Apple, Apple, Apple...

"Apple's official position is that we will not support development of copy-protection schemes," according to an article published in both *Apple Direct* (January 1989, page 19) and the new *APDAlog* (Winter 1989, page 25), which are Apple publications for developers.

Simply put, copy protection is not a worthwhile endeavor. We can't help or support you in developing copy protection mechanisms. Obviously there's no way to stop you from implementing copy protection, but we won't help with schemes or problems that you get into when things break.

The article says that developers who insist on copy protecting their products will have to invest their own time and money in engineering a scheme; in maintaining it when new Apple hardware is released; in testing it on all machines and products, including third-party hardware; and in interacting with angry customers about bad key disks, hard disk installations, and backup disks. "Anything that runs can be debugged and patched...It may also be worth considering the bad press currently given to all products with copy protection," the article says.

The Apple II Technical Note suite has been completely updated and converted from MacWrite files to ProDOS text files. Apple's Developer Technical Support team invested hundreds of man- and woman-hours in this project during 1988. The results of all that effort have recently been made available to the public. Since the notes may be freely copied, though not resold, we have included them, in total, in the extra space on our monthly disk this month. They're also available in the A2Pro library on GEnie and on other online services.

This was quite a trick, since in their natural form the tech notes fill up one-and-a-half 3.5 disks. We had to dehydrate them to get them to fit on the disk. This means we also had to include a program that can rehydrate them back to normal size once you have them at your house. Complete details about how to do this are on the disk. On the next page you'll find a chart that lists each tech note currently available by number and name and that tells whether a note is new or revised (and, if revised, whether the changes were major). Those of you who are doing full-text searches of our back issues can now do the same with Apple's tech notes, too.

Apple is continuing to look for knowledgeable Apple II people to work in Cupertino. This enlightened policy beats the pants off Apple's old way of hiring minicomputer people to guide the personal computer revolution. Apple is looking for programmers, developer technical support engineers, and several kinds of marketing managers for its Apple II areas. Write Apple's Human Resources department at mail stop 27AS, 10500 N De Anza Blvd, Cupertino, CA 95014. According to an ad Apple recently placed in Call -A.P.P.L.E.:

Personal computing began with the Apple II. Thousands of applications and millions of computers later, it is the clear choice in homes and schools. The Apple II family of computers has had an incredible effect on our past and will have an even greater impact on our future.

We're committed to the continued success of the Apple II family, and that presents a rare opportunity for your future. We need more people to help us run with the Apple II and see just how far it can go. If you want to be a part of our unique and exciting team, send us your resume. You'll influence our next computers, and our next generation.

On January 27 Apple announced that its string of 12 consecutive quarters of surging profits has ended. Apple's profits for the current quarter, which will end March 30, are expected to be about 40 per cent less than the same quarter last year. Apple blames the losses on falling RAM chip prices. According to an article in the *Wall Street Journal* (January 30, 1989, page B10), Apple still has millions of 1 megabyte chips in inventory for which it paid $38 apiece last summer.

Street prices on RAM chips are below that now; in quantities as large as Apple buys, the chips are thought to be available for about $23 apiece. Apple attempted to cover its costs on the expensive chips by raising prices last September, however, that resulted in customers and dealers buying low-end machines and adding their own chips. "We thought we were ordering chips to be used in the next month or two, not the next year," John Sculley, Apple's president, told the *Wall Street Journal.* "We also thought that one-megabit chip supplies would get even tighter toward the end of the year."

Since Apple doesn't sell any Apple II products that use 1 meg chips, it would be fair to place the entire blame for the profit drop on the Macintosh product line.

On January 19, Apple announced major changes in the way it provides support to third-party hardware and software developers. The changes were driven by needs in the Macintosh marketplace, but with some concern for Apple II developers as well.

Apple's heretofore no-cost Certified Developers Program will be replaced by two new fee-based programs: Apple Partners and Apple Associates. A third level of support will be provided to developers world-wide by a reorganized APDA, which Apple bought back from A.P.P.L.E. Co-op (now called TechAlliance) late last year, as mentioned here in December (page 4.85).

The Apple Partners program is intended specifically for U.S. devel-

"YOU KNOW THE RULES, COMRADE. SMUGGLED AMERICAN COMPUTERS CANNOT BE EXAMINED UNTIL KGB REMOVES ALL 'ESCAPE' BUTTONS."

opers who publish and sell their products in the open market. Apple Partners must qualify for the program by substantiating that they have products for sale and they must pay an annual fee of $600.

The Apple Associates program is intended for those who influence or support the development of Apple products, but who don't bring their own products to market. This would include computer profes-

---

Released Apple II Technical Notes

*** New   *R* Revised

November 1988

no major changes *

Apple IIc
R	1	Mouse Differences On IIe and IIc	*
R	2	40-Column and Double High-Resolution Graphics	*
R	3	Foreign Language Keyboard Layouts	
R	4	Dvorak Keyboard Layout	
R	5	Memory Expansion on the Apple IIc	
R	6	Buffering Blues	*
R	7	Existing Versions	

Apple IIe
R	1	Overview of the Apple IIe	
R	2	Hardware Protocol for Doing DMA	*
R	3	Double High-Resolution Graphics	
R	4	RDY line	
R	5	/INH line	
R	6	The Apple II Paddle Circuits	
R	7	Interfaces--Serial, Parallel, and IEEE-488	
R	8	Known Anomalies of Enhanced IIe ROMs	*
R	9	Switch Input Changes	*

Apple IIGS
R	1	How to Install Custom BRK and /NMI Handlers	
R	2	Transforming I/O Subroutines for Use in "Native" Mode	
R	3	Window Information Bar Use	*
R	4	Changing Graphics Modes in Mid-Application	
R	5	Window and Menu Titles	
R	6	Pen Pattern Data Structure	
R	7	Halt Mechanism in IIGS SANE	
R	8	Elems Functions in IIGS SANE	
R	9	IIGS Sound Expansion Connector: Analog I/O Impedances	*
R	10	InvalRgn Twist	*
R	11	Ensoniq DOC Swap-Mode Anomaly	
R	12	Tool Set Interdependencies	
R	13	ROM 1.0 Modem Firmware Bug	
R	14	Standard File Calls and GrafPort Records	
R	15	InstallFont and Big Fonts	*
R	16	Notes on Background Printing	*
R	17	Application Memory Management & the MMStartUp User ID	*
R	18	Do-It-Yourself SCC Interrupts	
R	19	Multichannel Output with Apple IIGS Note Synthesizer	
R	20	Catalog of APW Language Numbers	
R	21	DMA Compatibility for Expansion RAM	
R	22	Proper Unloading of Dynamic Segments	
R	23	Toolbox Use of DOC RAM	
R	24	Apple IIGS Toolbox Reference Updates	
R	25	Apple IIGS Firmware Reference Updates	
R	26	ROM 2.0 Revision Summary	
R	27	Graphics Image File Formats	
R	28	Interface Card Design Guidelines	*
R	29	Monochrome High-Resolution Mode	*
R	30	Apple IIGS Hardware Reference Updates	
R	31	Redirecting Output in APW C	*
R	32	/INH Line Anomaly	*
R	33	ERRORDEATH Macro	
R	34	Low-Level QuickDraw II Routines	
R	35	Printer Driver Specifications	*
R	36	Port Driver Specifications	
R	37	Free-Form Synthesizer Tips	*
R	38	List Controls in Dialog Boxes	*
R	39	Mega II Video Counters	
R	40	VBL Signal	*
R	41	Font Family Numbers	
***	42	Custom Windows	
***	43	Undocumented Feature of CalcMenuSize	
***	44	GetPenState and SetPenState Record Error	
***	45	Parameters for GetFrameColor	
***	46	DrawPicture Data Format	
***	47	What SetDataSize Does	
***	48	All About AlertWindow	
***	49	Rebooting (Really)	

