helps. Aside from fully appreciating the painstaking detail and design efforts we've put into the game, a hardcore strategist can more effectively deal with the complexity, frustrations, and uncertainties inherent in Napoleonic warfare.

Leipzig and Waterloo are the two campaigns featured. Like Napoleon, Schwarzenberg or Wellington, all your commands are sent to your troops and information about them received via dispatch. Knowledge of troop positions and estimates of enemy strength are only as good as your reconnaissance patrols.

The computer plays the individual corps commanders, whose leadership ratings have been preprogrammed based on historical data. How your orders are carried out depends entirely on the corps commanders, who may follow them to the letter or do so with hesitation. Misinterpretation and even outright disobedience of your directives are also possible.

Night, rain, and terrain all affect troop movement. Intricate rules that deal with the effects of fatigue, corps morale, and leadership on battle outcome serve to mirror history faithfully. They also complicate your decision making and strategy planning.

We know we've painted a pretty tough picture of this job, and we don't expect everyone to apply for it. We're looking for those who can meet the challenge and overcome the obstacles. For these people, we guarantee the same feeling of gratification the Emperor himself often felt when he added up his victory points.

Two-player and solitaire scenarios are provided for both campaigns.

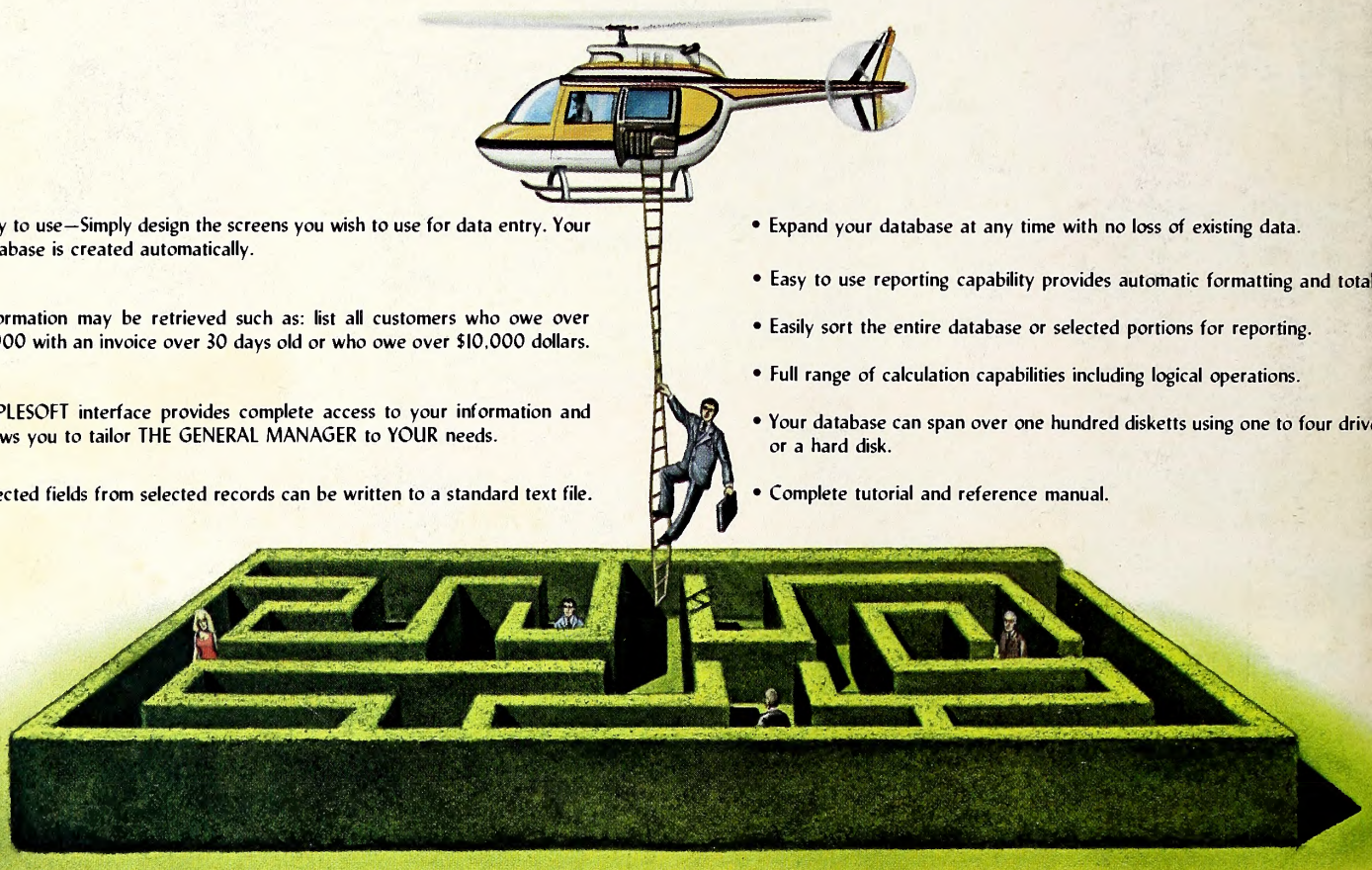
Our Want Ad has all the information you need to land this great job.

