

# Open-Apple™

March 1986  
Vol. 2, No. 2

ISSN 0885-4017  
newsstand price: \$2.00  
per page photocopy charge: \$0.25

**Releasing the power to everyone.**

## Drumming insight out of data

Chemists at East Carolina University have converted the output of an infrared spectrophotometer, a tool that normally identifies a chemical by graphing the peaks and valleys of its infrared spectrum, to music. "The human sense of hearing has remarkable powers of pattern recognition," one of the chemists says, "but hearing has largely been ignored as a means of searching for patterns in numerical data."

Stock market analysts have converted the annual reports of companies into drawings of human faces. The length of the nose, the size of the eyes, and the tilt of the lips are related to a company's financial performance. By looking for big smiles the analysts can quickly pick out companies with the financial characteristics they want.

These chemists and stock analysts are on the cutting edge of knowledge in the field of *data reduction*. As you probably know, there's too much data. This is not a new, computer-age development, incidentally—there *always* has been too much data. Humans have traditionally dealt with it by organization and summarization.

The ancients grouped stars into constellations. Teachers distill hours of scholarly progress into one-letter grades and administrators reduce grades into three-digit grade-point averages. The human mind isn't big enough to survive without these tricks, yet important details often disappear when data is reduced.

Perhaps a grade-point pastoral would be a better way to summarize scholarly progress. Or maybe a drawing of a face—where the mouth indicates facility at languages, the forehead success in math, and the nose time spent on extracurricular activities.

**Input devices are the one area in which computers clearly don't match humans.** Computers have nothing close to our abilities to see, to hear, to touch, taste, and smell. Not only can we see three dimensions in full-color at ultra-high-resolution, we can also follow moving objects, focus from a finger length to infinity, and see in near darkness. And just as importantly, we can quickly reduce what our eyes see, and what our ears hear, into relationships.

In the *output* area, on the other hand, computers have advanced way beyond what we can do. It's common to hear a computer speak. The human hand is no match for an ImageWriter when it comes to putting things on paper.

Out on the cutting edge of data reduction the computer's advanced output capabilities are being combined with the advanced input capabilities of humans. The hope is to find new methods of data reduction that are faster and less likely to lose details than those now in use.

What's exciting about this field is that an Apple II can provide all the computer power needed to draw pictures and to sing songs about data. The field is an easy one for "amateurs" to make major advances in. Sifting through data, organizing and summarizing it, and forming theories about relationships is hard work the way we do it now. It can be made easier.

**Meanwhile, back from the cutting edge,** back in the mines where today's humans toil, it's quite common to find Apple IIs being used to *collect* information. What's less common is to find them being used to *reduce* what has been collected. However, with a few tricks, you can easily use even a simple data base program such as AppleWorks to distill knowledge from a collection of data.

Given that it's hard to know much of anything without summarization, it's surprising that most data base programs provide few data reduction tools. AppleWorks can do a great job of keeping a group's membership roster, printing the group's newsletter mailing list, and sorting out the people who haven't renewed their membership. But, like many other data-base programs, it has no tool for telling you *how many* people haven't renewed.

Because it won't tell you how many, you can't determine the proportion or percentage of renewals. Counts and proportions are probably the most frequently used type of summary or data reduction.

The key to getting record counts out of AppleWorks and many other data base programs is to set up a category in every file called "counter," and to make sure that every record has the number 1 stored in that category.

In my own databases, I define "counter" as the last category in the record and I use open-apple-L(ayout) to move it to the bottom-right corner of the screen. I tell open-apple-V(alues, standard) to put a 1 in this category automatically as new records are entered. Occasionally I use open-apple-R(ecord select) to display all the records that don't have a 1 there. There shouldn't be any.

If all your files have a field like this, and all the records in the file have a 1 in it, you suddenly have a whole new way to analyze and reduce your data.

For example, imagine you have a membership roster and you want to know how many people live in each zip code.

Use open-apple-P(rint) to create a new "tables" format. Delete all the categories from this format except for the zip code and the counter. With the cursor on the counter category, press open-apple-T(otals). This tells AppleWorks to add up the 1s as the report is printed. Next, put the cursor on the zip code category and press open-apple-G(roup totals). This tells AppleWorks to print a total for "counter" every time the data in "zip code" changes. AppleWorks will ask you if you want to "print group totals only?" Say yes. This means that instead of printing *every* record in the report, it will print only when the zip code changes.

Now, use open-apple-A(rrange) to sort the zip code category. This puts all the records with the same zip code next to each other. Finally, open-apple-P(rint) the report. AppleWorks will quickly count how many records have the lowest zip code and print out both the zip code and how many records have it. Then it will go on to the next zip code. In less than five minutes—start to finish—you can know exactly *how many* of your members live in each zip code, *how many* of your Cabbage Patch dolls have red hair, or *how many* books are checked out of your library.



Once you know how many, you can use a spreadsheet or a calculator to figure out percentages. These tricks allow you to reduce the data in your file into a few numbers. You'll lose some detail, but you'll end up with a better understanding of your data. And someday soon an *Open-Apple* subscriber will send us a program that will read a DIF file filled with AppleWorks data and make a symphony out of it. I can't wait.



## Miscellanea

**Central Point Software has had a big year so far**—it has introduced new flagship products in both the hardware and software markets and its president got his picture in the *Wall Street Journal* on February 7, page 17.

The hardware product is the Lazer 128, an Apple-compatible IIc-like computer that sells for \$395 plus \$20 shipping. The machine is made in Hong Kong by Video Technology Computers Ltd. It's slightly wider than the IIc—the extra width goes into a numeric keypad on the front and a parallel printer port on the back. It also has a single Apple-compatible slot on the side (expandable with an optional extender to two slots, 5 and 7) and 10 non-programmable function keys that return control-codes. Unlike the new Franklins, the keyboard on this machine is an exact duplicate of the standard Apple IIe/IIc keyboard (except for all the extra keys), which makes it suitable as a second computer in an Apple office or household.

According to the manual, the computer has a licensed version of Microsoft Basic built in. Microsoft wrote Applesoft, but I haven't had a chance to determine if there are any differences between real Applesoft and what's in the Lazer. The few Applesoft programs I've tried ran fine, as did ProDOS and AppleWorks. The unit has 32K of ROM and the Monitor appears to have been completely rewritten. It doesn't seem to have the L(ist) command and the Lam routine (see February 1985, p 12) entry point, CALL -141, is gone. On the other hand, all the mouse characters are there except the apples, which have become triangles. Compatibility with existing software and hardware is likely to be a mixed bag, but at least the thing looks legal. The ROMs are accessible through a door on the bottom and Central Point says it intends to revise them.

Sales of the computer are at a standstill at the moment, except to schools and businesses, because of delays in FCC radio-frequency approval. A spokeswoman for Central Point told me in late February, however, that the computer had passed the FCC's tests and that shipping would begin as soon as all the paperwork had cleared.