ImageWriter
R	1	Custom Font Selection	

Memory Expansion Card
R	1	Questions and Answers	

Apple II Miscellaneous
R	1	80-Column Screen Dump	*
R	2	Apple II Family Identification Routines 2.1	
R	3	Super Serial Card Firmware Bug	
R	4	AppleWorks Keys	
R	5	AppleWorks File Formats	
R	6	IWM Port Description	
R	7	Apple II Family Identification	
R	8	Pascal 1.1 Firmware Protocol ID Bytes	
R	9	AppleSoft Real Variable Storage	*
R	10	80-Column GetChar Routine	*
R	11	Examining the $C800 Space from AppleSoft	*
R	12	Apple II Firmware WAIT Routine	

AppleTalk
***	1	Identifying AppleTalk	
***	2	ProDOS 8 Compatibility on the IIe and IIGS	

GS/OS
***	1	Contents of the System and System Tools Disks	
***	2	GS/OS and the 80-Column Firmware	
***	3	Pointers on Caching	

Mouse
R	1	Interrupt Environment with the Mouse	*
R	2	Varying VBL Interrupt Rate	*
R	3	Mode Byte of the SetMouse Routine	*
R	4	Mouse Firmware Bug Affecting ServeMouse	*
R	5	Check on Mouse Firmware Card	
R	6	MouseText Characters	
R	7	Mouse Clamping	

Pascal
R	4	Pascal Declarations & Directory Structure	
R	10	Configuration and Use of Pascal Run-Time Systems	*
R	12	Disk Formatter Routine	*
R	14	Apple Pascal 1.3 TREESEARCH and IDSEARCH	*
R	15	Apple II Pascal SHORTGRAPHICS Module	*
R	16	Driver to Have Two Volumes on One 3.5" Disk	*

ProDOS 8
R	1	The GETLN Buffer and a ProDOS Clock Card	
R	2	Porting DOS 3.3 Programs to ProDOS and BASIC.SYSTEM	
R	3	Device Search, Identification, and Driver Conventions	
R	4	I/O Redirection in DOS and ProDOS	
R	5	ProDOS Block Device Formatting	
R	6	Attaching External Commands to BASIC.SYSTEM	
R	7	Starting and Quitting Interpreter Conventions	
R	8	Dealing with /RAM	
R	9	Buffer Management Using BASIC.SYSTEM	
R	10	Installing Clock Driver Routines	
R	11	The ProDOS 8 MACHID Byte	
R	12	Interrupt Handling	
R	13	Double High-Resolution Graphics Files	
R	14	Selector and Dispatcher Conventions	
R	15	How ProDOS 8 Treats Slot 3	
R	16	How to Format a ProDOS Disk Device	
R	17	Recursive ProDOS Catalog Routine	
R	18	/RAM Memory Map	
R	19	File Auxiliary Type Assignment	
R	20	Mirrored Devices and SmartPort	
***	21	Identifying ProDOS Devices	

SmartPort
R	1	SmartPort Introduction	
R	2	SmartPort Calls Updated	
R	3	SmartPort Bus Architecture	
R	4	SmartPort Device Types	*
R	5	SCSI SmartPort Control Call Changes	
***	6	Apple IIGS SmartPort Errata	
***	7	SmartPort Subtype Codes	

UniDisk 3.5
R	1	UniDisk 3.5 Internals	
R	2	UniDisk 3.5 ID Bytes	
R	3	STATUS Call Bug	
R	4	Accessing Macintosh Disks	
R	5	Architectural Differences Between 3.5" Drives	

sionals in large companies, educators, scientists and researchers, consultants, training providers, industry analysts, distributors, and members of the trade press. The Apple Associates program is open to almost anyone in the U.S. willing to pay the annual fee of $350.

APDA is for everybody, anywhere. Well, at least for everybody who is interested in obtaining Apple documentation or development software. For an annual fee of $20 (in the U.S.; $25 in Canada and Mexico, $35 elsewhere) members receive a quarterly magazine/catalog, called *APDAlog*. The Winter 1989 *APDAlog* includes 32 pages of technical articles about both the Apple II and the Macintosh and 54 pages of catalog.

The APDA catalog includes some items that aren't available elsewhere. These include Apple's own programming tools and references, such as the *Apple Programmer's Workshop* ($100), the *Apple II SCSI Card Technical Reference Manual* ($20), the *AppleShare Programmer's Guide for the Apple IIgs* ($20), and the *AppleCD SC Developer's Guide* ($22). Also included in this category are beta versions of documents such as the *GS/OS Reference*, which will be available in bookstores everywhere someday.

In addition to Apple's own products, APDA also stocks a wide range of third-party development tools and references. However, whereas APDA under TechAlliance sold these items—such as Addison-Wesley's Apple-authored reference books and Byte Works' ORCA language series—at discounted prices, APDA under Apple lists these products at their suggested retail prices (TechAlliance may continue to carry these products at a discount).

APDA's new address is mail stop 33G at Apple's Mariani address (APDA, Apple Computer Inc, M/S 33G, 20525 Mariani Ave, Cupertino, CA 95014.) The phone number is 408-562-3953 or 800-282-2732 (APDA expects to also have a Canadian 800 number by March 1).

Apple Partners and Associates automatically get APDA memberships. They also get both of Apple's developer publications, the weekly *Apple Viewpoints* and the monthly *Apple Direct;* monthly information and technical note mailings; automatic System Software updates, including a site license, documentation, change guides, and bimonthly technical bulletins (this program is also available, in two versions, separately—more in a moment); access to additional-fee developer training; and *The Information Exchange: Technical Guidebook.*

Participants in the programs also get an account on Apple's internal AppleLink email system with access to the developer bulletin board and technical libraries. Partners can also use AppleLink to contact Developer Technical Support by email with specific questions, however, Associates can't. Partners also get 12 months of AppleLink minimum fees prepaid (at $12.50 a month), Associates get only 1 month prepaid.

Partners, but not Associates, also are able to buy Apple hardware at discount; to obtain Apple Service training; to obtain marketing assistance in directories, guides, and new product publicity programs; and, for an additional fee, to participate in Apple's Annual Developer Conference (this year's conference is scheduled to begin the day after AppleFest Boston closes—Monday, May 8, in San Jose).