SSI

STRATEGIC SIMULATIONS INC

THE GENERAL MANAGER

THE SHORT CUT TO YOUR INFORMATION MANAGEMENT NEEDS



- Easy to use—Simply design the screens you wish to use for data entry. Your database is created automatically.
- Information may be retrieved such as: list all customers who owe over $1,000 with an invoice over 30 days old or who owe over $10,000 dollars.
- APPLESOFT interface provides complete access to your information and allows you to tailor THE GENERAL MANAGER to YOUR needs.
- Selected fields from selected records can be written to a standard text file.
- Expand your database at any time with no loss of existing data.
- Easy to use reporting capability provides automatic formatting and totals.
- Easily sort the entire database or selected portions for reporting.
- Full range of calculation capabilities including logical operations.
- Your database can span over one hundred disketts using one to four drives or a hard disk.
- Complete tutorial and reference manual.

Some database programs let you perform calculations so you can easily handle economic projections, sales forecasting, and cost accounting. Other database programs give you a wide range of search and selection options so you can find the information you need to make daily decisions. Still others format the screen for easy data entry so anyone can use the database. But now, one database program PUTS IT ALL TOGETHER.

THE GENERAL MANAGER is designed after large mainframe database management programs. The hierarchial structure allows up to sixteen unique record formats which interrelate to form the total database. You design the screens the way you want them and then fill them in. It's just like working with a set of blank forms. Information from one screen can be automatically used in other screens. Any fields can be used in complex calculations to figure costs, interest, payment terms, discount rates, recipe proportions, and many more applications. The data built on one screen may easily be used to make comparisons or build totals on another screen.

THE GENERAL MANAGER provides easy-to-use APPLESOFT interface using the ampersand command which allows user programs to read, write, update and delete database records. Uses for this interface are limited only by your imagination. In addition, using your selection criteria, THE GENERAL MANAGER can write selected fields of selected records to a standard DOS text file to be used by word processors or other existing programs.

Visa, MasterCard, Check, COD

With THE GENERAL MANAGER you are never stuck with your original database format. You may change the length and type of fields, add and delete fields and add entirely new screens at any time with no loss of existing data. As your needs change The GENERAL MANAGER will change with you. The unique access method of The GENERAL MANAGER allows your database to span over 100 diskettes using from one to four disk drives or a hard disk while at the same time using the minimum amount of disk storage to store your data. The GENERAL MANAGER has the flexibility to handle a recipe file or an entire business system.

A complete tutorial and reference manual leads you through the many uses of The GENERAL MANAGER and presents a number of solid applications.

THE GENERAL MANAGER is designed to work FOR you. You mold it to your operation and it gives you the information you need, when you need it, in just about any format you want it.

THE GENERAL MANAGER, a database management program that finally BRINGS IT ALL TOGETHER for only $99.95

THE GENERAL MANAGER runs on any 48K Apple II or II Plus with single or multiple disk drives and is available at your local computer store or order direct by sending $99.95 plus $1.00 to cover shipping and handling to . . .

ON-LINE systems

36575 Mudge Ranch Road
Coarsegold, CA 93614
209-683-6858