Most clones offer either a lower price or additional features; few offer both as the Lazer does. It isn't a perfect machine—the serial ports have the same timing problem as the original IIc and the quality out of the box isn't quite up to what we have come to expect from Apple. My unit's screen display is off center and the period key on the keypad doesn't work reliably. But it's pretty close to perfect for \$395. This one will put some significant price pressure on Apple.

**More price pressure is coming from cheap IBM-clones.** *Open-Apple* subscriber Tom Smith points out that systems built around the Leading Edge Model D are cheaper than Apple IIe systems. Smith also reports that the computer store he has dealt with in the past is dropping the Apple II when it sells the last of its stock in favor of "business" machines. Smith and I wonder whether Apple's Macintosh promotions have helped that machine as much as they have hurt the Apple II in the business market. An Apple II with a hard disk and a memory expansion card running AppleWorks is a better, less-expensive system for most corporate desks than an IBM running Lotus, but have you ever heard Apple say so?

Central Point includes its new ProDOS version of *Copy II Plus* in the box with the Lazer 128. This top-selling disk utility is available separately for

\$39.95 plus \$3 shipping or for \$18 on an upgrade basis (9700 SW Capitol Hwy, Portland, OR 97219 503-244-5782). It allows you to convert files between DOS and ProDOS (including random-access text files); copy, delete/undelete, rename, lock/unlock, verify, and view both DOS 3.3 and ProDOS files; and copy disks—including most copy-protected ones. When you catalog a ProDOS disk with this program, it draws the subdirectory paths on the screen and you can move from subdirectory to subdirectory by pressing the arrow keys.

**Don't try to sort ProDOS directories** with version 6.0 of *Copy II Plus*, however (they're up to version 6.1 already), or with most other ProDOS directory sorters. They seem to have a common bug that eventually wrecks your disk. The bug was discovered by Glen Bredon, who now includes a directory sorter on his \$20 ProSel disk. (The disk also includes a ProDOS program selector, a RAMdisk driver for auxiliary slot memory cards, a hard disk backup utility, a disk copy program that can backup an 800K disk in one pass if you have enough memory, ProDOS 40-track floppy utilities, and other goodies—521 State Rd, Princeton, NJ 08540.)

Bredon suggests that before you use *any* ProDOS directory sorter you make sure it passes the following test. Format a new floppy. Name it /TEST. Then:

```
PREFIX /TEST
CREATE DIR2
CREATE DIR1
```

--Use your sorter so that DIR1 occurs first in the directory.

NEW

```
10 PRINT CHR$(4); "PREFIX /TEST/DIR2"
20 FOR I=15 TO 0 STEP -1
30 PRINT CHR$(4); "CREATE SUBDIR.";I
40 NEXT
```

RUN

CATALOG

If you get a RANGE ERROR in response to the CATALOG, you have a bad sorter. If the CATALOG works, sort DIR2, delete about 10 of the files, and check again.

**Speaking of program selectors, International Solutions has decided to "give away" Mouse Desk.** This is their ProDOS-based, double-high-resolution Macintosh desktop mimic for 128K Apple IIes and IIcs. They say you can have the program for "free" until March 31, but they want a \$10 check to cover shipping and handling and to keep the riff-raff away (International Solutions, PO Box 3541, Saratoga, CA 95070-1541 USA). I haven't seen the program, but in the March A+ (page 34) Gary Little compares it at length with Catalyst 3.0, if you're interested in knowing more. If you send in an order, tell them Uncle DOS sent you.

**The most exciting product I saw at the Apple World Conference in January** was an interactive videodisk set-up. It consisted of two Apple Color Monitors and an Apple IIc serially connected to a black box that was connected to a videodisk player. One of the monitors was for the IIc and the other was for the player. A floppy in the IIc held an index to the videodisk in the player; the videodisk held photographs and movies from the Apollo 17 trip to the moon. Software running on the IIc allowed you to search the index on the floppy by keyword—the results of the search appeared instantly on the videodisk's monitor.

Part of the excitement was seeing television-quality pictures on an Apple Color monitor. With a little more wizardry you could have both the computer's output and the videodisk's movies appear on the same screen. The company displaying the product was Video Vision Associates Ltd, PO Box 5025, Huntington Beach, CA 92615 714-962-6126. Products like this one will make slates out of what we now call "computer-aided instruction."

Videodisk players, of course, use disks about the size of record albums and use an analog recording scheme. Consequently, there's no way to take information off of the videodisk and actually get it into the Apple. The videodisk holds pictures that it displays on the monitor, but the pictures don't pass through the computer. Instead, a floppy disk holds data about the pictures and the computer manipulates the data on the floppy and tells the videodisk player which picture to display next.

A similar device that uses a digital recording scheme is the compact-disk read-only-memory, or CD-ROM. The CD-ROM uses a much smaller disk, which is slightly smaller than a 5-1/4 inch floppy. These are the same digital compact disks that are sweeping through the music industry. Because it's digital, the data on these disks can go directly into a computer. In theory this data could include pictures, but the quality would be limited by the graphics resolution of the computer. The future may bring CD-ROMs capable of both displaying pictures like a videodisk and passing data to the computer like a floppy, but such machines aren't available now. At the moment CD-ROMs are being used primarily for text storage, which they are quite good at. One disk can hold more than half a million K—the contents of 54 10-Megabyte hard disk drives. Grolier Electronic Publishing offers the entire contents of its 20-volume encyclopedia on a single CD-ROM disk for \$199. You can get more information on it, on CD-ROM players in general, and on a CD-ROM interface card for the Apple II that includes 512K of RAM and a 68000 microprocessor from MicroTRENDS, 650 Woodfield Dr, Suite 730, Schaumburg, IL 60195.

**I also heard Alan Kay speak at Apple's conference.** Kay devised many of the elements of the "mouse interface" as a researcher for Xerox and is now a thinker for Apple. He's working on what he calls the "agent interface." Long before Apple laid claim to the mouse interface, Kay decided it was inadequate. The interface is based on selecting "objects," such as programs or files, and doing something to them. Kay says the system works fine until you have several hundred objects to work with, at which point it breaks down rapidly. Kay's agent interface embeds a genie inside your computer that looks for and brings back the objects it thinks you'll want to act on. That's the kind of data reduction I'm waiting for.

**My kids' social schedules are already packed,** but if you know some 5- to 18-year-olds with nothing to do after school you might encourage them to start a computer club sanctioned by Apple Computer Clubs International (217 Jackson St, Lowell, MA 01853 800-343-1425). These clubs are in the midst of their third annual merit competition for student members and club advisors. The competition, sponsored by Apple, is designed to encourage development of and provide recognition for useful applications of Apple computers. Winners in four categories and three age levels will win trips to Washington, DC and Apple IIc systems.