New participants in each program can elect to buy a "Developer's Library" of technical manuals for an additional $150. The library for Apple II developers includes *Technical Introduction to the Apple IIgs, Programmer's Introduction to the Apple IIgs, Apple IIgs Toolbox References Vol 1 & 2*, GS/OS manuals, international localization guidelines, *Apple Numerics Environment Manual, Human Interface Guidelines*, and all back issues of the Apple II technical notes.

David Szetela, Apple's Manager of Developer Services, told us that the new fees don't come close to covering Apple's costs for the programs, but do provide a way for Apple to expand its support to cover many individuals who were previously ineligible because they weren't true commercial developers. Szetela also said there would be improvements to the program and additional support products announced during the coming year.

Szetela also said that while the Partner and Associate programs are available only within the U.S., most Apple offices outside the U.S. have similar programs and Cupertino is working with these offices to strengthen their support efforts. However, based on the reports we've received, my guess is that this support is almost totally Macintosh-oriented. If you are an Apple II developer outside the U.S. and you can't get development support, write to Kirk Loevner, Director of the Apple Developer Group, at Mariani mail stop 23-EV, and tell him about your experiences.

If you're in the U.S., on the other hand, and would like to join either program, call Apple's Developer Programs Hotline at 408-974-4897 and ask for an application form, or write Apple Developer Programs at Apple's Mariani mail stop 51-W.

Automatic Apple II system software upgrades are a new service Apple is offering its customers, as mentioned earlier. The program comes in two versions, "Basic Service" at $90 a year and "Comprehensive Service" at $190 a year. Both programs include the right to make copies of the new system disks for use at a site (the update program is designed for institutions) and "full documentation."

The comprehensive service also includes a bimonthly Apple II Technical Bulletin ("provides the latest compatibility listings, power-user tips, connectivity guidelines, bug fixes and work arounds, and selected service announcements; identifies and discusses critical technical support and service issues" according to Apple) and "release notification," mailed by Apple on the day the new software is announced, that includes a list of new features as well as software and hardware compatibility information. "This allows sites to plan ahead for updating all machines on site and to acquire any necessary application or utility software updates." The system disks themselves "should arrive approximately one to two weeks after notification," a document about the new program on *AppleLink Industrial Edition* says.

"To order, you must complete a Support Services order form. For more information and an order form, contact your authorized Apple sales representative," the *AppleLink* document ends. As mentioned earlier, participants in the new Apple Partner and Apple Associate Programs get a free subscription to the comprehensive update service.

We've used the *AWP.TO.TXT* utility that comes with *TimeOut PowerPack* pretty extensively in creating our back issue and monthly disks. *AWP.TO.TXT* allows you to save an AppleWorks word processor file in a text file without returns at the end of each line. Returns appear in the text file only where they appear on the AppleWorks screen when you open-apple-Z(oom) in on them. We had a small problem, however, because *AWP.TO.TXT* gives the text files an aux type of $2000 instead of the $0000 Larry Skutchan's *READER* program (and the rest of the universe) expects. We got a patch out of Beagle Bros to fix this, and here it is for v1.1 (the version number appears when you run the utility).

```
From Applesoft, set the prefix to the directory AWP.TO.TXT is in.
POKE 768,0
BSAVE TO.AWP.TO.TXT,A768,L1,B1668
```

If you have an earlier version, search through the file for the hex bytes E3 04 00 20 and change the "20" to "00".

If you run AppleWorks from a RAMdisk, you'll be interested to know you can reclaim 8K of space by deleting two files from the AppleWorks startup directory. We've published this information before ("End of the line", page 3.95), but it's not in a table and therefore it's nearly impossible to find unless you're using something like *FastData Pro* to search through our back issues on disk. This list shows which file AppleWorks uses when it expands its desktop:

```
SEG.00   (Aux-Slot memory, incl 128K)
SEG.XM   (Std-Slot memory)
SEG.RM   (IIgs Mem-Slot memory)
```

If you are using Applied Engineering's expansion software, update AppleWorks before deleting the two unneeded files since AE's program patches all three and won't work if one or more is missing.

If you work with Apple IIs and disabled people, don't miss the catalog published by Don Johnston Developmental Equipment, Inc. (P.O. Box 639, Wauconda, IL 60084 312-526-2682). It's filled with specialized switches, keyboards, and other equipment, as well as specialized software.

Just Add Water **is the name a new booklet published by Apple's User Group Connection folks.** It brings together a series of articles on starting and sustaining a user group. Many of Apple's publications suffer from the "all form, no content" Macintosh syndrome, but this one gets pretty specific. If you already belong to a user group, it recently received two copies. If you're interested in starting a group, copies will be available while supplies last from The Apple User Group Connection at Apple's Mariani mail stop 36AA.

Rumors of a $1,000 Macintosh designed to end the Apple II line are false, Jean-Louis Gassee said in his keynote speech at January's MacWorld Expo in San Francisco. Apple doesn't plan to offer a low-cost Macintosh anytime in the next two to three years. "We have this tremendous pressure to put more power in the computer to make it more usable. And that translates into more features and more function at the same price, as opposed to the same capabilities at lower prices," Gassee said.

Cambridge Marketing, the company behind AppleFest, is urinating in the Apple II soup. Roy Einreinhofer, president of the company, promises in Cambridge's January/February newsletter that "you will see more Macintosh in the conference sessions and on the exhibit floor at this year's shows....To increase the number of attendees who use the Macintosh in upcoming shows, we will target a portion of our show promotion toward these users in the appropriate home, education, and home office markets." If Cambridge is successful, it will be the death of Applefest. No combined Apple II/Macintosh magazine or conference has been successful in the past and there's no reason one would be successful now. Why? Apple II advertisers/exhibitors aren't interested in paying to reach Macintosh readers/show attendees, and vice-versa. You'll never see more than a handful of Apple IIs at Macintosh shows and the opposite should be true as well. If you are an AppleFest exhibitor and as concerned about this trend as I am, make sure Cambridge knows that you're not interested in renting booth furniture to impress people who don't own Apple IIs.

Claris stopped supporting Apple-labeled software on February 1. From now on, to get telephone technical support from Claris you must be a registered Claris user. To do that, you either have to buy a new Claris product or a Claris upgrade to an Apple-labeled product. If you own an Apple-labeled (or Claris-labeled, for that matter) Apple-Works 2.0, you can update to the current Claris v2.1 free. Upgrades from earlier versions of AppleWorks are $75. There are also upgrades available from AppleWorks to AppleWorks GS (from v1.X, $169; from v2.x, $99). For questions regarding warranty replacement, upgrade information, registration status, change of address, and other non-technical questions, call the Claris Customer Relations desk at 415-962-8946 from 8 AM to 5 PM Monday through Thursday, 8 AM to 3 PM Friday. For technical support, call Claris at 415-0962-0371 6 AM to 8 PM Monday through Thursday, 6 AM to 3 PM and 5:30 PM to 8 PM Friday. All times Pacific. Claris now has a technical support staff of 25 and a goal to maintain an average queue time of less than two minutes.
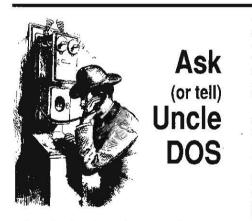
Where once there was the Apple II assembly language newsletter *Apple Assembly Line,* **there is now** *The Sourceror's Apprentice. Apple Assembly Line* died after eight years of publication last summer (see October 1988, page 4.65). *The Sourceror's Apprentice* will be published by Ross Lambert's Ariel Publishing Co. (P.O. Box 266, Unalakleet, Alaska, 99684 907-624-3161), which also publishes the Applesoft programming newsletter *Reboot* (see May 1988, page 4.28).

Source code in *The Sourceror's Apprentice* will be in *Merlin 8/16* format. The newsletter will be divided into three sections each month; one on eight-bit assembly language, one on 16-bit, and one in which the Alaskan second cousin once removed of Uncle DOS, Mike Rochip, answers readers' questions. The price is $28 for 12 monthly issues.

Besides *Reboot* and *The Sourceror's Apprentice,* Lambert also publishes *ZNews,* a $29.95/yr monthly newsletter for *ZBasic* users.

Another Apple II newsletter many of you would like is Al Martin's *Road Apple: An Apple II "End Users" Underground Newsletter.* Those of you who think I'm not hard enough on Apple and the Macintosh should read and write for this one (6 issues/yr, $9.95 U.S., $10.25 Canada, $12.95 elsewhere; 1121 NE 177th, Portland, OR 97230 503-254-3874). Sample headline from the current issue: "AppleWorks GS or How much is that doggie in the window?"

It may have been GS/OS, then again, it may have been our article on how to build your own SCSI hard drive. At any rate, sales of Apple's SCSI card have been much higher than Apple's projections and they've been in very short supply since December. Our sources at Apple say January and February production rates were increased and back-orders should be filled by the time you see this.

Ask (or tell) Uncle DOS

Let's begin with a few corrections to last month's issue. Gentlepeople, start your pencils.

On all inside pages the issue date and numbers are wrong. They should say, obviously, Vol 5, No 1--February 1989. At least I got the front page right.

In the item about Dealer Service Notices, page 5.6-5.7, I said the latest version of the Apple SCSI card ROM is called "part 341-0437, Rev A". That would be better expressed as "part 341-0437-A, popularly known as Revision C".

In the item called GS/OS tips, page 5.7, the advice is bad. Subscriber David Lyons reminds us that while you **can** *deactivate the UniDisk 3.5 driver by selecting the "inactivate" box, Apple recommends against it. Without that driver active, UniDisk 3.5 disks can be damaged if you switch them while blocks are cached. Blocks will be read from the cache, modified, and written back to the switched*

disk. Bad news. Lyons also points out that the inactivate box works by setting bit 15 of a file's aux type and that inactivation is a feature based on filetype, not on the folder containing the file. Lyons determined (by trying all the possible combinations) that the following file types show up with inactivate boxes in **Finder 1.2:** *$B6 (Permanent Init File), $B7 (Temporary Init File), $B8 (NDA), $B9 (CDA), $BB (Driver), $BC (Generic Load File), and $BD (GS/OS FST).*

Back to basics

The Apple II is heading in a direction that disturbs me greatly. That is toward increasingly complex technology, toward more and more complicated operating systems, toward a learning curve that is beginning to look like a graph of the national debt.

The doctor who wired the Apple II to a paraplegic's legs so that she could walk couldn't do it with GS/OS. While I believe Apple is on the right track in developing more powerful operating systems, the learning curve has become so steep that it takes someone wholly dedicated to the system to do anything useful with it.

The historical importance of the Apple II is that it made it easier for an expert in a field to learn enough programming to solve his own specialized problem than for a programmer to learn enough of the specialty to write the same program. On the IIgs this is no longer true. It has become too sophisticated and too complicated.

Furthermore, hiring a programmer to learn enough about either today's operating systems or about the specialist's problem to do a credible job of either is a very expensive proposition. Most research installations cannot afford the luxury. Small companies can't support one or more programmers while they learn the IIgs inside and out.

There is a place for simple and there is a place for sophisticated. The Apple II can satisfy both needs. Unfortunately, Apple is trying to kill the former and force feed Apple users with the latter.

Hannelore S. Maddox
Sunnyvale, Calif.

Apple is the only company in the world with two families of successful personal computers. One family, the Macintosh, became successful because it was easy on users. The other family, the Apple II, became successful because it was easy on programmers.

I don't think Apple has ever understood the difference, however. Instead of continuing to build easy programming into the Apple II family, the plan at Apple for five long years has been to squeeze the user interface of the Macintosh into the Apple II. Programmers who insist on using that interface lose all ease and saddle themselves with more overhead than a computer the price of an Apple II can absorb.

Interestingly, Apple itself is nearing the same conclusion. Jean-Louis Gassee, head of everything important at Apple, wrote an article for the February 6 edition of Apple's **Apple Viewpoints** *called "Algorithms Always Break Down."*

"The implementation of any algorithm always breaks down when driven beyond its range," Gassee writes. "Somewhere there is a counter with a finite number of bits that the algorithm

will underflow or overflow....Once you implement the algorithm in a machine with finite resources, it is easy to design an experiment that will create demands that exceed the ability of the resources....In real life, algorithms always have a limited range....Extending the range makes the implementation fatter, slower, and ultimately more fragile."

Now, we could meditate a little bit in front of the claims made that a single hardware architecture and/or a single operating system program will address all of mankind's woes–from user-centric personal computer uses to technically oriented workstations, process-control, or transaction processing applications.

But somewhere there are breakdowns in the algorithm, or the quality, or the performance, or the cost, or all of the above, as the algorithm becomes longer, slower, and more delicate while it grows to extend its range.

This is my point: Sometimes, just sometimes, beauty–and ultimately cost-effectiveness–can come from restraint.

Could it be that Gassee is talking about the event-driven desktop-interface algorithm as implemented on the Apple II? GS/OS is not the problem. The idea of a built-in programmer's toolbox is not the problem. The problem is putting a graphics-based, event-driven interface at the heart of a computer that has to be cheap enough for homes and schools. Neither the performance nor the programming ease of the interface match our expectations.

The problem isn't, as so many people say, that application software isn't good enough; the problem is that programming the machine is too complicated. The problem isn't, as so many people say, that the IIgs isn't fast enough; the problem is that the interface algorithm requires the horsepower of a $3,000 Macintosh, not a $1,000 Apple IIgs.

Restraint is what we needed in 1984 and 1985 when Apple decided to saddle the IIgs with the Macintosh interface. It's probably too late for restraint now, but let's hope Apple will learn Gassee's lesson and give us back a computer we can program.