**The new Addison-Wesley version of the Apple IIc Technical Reference Manual** was finally published late last year (\$24.95, Jacob Way, Reading, MA 01867, 800-238-3801, 617-944-3700). It was apparently delayed so long so that the 3.5 ROM information could be included. The price is half of what it was when Apple itself published it. Addison-Wesley's books are available in most bookstores. Here's a complete list of the official Apple II books now available from Addison-Wesley:

The Applesoft Tutorial \$29.95 (with disk)  
 The Applesoft Basic Programmer's Reference Manual \$22.95  
 Basic Programming with ProDOS, Revised Edition \$29.95 (with disk)  
 ProDOS Technical Reference Manual \$29.95 (with disk)  
 The Apple IIe Technical Reference Manual \$24.95  
 The Apple IIc Technical Reference Manual \$24.95

And here's a list of official Apple documentation not available from Addison-Wesley—these are "free from your dealer" (if you can find a dealer who has them) or available to certified software developers from Apple technical support. Apple is also in the process of starting a "programmers club" to make it easier for anyone with an interest to get this stuff. We'll have more details when there are some.

Technical Notes—all were revised November 1985  
 ProDOS: 1 through 16  
 Pascal: 4, 10, 12, 14, 15  
 Mouse: 1 through 6  
 3.5 UniDisk: 1 through ?  
 Guide to the New Features of the Apple IIe (A2F2114)  
 Design Guidelines (A2F2116)  
 Apple IIc-IIe Delta Guide (number unknown)  
 The Apple Interface Manual (no number)  
 Apple IIe Enhancement Programmer's Guide (A2L2022)  
 Apple IIc Programmer's Guide to the 3.5 ROM (A2L4037)  
 Apple II Memory Expansion Card Programmer's Guide (number unknown)

Two new items have also shown up on Apple's price list for software developers but I haven't yet found anyone who knows what they are. They're called *Support Training Library Manual—Apple II* (\$70) and *Support*

*Training Library Manual—Peripherals* (\$55). There may also be other documentation available from Apple—let us know if you've seen something good that's not on this list.

**Have you heard the one about the Toronto suburb** that closed down a computer bulletin board last October because it was a "business" in an area zoned for residential dwellings only? "This thing just sits quietly in the basement and people phone," says David Bradley, the system operator. North York officials respond that only doctors, dentists, and music teachers are allowed to make money at home. There goes telecommuting.

**You can get a U.S. income tax spreadsheet for AppleWorks** from Douglas Smith, 3501 E Yacht Dr, Long Beach, NC 28461, for \$29.95. This 53K spreadsheet (requires a 55K or larger desktop) includes forms 1040, 2106 (employee business expenses), 2441 (child care credit), 3902 (moving expense adjustment), 4562 (depreciation), and schedules A (itemized deductions), B (interest and dividends), C (business profits), D (capital gains), E (supplemental income), G (income averaging), R (care of elderly credit), SE (self-employment tax), and W (working couple credit).

Of course you still have to do the hard part yourself—gather together all the numbers that need to go on the forms and figure out where they go. Once you've done that, however, the spreadsheet will do all your calculating and table look-ups. It will also print out clones of the IRS forms. You'll have to transfer the 1040 results to a real form 1040, because the IRS will take nothing less. However, the IRS will accept computer print outs of all the other forms *if* the print out is a reasonable facsimile of the form (Smith's are), *if* you use 20 pound or heavier green-bar paper, and *if* you are lucky. For more information, get IRS publication 1167 (Forms Distribution Center, PO Box 25866, Richmond, VA 23260). Now just get your state income tax figured out by April 15 and you're done for this year.

**Open-Apple has a policy of exchanging newsletters** with any Apple user group that wants to. From *The Harvest*, (Northern Illinois Computer Society), I learned of a new program called the *Pollyanna Word & Phrase Checker*. The program is meant to be used by magazine editors. It takes software reviews written by freelancers and marks negative words and phrases for change later. Or it can select a more positive word or phrase from its own lexicon. The review was written by Ann Baldrige, who indicates that the program has been used successfully on reviews printed over her byline. She says even the slightest hint of anything negative disappeared from the reviews.

**Back in 1981 a company called Saturn Systems** shattered the 64K "limit" of the Apple II with a product called the Saturn 32K RAM board. Later it came out with 64K and 128K versions of the board; then the company seemed to disappear. Meanwhile, a company called Titan Technologies started selling Neptune RAM boards, which I always figured must be a Saturn clone. Wrong. It turns out Saturn Systems underwent a name change in 1984 because somebody else had already been using that name. (If you intend to start a computer company, do it soon—I calculate there are only a hundred or so names left that haven't already been trademarked.)

The Saturn RAM boards, which go in standard Apple slots, became Titan RAM boards. They are still available and supported in 32K, 64K, and 128K versions. The Neptune RAM board is Titan's product for the IIe's auxiliary slot. It's available in 64K, 128K, and 192K versions. Titan's premier product nowadays, however, is its Accelerator IIe, a high-speed 65C02 co-processor card. Titan has also recently introduced a card called the "Titan III Plus IIe," which, for \$500, will let your Apple III run 128K Apple IIe software. Sometimes it's funny how history works out (310 W Ann St, Ann Arbor, MI 48104-1337 313-662-7777).

**As enamoured as I am of speed, you'd think I'd be wild about Forth,** the computer language of reverse Polish notation. Unfortunately I've never liked doing math backwards. But if raw speed on an Apple II is what you're looking for, don't forget the MicroSpeed system from Applied Informatics (15104 St Thomas Church Rd, Upper Marlboro, MD 20772 301-627-6650). Uncle DOS ran into these folks this month while unsuccessfully looking for ways to get double-precision math out of Applesoft (see this month's letters). Applied Informatics couldn't help with double-precision, but they reminded him that their combination of an arithmetic coprocessor card and DOS 3.3-based Forth software can put minicomputer number-crunching power in an Apple II for under \$400. The system also comes with Applesoft and Pascal booster disks that allow those languages to take advantage of the arithmetic co-processor. The Applesoft booster is compatible with Roger Wagner Software's *SpeedStar* compiler (PO Box 582, Santee CA 92071 619-562-3670) and any of the various cards like Titan's Accelerator IIe for those of you who want to make time go backwards.



## Ask (or tell) Uncle DOS

### Promises, promises

Unless you are privy to more information than you reveal, the "common sense" you see at Apple (February 86, page 2.1) looks like it's based on a politician's promise to me. All I hear are glittering generalities—no announcements, no plans, no word they are working on anything specific—just vague pep talks.

Wasn't Sculley around when Apple did away with mail order Apple products, foisted the dealer service departments on us by shutting down all service lines at Apple (promoting dealers who for the most part were bummers), and tried to kill the II-Plus and IIc while shoving the IIc down our throats?

Wasn't Sculley around when Apple made it almost impossible for many II and II-Plus owners to move to a IIc by selling only "systems" to dealers? Dealers have been unwilling or unable to unbundle these systems at a reasonable price for those of us who already own disk drives and monitors.

It can't be all Steve Jobs doing. I just don't see Sculley as a hero. Maybe the almighty buck will make them improve the IIc but their loyalty to their customers won't.

Bob Brown  
Garden Grove, Calif.

Your letter makes a nice counterpoint to last month's article on the January Apple World Conference. I may have been a bit dazzled, but I don't think so. The real changes I saw were Apple's new attitude toward user groups, equipment upgrades, and trade-in policies. They will have to do a lot more than that, obviously, to become a truly customer-oriented company. But I detect a change and I think it started when Sculley broke with Jobs last year.