In search of an 8-bit Pascal

I have been teaching programming to high school students on an Apple IIe and have had very satisfactory results. At home I use ORCA/Pascal on a IIgs for programming. At school I've been using Instant Pascal and Kyan Pascal to teach the Pascal class.

My problem is that Instant Pascal is copy protected and slow. I can't put it on a RAMdisk because of the copy protection. Kyan Pascal seems to work well enough, but the company seems to be out of business. There's no one to answer questions, provide support, or sell any of the enhancements they once had.

I'm looking for another good ProDOS-based Pascal for the IIe. Do you have any suggestions for a program that will fill my needs, or a way to get around the minor technicality of Kyan not existing anymore?

Michael Colville
Del Norte, Colo.

The lack of languages other than Basic and assembly language for classic Apple IIs has always been a critical weakness. Imagine where the Apple II would be today if Borland's

Turbo Pascal had been written under ProDOS instead of MS-DOS. Instead we've had a string of language products that have generally come up short in some critical area. **Apple Pascal 1.3** doesn't run under ProDOS. **Kyan Pascal** and **PROMAL** lack dynamic string handling. **Instant Pascal** and **Logo** weren't designed to be used as development tools. **Manx C** screen output looks like the computer is running in slow-motion. **Forth** makes you relearn algebra.

Applesoft and the **Beagle Compiler, Micol Basic,** and **ZBasic** (see "Basic choices," December 1987, page 3.88) are all still available. **ZBasic** has laid off its Apple II programmer, however (he now works for Apple).

The classic Apple II community critically needs a good, ProDOS-based Pascal. Apple has taken the initiative of hiring Gary Little, the former editor of A+ and prolific author of Apple II-related books, to persuade language developers to write for the Apple II. If he succeeds, we'll have an answer to your question. If he doesn't, your only choice will be to buy another kind of computer for your classroom. Apple hopes it will be a IIgs.

A straighter straightener

Your suggestion to the teachers who need to recover from vandalized control panels only works if the default values are acceptable. I've written a very simple S16 program that saves "snapshots" of the Control Panel's settings to disk and that can restore those settings when needed. In fact, the various users in a lab could each have a control panel file of their own or special files for specific uses. Readers may contact me for more information if interested. (On GEnie I'm L.WALTON.)

Lorne Walton
11672 212th St
Maple Ridge, BC V2X 4Z5
Canada

GS/OS problems

I've had a number of problems with GS/OS, including false messages that my disk is damaged, inability to launch programs, and more. I've heard of similar problems from other members of my users group. Calls to Claris and Apple sent me to my Apple dealer, who wasn't much help. What can you tell us?

Ronald A. Mittelstaedt
Trenton, N.J.

When Apple sends you to a dealer and the dealer isn't able to help, Apple really does want to know about it, as mentioned here in September. Write up a calm note giving the name, address, and phone number of the dealer and the name of the person to whom you spoke, your equipment and software configuration, and a brief summary of your problem. Send the calm note to:

Dealer Evaluations
c/o Customer Relations
Apple Computer, Inc.
20525 Mariani Ave, M/S 37-S
Cupertino, CA 95014

There are lots of people having trouble with GS/OS, but we haven't been able to trace many of the problems back to GS/OS itself. Given its size and complexity, GS/OS seems remarkably reliable to us.

We've traced some problems to misunderstandings of the system disk concept. GS/OS is very different from DOS 3.3 and ProDOS 8. It

works best if you leave the disk you boot from "online", which means "in the drive" if you don't have a hard disk or RAMdisk to boot from.

We've traced some problems to system disks that are missing critical files. We suggest using **Installer** to create new system disks. If you have a SCSI hard drive, a UniDisk 3.5, or 5.25 drives connected to your system, you **must** take the extra step of telling **Installer** to put those drivers on your system disk. The system disk concept and installation (or "creation") procedures were thoroughly covered here in "Breaking the incompatibility barrier: An introduction to Apple's GS/OS" in November 1988, pages 4.75-4.78, so refer to that article for more information.

GS/OS uses more memory than ProDOS 16 did. You may simply need to add more memory to your machine.

If the additional memory requirements of GS/OS make your application creep into marginal memory chips you had installed previously but never actually used before, your computer may crash under GS/OS but not ProDOS 16. All bad chip problems we've encountered are related to chips that aren't CAS before RAS. We've been covering this problem for months–see May 1987, second paragraph on page 3.30; August 1988, "Apple has also discovered...", page 4.49; and February 1989, "Apple's December Dealer Service Note...", page 5.6. Diagnostic programs that test for CAS-before-RAS haven't been widely available until recently, which is why these bad RAM chips are so widespread. Even the new diagnostic program that Apple dealers were recently issued only tests the first megabyte of memory on a card.

Some desk accessories aren't GS/OS-compatible. If you use desk accessories, remove them and see if that helps.

GS/OS won't boot from a disk attached to a controller that needs a special driver unless that driver is available on the System Disk. This includes SCSI hard drives and Video Technology's Universal Disk Controller. Since Video Technology hasn't yet created a driver for the UDC, you can't use it for booting. If you can't boot from it, you can't use it to hold your System Disk.

The only safe way to boot GS/OS is from power up or with a three-finger salute (open-apple/control/reset). Neither a "PR#" nor executing the PRODOS file work reliably for booting.

A new feature of the GS/OS Finder (or, more exactly, of the ProDOS FST) is that files with large numbers of consecutive zeros will be made "sparse" when copied. This means the zeros won't actually be stored on the disk–ProDOS just remembers where they go and reinserts them when a program reads from that part of the file. The fact that a file has been made sparse is transparent to other programs unless those programs try to access the file by "block" reads or writes or unless they check the block size of the file. There are a few programs that use these techniques, however, and they fail after being copied with the GS/OS **Finder.** These applications can be updated for GS/OS by their programmers simply by avoiding large sections of embedded zeros in a file. Change them to "A2", for example.

GS/OS doesn't work on an AppleShare network. Apple still supplies the earlier Apple IIgs System Disk v3.2 for use with its network.

One bug that we can confirm in GS/OS is that the part that displays the startup thermometer on the super-high-resolution screen doesn't restore the IIgs graphics configuration to its default state, which can cause graphics to look "torn". **Finder** and the latest version of ProSEL fix the screen, so if you use either of those as a program selector, you probably haven't noticed the problem. See "GS/OS and torn graphics" in our January 1988 issue, page 4.93, for more on this.

IIgs and 5.25 drives

I have an Apple IIgs, a 3.5 drive, and two older 5.25 drives connected through a Disk II interface card. When I set the control panel to "scan", it doesn't switch to the 3.5 drive if the 5.25 drive is empty.

And why doesn't GS/OS recognize my 5.25 drives?

Rami Levy
Brooklyn, N.Y.