No company as large as Apple is a monolith. There are people inside Apple who have great respect for the Apple II and there are people inside Apple who classify it with the Sinclair. I see a new respect for the Apple II at the highest levels of Apple. I **did not** see that same respect in the Apple Evangelist group, which conducted January's subconference for software developers. Apple is a big ship—it will take awhile to turn it around. But I think Sculley, Yocam, and Gasse are on the bridge and have turned the wheel.

Several people have asked whether I "really" know more than I let on last month. I don't. Please take last month's predictions with all the grains of salt I laced them with.

### watts = amps \* volts

In your January 1986 issue you say (page 98), "if the rating is in amps, multiply it by 120 to obtain

watts." You should be more careful in a publication aimed at a world-wide audience. Hereabouts you had better multiply by 250 to obtain watts.

Ken Ozanne  
Faulconbridge, NSW Australia

*That was a bit ethnocentric of me, wasn't it—particularly since we in the U.S. are nearly as alone in our voltages as we are in our feet, inches, quarts, pounds, and Fahrenheit's. However, we do drive on the right side of the road.*

### Word processing and RAM cards today

I am searching for the ultimate IIc word processor. I need one that is capable of handling a 300-page document (in RAM—I have a RAMWorks card) and that will automatically do footnotes and endnotes, an index, a table of contents, and check spelling. My first choice is the IBM version of *WordPerfect*, but I don't want to abandon my IIc.

Peter A. Lemettais  
Houston, Texas

*There is a new Apple II version of **WordPerfect** you should look at (Satellite Software, 288 W Center St, Orem UT 84507, \$179.95). The size of the file this word processor uses is limited only by the size of your disk. The program lacks a few of the features you're looking for, but otherwise looks very good. There are lots of word processors available for the Apple II, do we have any reader nominations for the ultimate?*

*The primary feature I look for in any piece of Apple software is speed. If I could put up with ten-second response times I'd use a mainframe instead of an Apple II. But I can't stand to wait for a computer. Normally this means I don't give programs that manipulate disk-based data, such as **WordPerfect**, a second look.*

*However, there are so many different ways to add memory to an Apple II that it has become a programmer's nightmare to try to use memory beyond 64K in any way other than as a RAMdisk. Apple's new memory card, which is most easily used this way, has added to the RAMdisk momentum. Thus, if you configure your **RAMWorks** card as a RAMdisk and tell **WordPerfect** to use it for storing the file it is working on, you may have nearly the speed of a RAM-based, rather than disk-based, word processor.*

*Look for more software to adopt this scheme for using additional memory. If a program is written for 64K and a RAMdisk rather than 128K it can run (albeit very slowly) on Apples without RAMdisks and on RAMdisk-equipped Apple IIs and II-Pluses as well as IIs and IIcs. I expect the built-in memory in next-generation Apple IIs to look like an Apple memory card—that is, a RAMdisk—when the computer runs in 6502 mode. So such software should also run on these machines. All this makes 64K plus RAMdisk THE configuration to write software for, in my opinion.*

*Along these lines, we have gotten several questions about the availability of ProDOS RAMdisk drivers for older RAMcards. The market seems ripe for a do-all ProDOS driver that would recognize any and all RAMcards in a computer and turn them all into one large /RAM volume. We don't know of one of those yet. But for those of you who've asked about specific products, we've found a ProDOS driver for the **Flashcard** from Alfred Steele (Box 6296, APO New York 09012, \$15—money orders only) and for **Nep-tune** and **Saturn** RAMcards and work-alikes from*

Steven Humpage (2427 NE 24th Ave, Portland, OR 97212, \$20)

### Enhancing track 0

This is a follow-up to my January letter (pages 103-104) concerning the track 0 crash problem on enhanced IIs. The problem is still with us and still unresolved, but Apple is now interested.

A local Detroit representative came to our school and took two of our troublesome machines and left us two new ones to try. He told us that Apple is not aware of this problem occurring anywhere else. He said that an Apple service technician in Cupertino "is going crazy" looking for the problem and that Apple is very concerned.

I had not yet received the February issue of **Open-Apple** when he was here, but I was able to tell him that I had received two calls from people who read the original letter and who had the same problem.

What I learned from these two calls would seem to put into doubt Ken Kashmarek's idea (February issue, page 2.6) that the problem is enhanced *drives*, although doubting Ken's wisdom seems like heresy to me. The first call came from a user in Richmond, Virginia who has an enhanced IIc with Rana drives and an Applied Engineering RAMWorks card. The second came from a Washington, D.C. user who has an enhanced IIc with Comrex drives and the Checkmate MultiRam card. Both are having problems with ProDOS applications. Our own computers have Apple UniDisks and Apple 128K cards. We also tried older Disk IIs on a couple of our computers and they still crashed. The problem would seem to be the enhanced IIc and ProDOS combination.

When we find out what the problem is I'll get the information to you so you can inform your readers. Thanks for your concern. When I decided to try to deal with this problem publicly, there was no question as to which Apple publication to use. I made the right choice.

J. Ernest Cooper  
Lathrup Village, Mich.

*We continue to get a letter now and then about this problem. It is definitely not isolated to your school, but neither is it widespread. All I can say now is that I'm real glad I'm not the technician who's trying to figure it out.*

### Back up the creek

Software warranties look really stupid in combination with copy protection. Many companies offer to replace original disks should they become defective. How am I supposed to know whether the disk is defective when it's not even guaranteed to do anything?

Bobby Woolf  
Gulf Breeze, Fla.

### Getting IIc input

The answer to this question has probably been discussed at length somewhere, or else is so obvious that it hasn't, but either way I've missed it.

Apple says that programmers should not use the standard Applesoft INPUT and GET statements with the Apple 80-column card activated, but I can't find anything that explains why. Is it related to the way characters on the 80-column screen are stored in the two banks of memory? If so, does that mean the problems occur when INPUT and GET are used for input from the keyboard, but not from files on disk (or from a modem)?

Also, in the Applesoft tutorial manual, Apple included what it calls an underline cursor input routine, which is interesting but extremely slow. Apple said a couple of years ago it would be releasing a machine-language version—do you know whether one ever appeared?

Eric Schoch  
Indianapolis, Indiana

You aren't the only one who has missed answers to these questions. I've been waiting for them for a long time myself, with no results. I finally turned your letter over to Dennis Doms, a local malcontent who has written letters to **Call A.P.P.L.E** and **Byte** complaining about Apple's silence on these issues, and here's what he reports:

The primary problem with the original IIe's use of INPUT and GET was the way it handled escape sequences and control characters.

Within INPUT, using escape or control characters would result in their immediate execution. This could change the formatting of the screen or even disable the firmware. In addition, a bug in the routine that scanned the IIe's internal escape table usually caused the sequence ESC control-L to bomb the system.