When you use 5.25 drives connected to a controller plugged into slot 6, they work just like they do in a II-Plus or IIe. On those machines, if there is no disk in the drive, the drive just spins and spins until you insert one. To get the true scan that the IIgs control panel promises, you have to plug the 5.25 drives into the back of your 3.5. You'll find that's difficult, however, because the connectors are different. If you're handy with wires, our December 1985 issue, page 1.90, explains how to make an adapter. If you're not handy with wires, Redmond Cable makes and sells the adapter you need (17371-A1 NE 67th Ct, Redmond, WA 98052 206-882-2009).

To get GS/OS to recognize your 5.25 drives you have to move a copy of the file called APPLEDISK5.25 from the SYSTEM/DRIVERS folder of your /SYSTEM.TOOLS disk to the same folder on your System Disk and reboot.

Up to the second

I thought you might be interested in the enclosed item from Physics Today, announcing a time service accessible by modem offered by the National Bureau of Standards. I've logged on to this service; it prints the time on your screen every second. I would be nice to have a program that would use this to reset an Apple II's clock, but I have no idea how to do this from within a communications program.

Bernard W. Bopp
Toledo, Ohio

I called the number given in the article, 303-494-4774, and got the following display:

```
? = HELP
National Bureau of Standards
Telephone Time Service
                     D  L D
MJD   YR MO DA  H  M  S ST S UT1 msADV        OTM
47566 89-02-09 04:24:01 00 0 -.2 045.0 UTC(NBS) *
47566 89-02-09 04:24:02 00 0 -.2 045.0 UTC(NBS) *
47566 89-02-09 04:24:03 00 0 -.2 045.0 UTC(NBS) *
47566 89-02-09 04:24:04 00 0 -.2 045.0 UTC(NBS) *
47566 89-02-09 04:24:05 00 0 -.2 045.0 UTC(NBS) *
```

When I pressed "?" for help I learned that the column marked "MJD" holds the "Modified Julian Date," which is used to tag certain scientific data. I could figure out "YR", "MO", "DA", "H", "M", and "S" myself, but discovered that the rea-son the service said it was 4:24 when it was in fact 10:24 (Central) was that the service gives "UTC" or Universal Time Coordinated (also known as Greenwich time).

The column marked DST holds "00" when the continental U.S. is on standard time and "50" during daylight savings time. "99" to "51" means now on standard, but we'll switch to daylight when the local time is 2 am and the count is 51. The count is decremented daily. "49" to "01" means now on daylight time, go to standard when the count reaches "1". This provides 48 days advance notice of a change in time.

The "LS" column has to do with leap seconds. "DUT1" is the approximate difference between earth rotation time (UT1) and UTC, in steps of 0.1 second (DUT1 = UT1 - UTC). The "msADV" column tells how far in advance of the exact time the "OTM" "*" marker at the end of the line was transmitted. The 45 milliseconds shown is the average amount of time it takes for the marker to reach your computer. If you echo all characters back to the National Bureau of Standards, its computer will measure the round trip delay and advance the on-time marker so that the midpoint of the stop bit arrives at the user exactly on time. In that case, the amount shown under "msADV" will reflect the actual required advance in milliseconds and the OTM will be a "#". For the best possible accuracy (1 millisecond) call in at 300 baud.

You can use the service to set your watch right now, but to use it to set your computer's clock we need someone to write us a program. Any volunteers?

French in France

After reading the different contributions under the header 'Foreign accents, cont' in your January issue (page 4.95), I would like to promote a little bit more the European developers working on the Apple IIgs.

The KEY.FR desk accessory that you mentioned is good but sometimes unexpected incompatibilities are noticed either with applications or other NDAs. I am currently using another NDA for the same purpose called Keyboards. It's freeware and is available from its developer, who will also supply the source code upon request–Marc Klein, 9 rue des Rochers, 67190-Gresswiller, France.

And an excellent word processor you should know about is Gribouille. It's written under ProDOS 16 and includes a font editor. It supports alphabets that read from either left to right or right to left, which means you really can use it for Hebrew or Arabic! The price is about US $150 from Gribouille, 5 rue Humblot, 75015-Paris, France.

Jean-Francois Sauvage
Strasbourg, France

TimeOut vs AE Expander

I've just found out from Beagle Bros that their TimeOut Paint program won't work with Applied Engineering's AppleWorks expansion software unless AE is willing to make some changes. This paint program isn't the most powerful available, but it is adequate for my purposes. Plus, it does pop-up inside AppleWorks (always a desirable feature).

Both Applied Engineering and Beagle Bros are important to Apple II users and both are largely dependent on this market. It would mean a great deal for one to be willing to change in order to accommodate the other. We should encourage this to happen, not just for the paint program, but as an ongoing effort to reduce software incompatibilities.

Jim Pendarvis
Fairfax, Va.

Historically, Applied Engineering's expansion software was very important to AppleWorks and the Apple II (see "RamWorks transforms AppleWorks," May 1985, page 1.34). AE invented the expanded desktop. But AppleWorks itself has included automatic desktop expansion since v1.3. We stopped using AE's expansion software around here shortly after that.

Alan Bird wrote the original version of **Time-Out** to work with the AppleWorks expanders that existed at that time, including AE's. He contacted the authors of the existing expanders and suggested some ground rules they could all follow to stay compatible with each other. The companies have largely cooperated with each other, but there does seem to be an incompatibility between **TimeOut Paint** and AE's expanded clipboard. Let's hope they can work it out, at least for the sake of those of you who have Applied Engineering's aux-slot type memory cards and who can't get desktop expansion without AE's software.

AppleWorks reset revisited

There are times when the reset patch you've given for AppleWorks (May 1987, page 3.32) doesn't work. AppleWorks seems to come back to life, but with someone playing the keys. It walks through various menus, fires up disk drives, and so on. Nothing will stop it but another control-reset. It occurred to me that the problem might be that the "key pressed flag" wasn't being cleared, so I altered your patch to do that. So far it seems to work. But what's the address you call for AppleWorks 2.1?

Cecil Fretwell
Waterloo, Iowa

The jump for v2.1 appears to be the same as the one for v2.0. Here's an update of our AppleWorks reset patch, with your alteration included.

```
Press control-reset to get into the Monitor.
    (If you can't get to the Monitor,
    you're stuck.)

At the asterisk, enter (all "0"s are zeros):
  C073:0 (return)
  C300G (return)
    (If this doesn't get you an 80-column
    display, you're stuck.)

  2F0:2C 83 C0 2C 83 C0 2C 10 C0 4C (return)

for AppleWorks 1.1 - 1.3 only
  :33 10 (return)
for AppleWorks 2.0 - 2.1 only
  :27 11 (return)

  2F0G (return)
```

How does DOS move?

I am very curious about the technical details of DOS movers. I was very disappointed to find that you had made no mention of this in your otherwise decent DOStalk Scrapbook, even though you have written such a program.

Note: Apple didn't ask for (or pay for) the rights to my initials on their current top-of-the-

line II. I wonder if they acquired them from the Girl Scouts, who implicitly have my permission, support, etc.

> Gary Swift
> Urbana, Ill.