With GET, the escape character had problems because the original IIe firmware detected the escape before the Applesoft GET did. That meant it was impossible to GET an ESC while the 80-column firmware was enabled—a rather significant limitation. Most of these restrictions also apply to INPUT and GET from disk files (see September's "Input Absolutely Anything", pages 65-68 and 74, for a fix), although usually the problem characters do not appear in text files.

The good news is that INPUT and GET seem to work properly with IIc and enhanced IIe firmware. Most of the output anomalies, such as comma-tapping, have also been fixed.

The bad news is that there are additional anomalies in the original IIe firmware that the documentation doesn't even mention, such as the fact that the left and right edges of any window in 80-column mode are "rounded down" to occur at even-numbered columns. The only example of 80-column programming Apple has given us is MAGIC MENU, an Applesoft program in the IIe version of **The Applesoft Tutorial**. Few programmers who learned Applesoft on a II-Plus ever discovered MAGIC MENU, however, since they were unlikely to buy an updated tutorial. No such information was included in its proper place—within the 80-column documentation.

The MAGIC MENU program and protocol were developed by Apple's Bruce Tognazzini. MAGIC MENU was Apple's original definition of the now-famous interface used in AppleWorks. In addition to demonstrating how to implement menus, the program includes the blinking underline cursor and provisions for the open- and closed-apple keys and lower-case characters—things new on the IIe that are not expected or directly supported by Applesoft.

I complained about the extreme lack of information on Apple's 80-column firmware in a letter that appeared in the December 1985 issue of **Call A.P.P.L.E**.

I wrote the letter in response to a letter from Tognazzini that appeared in the July 1985 issue, in which he called on programmers to use "published entry points." I have been trying to extract the 80-column entry points from Apple since 1983 with no real results.

I have also been trying for months to obtain the

"6502-based version of the input routine" that Apple said it would be releasing "soon" in 1983. I have received no definitive reply to written inquiries to Apple regarding the routine itself or its release status, and I have reason to believe Apple has never officially released it. Apple does have the routine; as Christopher Hogue pointed out in **Open-Apple** (December, page 96) it's in the SU2.OBJ file found on the Systems Utilities disks for the IIc and the 3.5 inch UniDisk. The only place I have seen any documentation for that routine is in **Open-Apple**.

## Unpublished entry point found

My favorite data base program worked fine on my Apple II-Plus for all the years that I had it. "Upgrading" to an enhanced Apple IIe caused problems where the files had long records that needed a second page to display all the fields. Paging forward to the second page always bombed the program. I figured out that the program used a CALL to a machine language routine that JSRs to a ROM routine at \$FC5A to clear part of the screen. Guess what? That routine apparently doesn't live there any more. Because I have access to a non-protected copy of the program, I was able to get around the glitch with Applesoft commands that set the window size before using the Applesoft HOME, then restoring the old window. Maybe others could use this information.

Wayne Mitchell  
Buffalo Grove, Ill.

You did some good detective work to track down the source of the problem.

The monitor's HOME routine actually exists at \$FC58. \$FC5A is what is classically defined as an "undocumented entry point." On the original Apple II ROMs, entry at this location caused the screen to be cleared from the screen line number indicated by the contents of the 6502 accumulator to the bottom of the screen.

A look at the monitor ROM listings in the Apple II Reference Manual reveals the \$FC5A entry point was not even graced with an assembler label, which implies that it was not meant to be used as a "callable" routine. And therein lies the rub.

The new capabilities of the Apple IIe and IIc required revisions in the monitor ROM. When these revisions were done, Apple retained only the 40 or so "documented" entry points (such as HOME) within the old II monitor. Apple II software that uses ROM calls and that will still run correctly on the IIe and IIc uses only these entry points.

Use of other entry points causes problems. The formerly valid call to HOME (CALL -936 from Applesoft) is still valid because the entry point for HOME is at the same location in all versions of the Apple II. The actual code for HOME has, however, been moved in later versions, and \$FC5A no longer does what it used to do.

I asked the malcontented Doms to comment on your letter, too. His answer:

What I've been complaining about is Apple's failure to provide a list of "published" entry points and approved programming methods for the IIe 80-column card. Problems arise on IIe programs that use the 80-column firmware because there is no information from Apple on which entry points are immutable. People tend to work from the labels found in the source listings in the IIe reference manual.

Then, after providing zero information to programmers, Apple chose to move the 80-column

routines around in the enhanced IIe ROMs and **blame programmers for incompatibilities**. Some necessary functions, such as the routine for picking characters off of the 80-column screen, were omitted from the original IIe command set and had to be called directly; the only other option would be to write an independent set of screen I/O routines, which would seem to negate the function of the firmware.

This was Apple's fault; they gave us no other way to accomplish a necessary function and then later decided to relocate it without warning. Some very competent programmers were caught.

On the other hand, Tognazzini is correct in lambasting programmers who use entry points that are clearly fair game for relocation, and \$FC5A is a good example. In this case there is not even a label in the II-Plus ROM source code (which, unlike the pathetic policy with the IIe and IIc, was properly supplied with the Apple II) to defend the original programmer's decision to use it.

## A new SU2.OBJ

I was very interested to see the letter from Christopher Hogue in the December **Open-Apple** (page 96) about SU2.OBJ. It set me to investigating the SU2.OBJ file on the new System Utilities disk that comes with the 3.5 inch UniDisk. One of the obvious things from looking at SU2E, the main Applesoft program that starts up SU2.OBJ, is that HIMEM must be set to 32256 before SU2.OBJ is BRUN.

I did a benchmark on the &PRINT command and found it to be slower than regular Applesoft. Applesoft was 1.13 times faster. I could not find any advantage to using &PRINT. The uses of it in SU2E all seem to be just like the use of Applesoft's PRINT, plus requiring for an HTAB that you POKE HL (which is 1403 for 80 columns or 36 for 40) with what would be your HTAB value minus one. How inconvenient! There must be some point to &PRINT but I haven't found it.

Hogue implied that ES had to be set to 1 before ESC could be used to exit from an &INPUT. It doesn't. If the command is terminated with ESC then ES=1; if with RETURN then ES=0. Similarly if &INPUT is terminated with open-apple or solid-apple after giving values to OP\$ or SO\$, then OA=1 if terminated with open-apple and SA=1 if terminated with solid-apple.

In the table within SU2.OBJ where the Applesoft variables used by the routine are listed, there is also a variable called CT. I haven't found out why. In SU2E, CT appears only in lines 2000 and 2030.

Please publish any and all additional info on the routine. What a find.

Stephen Bach  
Scottsville, Va.

The SU2.OBJ file provided with the new 3.5 system utilities is different from the one that comes with the IIc. I don't know the full extent of the differences, but a major one, as you point out, is that the start up program must do a HIMEM:32256 (\$7E00) to protect the routine from being overwritten. This means the newer SU2.OBJ **doesn't follow Apple's own protocol** for obtaining memory space. In the IIc version, the program that starts up SU2.OBJ includes a HIMEM:32000 (\$7D00) command, but that SU2.OBJ also moves HIMEM itself (to 34560, \$8700) automatically.

The purpose of &PRINT is explained in the next letter. CT appears to be a count for the number of options available under the "Advanced Activities" heading on the system utilities menu. CT is 4 for the

Ile and 5 for the IIc. The IIc has the additional activity of "Setting the Serial Ports." Note that if you run the 3.5 system utilities on a IIc, the main Applesoft program used is SU2C.

You are right about ES. Hogue probably thought ES had to be preset to 1 because that's how the IIc system utility program made the call.

OA, SA, OP\$, and SO\$ were designed to capture calls for "help". For example, OP\$ can be set to a string containing a set of displayable (non-control) non-alphabetic characters such as "?/". If the open-apple key is held down in conjunction with any single character within the string (such as the question mark) while &INPUT is waiting for input, &INPUT terminates with OA set to 1. Otherwise, OA remains set to 0. SA and SO\$ work the same way. To illustrate the setup:

```
10 OP$ = "1234" : REM Define exit characters
20 &INPUT AS$
30 IF OA = 1 THEN GOSUB 1000 : REM 1000 enters HELP
```

If the open-apple key is pressed along with any of the keys "1," "2," "4," or "\$" in the above example, the program will GOSUB 1000.

Here's a final goodie about the 3.5 version of the system utilities. Ken Kashmarek tells us the COPY FILES option will copy Pascal files to and from DOS and ProDOS (or other Pascal) disks. The IIc version recognizes a Pascal disk, but will not convert files between these formats.

## SU2.OBJ follow up

Thanks for publishing my letter about the IIc System Utilities file SU2.OBJ. Since then I have discovered the nature of the &PRINT command. It provides automatic word-wrapping of long strings on both 40- and 80-column screens.

The final command, &EXIT, removes SU2.OBJ from memory, thus giving up room for other machine language applications. To restart the file, it must be BRUN from the disk again.

I found one serious bug in the program, but like all good bugs it only occurs under the right conditions and I have not yet determined what they are. The bug is that the first letter of an &INPUTed string sometimes disappears.

For example, I have a program that contains a segment similar to this one:

```
510 HOME
520 &INPUT AS$
530 BS=AS$
540 PRINT AS$
550 PRINT BS
```

If you enter the word "wimp" in response to the blinking underline cursor that appears when this segment executes, the program will print the following:

```
imp
imp
```

I have found that adding the following lines to the program gives better results:

```
535 CS=AS$
560 PRINT CS
```

With the same input, the program now gives us:

```
imp
imp
wimp
```

Unfortunately, this program segment by itself will not show the bug. It only occurs in a larger program and I have no logical explanation for why.

I'd also like to hear your reaction to the advertisement on page 76-77 of the January Byte. If you haven't seen it, it's about a new storage medium for computers, called paper.

Christopher W. Hogue  
306 Burdick Crescent  
Tecumseh, ONT N8N 2E8 CANADA

Thanks for the additional research on the SU2.OBJ file. We couldn't duplicate the &INPUT bug you found. It probably has something to do with strings crossing a "page boundary" in memory. I've included your complete address this time so that other readers interested in SU2.OBJ can contact you directly if they want.

The Byte ad you mention is for the Cauzin Systems Softstrip reader (835 S Main St, Waterbury, CT 06706, 203-573-0150). It is a \$200 piece of equipment that can read stripes of specially coded data printed on paper. Cauzin hopes computer magazine and book publishers start using these strips to encode programs. I can't personally imagine buying one for that, however, since I don't think the software in a year's stack of computer magazines is worth \$200. Nonetheless, I'm sure there are market niches that need a practical and efficient way to distribute small amounts of digital data from one point to thousands of other points—Cauzin's technology fits this niche perfectly.

Cauzin's reader for the IIc uses the cassette tape plug, incidentally, which avoids the problem of yet another interface card. The IIc and IBM/Macintosh versions use a serial interface. Cauzin says up to 5,500 bytes can be encoded, on their equipment, in a single strip. Cauzin promises software that would allow dot-matrix printers to generate 500- to 1,000-byte strips and laser printers 3,400-byte strips.

Incidentally, this isn't the first time such a method has been tried for program distribution; Byte used to publish programs in an encoded form called PaperBytes. Of course, before that there were paper tape and card readers. The difference here is one of density—the Softstrip appears to be able to store a lot more information in less space, making paper a practical medium for the distribution of digital data.

## AppleWorks DB problems

When I try to convert an AppleWorks word processing file into an ASCII text file, load it into a data base file, and then merge the new DB file to an existing DB file, funny things happen and it usually either writes over old data, comes up with garbage, or locks up. Ever hear of this? Any guess as to the problem?

John R. Eiseman  
Ladue, MO

In using the AppleWorks data base, I have found that the time and date fields will not arrange properly when information has been brought into the data base from an ASCII text file. The data base will arrange the information in alphabetical order but not in time and date order.

Any suggestions as to how the text file information must be written so when the file is put into an existing data base it will allow for proper sorting of date and time fields?

E. Wayne Carter  
Louisville, Ky.

The only way AppleWorks allows you to load an ASCII text file into the AppleWorks data base is by making a new DB file "from scratch." AppleWorks will ask you how many categories are in each record,

then load the file. For this to work each record in your ASCII file **must** have the same number of categories (in this case a "category" means a sequence of zero to 78 characters followed by a carriage return). If one of your records is missing a category (or has an extra one), all the following records will have the data in the wrong place.

The easiest way to check whether all your records have the right number of categories is to simply load the file into the data base and check it. If a record has a missing or an extra category, it will be obvious and easy to find. Open-apple-Q (quick change) back to your word processor file, fix it, resave and reload the ASCII file, and check again.

Once you have the file loaded correctly, you merge it with another file by using the clipboard. When doing this, it is essential to remember that categories go to and from the clipboard in the order in which they were originally named. If you have changed the layout of the screen since you named the categories this will be extremely confusing. Changing the layout of the screen—in either single-record or multiple-record format—**does not** change the order in which categories go onto or come off the clipboard. To see the clipboard order, use open-apple-N(ame) on both of the files you want to merge. Unless the files have the same basic design, you will find it impossible to merge them correctly.

AppleWorks recognizes "date" and "time" fields by the presence of those words in the category name. But when you load an ASCII file, you are not allowed to name the categories until after the file is loaded. Consequently, fields with dates in them are not given the special treatment they deserve when the file is loaded and are not recognized for what they are.

To correct this, use open-apple-N(ame) to correct the category name. Then use open-apple-L(ayout) to get the category with the date to appear on the screen in multiple-record layout. When leaving L(ayout), tell AppleWorks you want the cursor to go down the screen. Then, using the insert cursor, insert a space at the beginning of each date and press return. The date will be corrected. By repeatedly pressing the space bar and return key you can correct about a hundred dates a minute.

None of these problems should cause a computer to lock up, however. I've never experienced lock up when doing these kinds of manipulations (although I have experienced it when deleting report formats). Lock up could be caused by exotic software bugs, of course, but it's also an indication of hardware problems.