There was nothing easy about writing DOS-UP, my program that moves DOS 3.3 to the 'language card'. To do it, you have to figure out a way for Applesoft's machine language instructions and DOS 3.3's to share the same address range (Applesoft in ROM, DOS 3.3 in RAM). You end up with a little piece of code in the memory area at $BF00-BFFF that flips between the two, depending on who's turn is next. Since DOS wants to look at every character you enter on the keyboard, the code gets a lot of use and the contents of the address range sometimes flutter like a hummingbird's wing.

*The truth is that I couldn't have done it without two articles written by a prolific European named Cornelius Bongers and published by **Call -A.P.P.L.E.** back in 1981. See if you can check these out of your local university or user group library: 'Loading DOS 3.3 on the Language Card', July 1981, page 9; and 'Notes About DOS Mover', November 1981, page 81.*

Computer education news

I'm in the process of compiling a list and exchange of educational computing newsletters. This project began as an article for the Ohio Educational Technology Association newsletter. When it reached seven pages, I decided there was a need for a central clearinghouse. If any of your subscribers know of such newsletters, I'd appreciate a note about it.

> Howard Moskowitz
> Toledo Public Schools
> Manhattan and Elm
> Toledo, Ohio 43608

International user group

Please announce to your readers that there is an international Apple II user group for American military personnel overseas (SF MAUG). Contact either of us for more information.

```
Pat O'Connor          Major Mike Murley
40 Mass Drive         Box 484
Ft. Bragg, NC 28307   APO New York, NY 09050
```

Better than *GS BASIC*

In January you mentioned that the *APDAlog* indicates there won't be a final version of *GS BASIC* (page 4.91). There is, in fact, a release version of the program, sort of. *TML BASIC* is a compiled superset of *GS BASIC*. In my opinion no one should buy *GS BASIC*. If they want a product like it, they should go straight to *TML BASIC*.

> Doug Trueman
> Renton, Wash.

Uni to Apple "kit" found

You had a question in the January issue (page 4.92) about making a UniDisk 3.5 into a 'dumb' Apple 3.5. My local dealer, actually a nice fellow, will sell me a new Apple 3.5 daisy chain 'analog' board for $53. Swap this with the 'digital' board in the UniDisk and you have an Apple 3.5. Opening the metal case takes a regular screwdriver and a little persuasion.

> Alan Womack
> Spokane, Wash.

ImageWriter LQ ribbons

The ImageWriter LQ doesn't need a twisted ribbon and you don't need to play with the color control codes to get the printer to use it all, as you suggested in January ('Yorkshire ribbons,' page 4.93). It has a ribbon sensor switch that can differentiate between a color ribbon and a black one. If a black ribbon is detected the ribbon will be shifted up and down to spread out the wear. E.G. Wood's printer probably has this feature disabled. He can fix it by setting switch B-2 to 'open' or by sending the following control code to the printer "ESC Z CTRL-@ CTRL-B".

I have had my LQ, which I use with a IIc, for over a year but just recently discovered the Letter Quality aspect of it. I had been setting the controls to Letter Quality and printing; however this is actually Near Letter Quality. To get true Letter Quality you also have to print in proportional mode, which is easy by means of the 'P1' and 'P2' commands in AppleWorks. The quality of NLQ was so good I thought it was LQ.

The only loss of compatibility in moving from the ImageWriter II to the LQ was that it does not eat paper. Ah, the sacrifices we make.

> Bob Church
> Athens, Ohio

Label relief

The next time you get a pressure-sensitive mail label caught beneath the platen of your printer, get a playing card (the queen of spades works best) and roll it in just as you would a sheet of paper at the place where you think the label is stuck. I hope you're surprised at how easily the card pushes them out.

> Stan Bellas
> Sepulveda, Calif.

We've also had typewriter cleaner suggested as a solution ('ImageWriters in the office,' October 1988, page 4.70). We've also suggested that the best way to avoid the problem in the first place is not to roll labels through your printer backwards ('Mail label musings,' September 1986, page 2.59). When you're finished printing, tear them off behind the printer and waste the ones you have to roll forward. I've heard there's a printer repairman here in Kansas City who adds an additional $50 to the bills of people who have this problem more than once.

AppleWorks screen snapshots

In various capacities I write about AppleWorks a lot. It would help a lot if there was a way to incorporate the various AppleWorks menus as they appear on the screen into word processor documents without having to type them. Is this possible? I think many persons who write about AppleWorks could benefit from such a function.

> Richard Marchiafava
> Fridley, Minn.

We agree. The obvious choice is to set up the open-apple-H(ard copy) printer so that it prints to disk, but I just tried it and AppleWorks won't allow it. Can anyone write a patch? (Having it print to the clipboard would be fine, too.)

*If you have a IIgs, the newest version of Roger Wagner's **SoftSwitch** includes a function that allows you to capture the current text screen and store it in a file. See our November 1988 issue, page 4.75, for more on **SoftSwitch**.*

Another possibility is to replace your printer with another computer running communications software. The open-apple-H command would then send the current AppleWorks screen to the other computer, where you could capture it and save it in a file.

Manual auto-recalc

I've found a problem with AppleWorks 2.1 I'm hoping you can help solve. When a program such as *Quicken* creates a spreadsheet and you look at it with 2.1, there's no way to recalculate it. Since AppleWorks thinks nothing has changed, open-apple-K(alculate) does nothing, even though most calculated data is incorrect. Using AppleWorks 2.0 to recalculate gives the correct results. Is there a way to force 2.1 to recalculate everything?

> Lawrence J. Merow
> Sayville, N.Y.

*Hummm...interesting oversight. It appears that AppleWorks 2.1 will recalculate all cells after you move a block to a new spreadsheet with the clipboard. Load your **Quicken** file, add a new spreadsheet to the desktop, then move the **Quicken** file to the new file using the clipboard. Then tell us if this worked.*

No-Slot Clock fix

On page 5.8 of last month's issue there's a letter from Frank Eddy about his No-Slot Clock not working with a 65802. I don't have a 65802, but my No-Slot Clock wouldn't work with my Rocket Chip until I put it on a slot-based card rather than on the motherboard. You have to instruct the No-Slot Clock software to look for it in the appropriate slot. You also have to re-modify your ProDOS file. But it works.

> Vic Weisskopf
> Lincolnwood, Ill.

Thoughts on Micol

I find your high recommendation of *Micol Advanced Basic* for the IIgs to be a bit unnerving. I, too, have been increasingly aware of Applesoft's inadequacy for the IIgs and I was looking for a program exactly like the one you describe this to be. I got the impression from your brief review that moving from Applesoft to Micol's Basic would be relatively straight forward and fun.

I've had the package for several days; I've read the manual and have done some experimenting with it. I'd like to share my knee-jerk reactions with you.

First, you may be right when you say that us old Applesoft hackers won't have to learn much to get started; but it's really a question of how much one has to un-learn! Despite all its enhancements to basic Basic (Dartmouth or Applesoft), there are so many places that Micol's package insists that you do things its way or no way at all that, instead of starting with a firm foundation of acquired knowledge, I find myself scrabbling for some kind of handhold.