In the October Open-Apple (page 77) Jim Luther reported a possible problem with the 74LS245 chip used in some Apple IIes. Luther reports he has gotten some additional information from Applied Engineering on how to test for this problem—load files onto the AppleWorks desktop until you've filled more than 50K of memory. Delete the files one at a time **in the order that you originally loaded them**. This causes heavy use of the 74LS245 by the IIc memory circuits. If the computer does weird things as you delete the files, then the 74LS245 is the likely problem.

## Toward a perfect Basic

I develop applications for track and rally racing and I'm constantly improving them. Most car racing in the Greek Championship and one rally in this year's European Championship have had their results tabulated by an Apple IIc, Profile hard disk, and one

of the modules I've developed (Rally Master, Speed Master, Hill Master). I believe that I have exhausted most of Applesoft's capabilities and I'm using a lot of calls and machine language sorting routines to speed up the programs. However, their sheer size and the lack of many commands in Applesoft make a large burden when you try to redesign parts of badly structured programs (renumbering, moving blocks of code). I'm sure a lot of amateur developers have reached the same point.

I'd like to see an improved version of Basic for the Apple II with IF-THEN-ELSE-ENDIF, WHILE-WEND, REPEAT-UNTIL, and CASE. It should PERFORM defined procedures. No more line numbers, please! No control-D for disk access commands, either. I'd like GLOBAL and LOCAL variables. I'm running out of variable names. Why not allow full names that make sense? STARTIME\$ and STOPTIME\$, although legal in Applesoft, are considered the same string because the first two letters match.

I'd also like to be able to CLEAR and STORE selected variables, strings, and arrays. With the existing commands I have to STORE a 70 sector VAR file in order to preserve just a couple of variables. Likewise, I'd like a RESTORE that will not clear all variables, strings, or arrays in the middle of a program.

I'd like a real interpreter that's 100 per cent compatible with existing Applesoft programs and ProDOS. Is that too much to ask?

Meanwhile, can you please tell me how I can preserve a couple of variables while passing from one program to another? STORE won't work because it saves all variables, both strings and arrays. In addition, RESTORE not only fills the next program with many unneeded strings and arrays, any new variables are cleared. What I do now is OPEN a sequential text file, WRITE the variables I want to preserve (only two in my case), CLOSE the file, then RUN the next program. The first lines in the new program OPEN this file, READ the variables, DELETE the file, and the task is accomplished. However, this is a time consuming procedure even if you use /RAM for the file and too much code is added. Is there any absolutely safe address to POKE the values of the variables, then RUN the next program and PEEK them up?

I also want to pass some experience on to fellow developers. My library is chock full of utilities bought from various software houses. The absolute Applesoft editor is GPLE from Beagle Brothers. I have used it for three years and find it indispensable. It is bulletproof, has all the functions anybody would need and it gives you total control. If only it had renumbering and merging. EXTRA.K by the same house is a must if you want to use the extra 64K in your extended 80-column card. PRO-BYTER and D CODE offer great help too. QSORT, a utility in PRO-BYTER, will knock your socks off with its sorting speed.

And a word of caution: if you use APA (Apple's own renumbering and merging utility) be sure you save your original version before fiddling with it. And after renumbering or merging, save the new version under a different name and run it or list it to be sure that everything is OK. I have found that many times my renumbered version is totally scrambled or messed up with unintelligible code.

Nick Andritsakis  
Paleon Phaleron, Greece

**Blankenship Basic** has many of the features you're looking for. It's available in either DOS 3.3 or ProDOS versions for \$25 (both for \$39.95 plus \$2 postage within the U.S., \$5 postage outside the U.S.)

from Blankenship and Associates, PO Box 47934, Atlanta, Ga. 30362. **Blankenship Basic** interfaces with Applesoft by means of a set of ampersand commands. A built-in editor keeps your program listings from being cluttered with obscure ampersands, however, and improves the listing format. There's a complete review of the program in the January 1986 Call **A.P.P.L.E.** (page 43).

**ZBasic** (Zedcor, 4500 E Speedway #93, Tucson, AZ 95712 \$89.95 plus \$5 postage in the U.S., \$20 postage elsewhere) is another version of Basic with many of the features you're looking for. We've had one subscriber report that the documentation of this product is full of bugs and inconsistencies and that technical support is nearly impossible to obtain. However, the January issue of the journal of the Northern Illinois Computer Society had a positive review of the program (three-and-a-half out of four stars). The big selling point of **ZBasic** is that programs are portable between different kinds of computers, including Apples, Macintoshes, IBM and compatibles, CP/M 80 machines, and several Radio Shack models.

In addition to these, there are other "advanced-Applesofts" for sale and under development—some of those being developed take advantage of the power available on the 65802 and 65816 chips.

Using a RAMdisk-based text file to pass a few variables between programs is probably about the fastest way to do it. Poking data into a "safe" memory area (even if you could find one) would probably take as long or longer—PEEK and POKE are pretty pokey. The **ProntoDOS** manual gives some tricks for saving numerical arrays in binary files. The information is not as complete as it should be, however, and is too long to reprint here. I'll try to do a complete article on this trick some month soon—it may be exactly what you're looking for.

## More math digits

I would appreciate information on how to get double precision arithmetic from my Apple IIe and DOS 3.3.

Norman Fine  
East Brunswick, N.J.

Until now Applesoft has been adequate for my purpose. But now I find that I need double precision arithmetic. I read about SANE in the July 1985 Call **A.P.P.L.E.** (page 19) but this is available for use only with assembly language and Pascal. I do not intend to learn either of these and I can't abide the Macintosh.

Is there any prospect that some form of extended BASIC will be available for the Apple II? Or, for that matter, is there some means by where the II can run, for instance, IBM BASIC?

I find it hard to believe that this problem has not been addressed before. Apparently engineering and scientific applications are done on IBM, Hewlett-Packard, etc. computers. Macintosh BASIC is not intended for use by anyone.

John Erickson  
Seattle, Wash.

Applesoft's math routines use 5 bytes to represent each floating-point number. This gives nine significant decimal digits when the numbers are converted to decimal representation. This compares to six significant digits in Apple Pascal and 19 significant digits in Apple's SANE package for Pascal and assembly language.

To the best of my knowledge, no one has come up with a way to use SANE and Applesoft together.

Hayden Software once sold a package called "Double Precision Floating Point for Applesoft." However, they don't anymore. Does anyone out there know whether it's available elsewhere?

If you're willing to move away from Applesoft, **ZBasic** (see previous letter) allows you up to 54 digits of accuracy—you specify how much you want. Obviously, the more accuracy you want the less speed you will have. If speed was your problem, we could tell you about California Computer System's 7811B arithmetic processor card. For about \$300 you get the card and Applesoft support software. However, while this card makes math fast, it actually has fewer significant digits than Applesoft does. (California Computer System's customer service phone is 408-263-4963.)

A final alternative is adding a co-processor board and using a Basic from another operating system. Microsoft's Apple CP/M version of **MBASIC** includes both single (4-byte) and double (8-byte) precision floating-point data types. Personal Computer Products, Inc. (619-485-6411) sells a coprocessor board for the Apple called the 88Card, which includes MBASIC running under MS-DOS.