The biggest problem I have with the package is its holier-than-thou attitude. In many places, the manual comes off sounding like a fundamentalist preacher warning of the evils of 'unstructured programming' (i.e., programming that contains GOTO statements) with *Micol Advanced Basic* being the key to the programmer's salvation. I get the feeling that the many things in the program that one can't do are illegal merely because the programmer feels that you should not be allowed to do them.

Examples: the program won't let you use a NEXT statement without a terminator variable (statements such as NEXT or NEXT Y,X are both
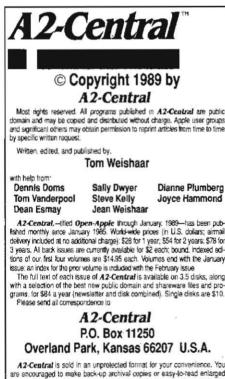
illegal). It demands one and only one NEXT statement for each FOR statement; to do otherwise is 'bad programming' and is not permitted. It demands a semi-colon between items in print statements (optional in Applesoft). One can't effectively alter the counter variable in FOR/NEXT loops, which is a pretty standard way to prematurely exit such loops. It insists each program have a header line, the operational necessity of which escapes me.

Now I'll grant you, for almost every limitation the program contains there is an alternate way to accomplish the task at hand. *Micol Advanced Basic*, as the documentation says, is hardly a compiled version of Applesoft, and I really don't expect it to follow each of Applesoft's idiosyncrasies; yet the word 'advanced' makes me think that the package should offer *more* possibilities, not just *different* possibilities, and certainly not *fewer*.

I really don't want to get into the structured vs unstructured programming debate, but one of the reasons Basic is such a friendly language is that you can have as much structure in your programming as you want without having it forced down your throat. If I'd needed or wanted that sort of imposed rigidity, I'd have taken up Pascal long ago.

Micol's biggest shortcoming, as far as my own programming style is concerned, is its Boolean variable type, which apparently precludes reducing a Boolean expression to a simple binary 1 or 0 (one of Applesoft's handiest features). A statement such as

`ON (X>1) GOTO 100`

is not permitted. This particular example is not an extreme hardship, for the line could almost as easily be written IF (X>1) THEN GOTO 100. But something such as this is equally invalid:

`Value = 5 * (X>Y)`

This sort of Boolean structure, it seems to me, is one of Applesoft's greatest strengths and worth its weight in WHILE/WENDS. I'll have a hard time giving up this one. It looks like only the most uncomplicated and elementary Applesoft programs will port into *Micol Advanced Basic* without some extensive reworking.

There are many other odd programming deci-

sions (bugs?, limitations?) in this package. The printer output routine, for example, gives you two configuration options: each carriage return issued either has no linefeed appended, or is surrounded by them fore and aft. This gives my ImageWriter II, configured as Apple suggests, the option of a single line continuously overwritten or double-spaced text. Granted, I could change the DIP switch setting or the control panel to compensate, but it seems unreasonable to have the make the printer incompatible with the rest of my entire set up just to get a printed listing. How come this isn't a problem in any of my other programs?

Micol Advanced Basic apparently has major limitations in dealing with strings. You cannot do anything with an element of a string array (other than printing it) without first copying it to a normal string variable and then copying it back when through. Complicated string manipulations must be coded in short, easy-to-digest steps.

With all this said, I'll still probably take the time to learn to use the package efficiently, as it does appear to be the simplest route from Basic into the IIgs Toolbox Routines. And it will certainly allow one to write very large programs. But then again, would the same programs coded in Applesoft *need* to be as large?

I put together a quick program that creates 50 six-letter strings and then does a simple bubble sort on them. Once I had the program up and working, I realized that the sort seemed to be taking awfully long. I constructed the equivalent program in Applesoft and ran it, just to compare. The Micol sort ran in 8.4 seconds; Applesoft ran in 8.8 seconds. Just for the heck of it, I used the Beagle Compiler on the Applesoft version, that made the program execute in 1.9 seconds.

The Micol program file is over twice as long as the text file that holds the program, which is over twice as long as the equivalent tokenized Applesoft file. The Beagle-compiled file is, of course, the shortest of all. I'll grant you that this is not exactly a scientifically controlled comparison of the two Basics. But if this sorting routine is any indication, Micol's package takes a lot more steps and disk space for a microscopic increase in execution speed. I am *seriously*

unimpressed with this package, especially in light of how much money Micol wants for it.

I can't imagine why you'd want to encourage the distribution of such an inferior package by offering it for sale at a reduced price. Perhaps you have family in Canada (grin)?

Elliot Lifson
Yonkers, N.Y.

*I think some of your criticisms of **Micol Advanced Basic** are valid. Areas in which I also have trouble with the package are where it doesn't match Apple II standards–such as the backward operation of the Delete key (which you didn't mention) and the linefeed problem. I think the package would be much improved if we could get Steve Brunier, the program's author, to sit down with AppleWorks, **ProSEL**, and Apple's **Human Interface Guidelines** for a week and convince him to make Micol's interfaces match those. This is also a valid criticism of many other programs, including **Fastdata Pro**, which we carry even though its user interface needs a lot of work. If programs of equivalent quality that conformed to the standard Apple II interface existed, we would carry them, but, unfortunately, there aren't any.*

*I also find the manual's attitude that Micol was written to save us from Applesoft troubling. Like you, I **like** Applesoft (especially with the Beagle Compiler). But the Applesoft/Beagle combination has a few weak spots. In my opinion, those are line numbers (rather than named subroutines), two-character variable names, lack of good support for add-on assembly-language enhancements, and lack of support for the advanced memory, graphic, and sound features of the Apple IIgs. Micol has none of those weak spots and is otherwise remarkably similar to Applesoft.*

Micol is weaker than Applesoft in handling Boolean and string array variables, as you mention. This is a valid criticism. My understanding is that Brunier wrote Micol this way for speed. (What speed, you say? Micol's speed suffers mostly because it uses the IIgs SANE toolbox for number crunching--bad. But this means programs written with Micol will easily adapt to the addition of a math coprocessor-good.)

Most of your other complaints have to do with the way Micol forces 'structure' into your programs. Write a few programs with Micol and Applesoft, wait three months, then go back and modify them. At that point I don't think the forced structure will seem like such a big issue to you.

Micol's program files are larger in part because they include "relocation" information. They will load and run at any IIgs memory address.

*You ask why I "encourage the distribution of such an inferior package". Inferior to what? Micol may be inferior to perfection, but perfection isn't an option in the world of IIgs-compatible Applesoft-like compilers. I'm encouraging the distribution of Micol because in my judgment it's the best IIgs-compatible language currently available for non-professional programmers. It's not perfect, but Steve Brunier is a reasonable person who has limited his development efforts to the Apple II family. He's worth supporting and I'm confident he'll incorporate reasonable ideas of what **Micol Advanced Basic** should be in future revisions of the software.*