## Fewer math digits

I am having problems with multiple subtractions in an Applesoft program. For example, PRINT (72.27 - 8.04) - 7.61 gives me 56.6200001. PRINT (1.36 - .42) - .94, which should be zero, comes up with 5.82076609E-10. What can I do to solve this problem?

Linda Bunney  
Little Rock, Ark.

Another subscriber sent in a similar example. Type it in and watch what happens:

```
FOR I = 0 TO 7 STEP .01 : PRINT I : NEXT
```

In each case the problem is related to Applesoft's conversion of numbers from decimal to binary and back to decimal and to other bugs that introduce small errors that are amplified by repeated calculations. The July 1985 issue of Call **A.P.P.L.E.** that had the article mentioned earlier on Apple's SANE package also has an excellent article on the limitations of floating-point arithmetic in Applesoft (page 15). This article was written by James W. Thomas, head of the Apple Numerics Group, which developed SANE.

Your second example is particularly good. The number you obtain, if you take it out of scientific notation, is 0.00000000582076609—and that's pretty close to zero, although it doesn't look like it at first glance.

The problems here aren't limited to Applesoft, but seem to be inherent in binary computers. Some machines use a special kind of math called "binary coded decimal" to limit these errors. The Apple II's 6502 microprocessor has built-in support for BCD math, but Applesoft doesn't use it and I don't know of any other software that does, either.

The typical way to circumvent these problems is to use the Applesoft STR\$ and VAL functions to truncate numbers back to the correct number of decimal places. For example, try these fixes:

```
FOR I=0 TO 7 STEP .01 : PRINT LEFT$(STR$(I),4) : NEXT
```

```
PRINT VAL(LEFT$(STR$(72.27-8.04),4)) - 7.61
```

You will find, however, that reliance on STR\$ and VAL will slow down your calculations and can also themselves cause even larger rounding errors when you get into multiplication and division.

## Apple DMP superscript blues

I have an Apple IIe with an Apple Dot Matrix printer and an Apricorn graphics interface card. I use *Apple Writer* for word processing and *AppleWorks* for everything else.

When I first got my computer I attempted to do superscripting and could not. I wrote to Apple and they sent me a program called *Apple Dot Matrix Printer Super/Subscripting Demonstration Diskette*. I couldn't get it to work and contacted the store from which I bought my computer. They tried the program and discovered that it would work with a "non-graphics interface card," but not with my Apricorn card.

But recently I was fooling around with the *AppleWorks* word processor and discovered that, while I don't like it as well as *Apple Writer*, it can do superscripts and subscripts on my printer. Now that I know it can be done, can you tell me how to do it from within *Apple Writer*?

Harry E. Jacobson-Beyer  
Louisville, Ky.

Since the *Apple DMP* supports reverse and fractional linefeeds, it can, in theory, easily handle superscripts and subscripts. Typically you do a superscript by commanding the printer to do a reverse half-line linefeed (which rolls the paper back half a line), printing the superscript, then doing a forward half-line feed, then resetting linefeeds to normal.

The *Apple DMP* lets you use reverse line feeds with the command "escape r", and half-line line feeds with the command "escape T12." However, when you actually issue a linefeed (control-J), the *DMP* not only rolls the paper as you have commanded, but also, without explanation, inserts a carriage return—taking the print head back to the left margin.

*AppleWorks* overpowers this flaw by keeping track of its exact horizontal position on the line at all times.

After issuing the control-J for linefeed, *AppleWorks* issues an "escape FXXXX" command, where the XXXX is a four-digit number representing the current horizontal dot position. This returns the print head to the correct place. The same thing happens when you end the superscript or subscript.

To see this in complete code, add a *Dot Matrix Printer to AppleWorks*, but specify "print to disk" when asked for a slot. Print a document to disk that contains superscripts and subscripts, then quit *AppleWorks* and enter *Apple Writer*. Load that file and you'll quickly see how much work *AppleWorks* has to do to support superscripts and subscripts on the *Apple DMP*. You can't do this with *Apple Writer* because it doesn't keep track of the horizontal dot position as it prints a line.

## ProntoRAM

Are the more-than-35-tracks-per-disk patches you gave in the April issue (page 31) compatible with *ProntoDOS*? Is there a way to permanently patch DOS so that it allows INITs disks with more than 35 tracks?

I use my IIc mainly for programming. The aspect of eventually adding more memory has intrigued me. But in *Applied Engineering's* ads for their Z-RAM, they only talk about using the card with *AppleWorks*. Would I be able to use the extra Ks when programming in DOS or *ProDOS*? If I can, will it also work with *ProntoDOS*?

Michael E Corwin  
Fresno, Calif.

The patches in the April issue are compatible with *ProntoDOS*. They automatically become part of the DOS on the newly initialized disk. The only trick is making sure you catalog with a disk that has the number of tracks you want just before initializing a new disk.

*Applied Engineering's* card, like all memory cards—including *Apple's* new one, gives the *Applesoft* programmer nothing more than a RAM disk—a very fast electronic disk drive that gets erased whenever the power goes out. All of *Applied Engineering's* memory cards now come with RAM disk software for both DOS 3.3 and *ProDOS*. All DOS 3.3 RAM disks I know of are both compatible with and speeded up by *ProntoDOS*. Assembly language programmers can use the memory cards in other ways.

## Interactive formulas

For years I have boasted to others that I had never found a task that *Applesoft* would not do; we both know the special plans the gods would have for such innocents. I now have a serious need that I have never seen discussed in any computer magazine or book. I am developing a program to tabulate and plot functions of formulas, and I can find no way to input a formula interactively into a running program. For myself, I can always break into the program and simply write the formula as a statement line of the program—but that is not satisfactory for the broader applications I desire for my students. Thank you for any tips.

David L. Atkins  
Washington, DC

*Applesoft* can do it. Look at the following articles:

William P. Miller, "Graphs 'On the Fly,'" in *Cider*, March 1986, p. 45.

Cornelius Bongers, "Execute Strings," Call -A.P.P.L.E., September 1982, p. 9.

Charles Kluepfel, "Getting the High Sine," Call -A.P.P.L.E., March 1982, p. 27.

David A. Lingwood, "Adding Lines to Running *Applesoft*," Call -A.P.P.L.E., January 1981, p. 51.

## Lama'la Kashmarek

I tried to resist, but couldn't. Clay Ruth certainly did speed up the Lam routine (February and May 1985, pages 12 and 40), but he did more than twice the work that was necessary to get there.

Ruth indicates that the time saving is in using the *Applesoft* routine MOVEIN to move the string to the input buffer. However, Ruth then proceeds to do all the work needed to move the string a second time while setting the high bits. I coded a shorter version (only 27 bytes versus 63 bytes) that moves the string and sets the high bit at the same time (the return character is not needed at the end). In this version, the data passed to the ampersand routine may be a string variable, literal, or expression. Here's how it works:

```

                                DRG $0300
0300: 20 7b DD      JSR FRMEVL   get string len in A,
0303: 20 00 E6      JSR FREFFAC  adr in INDEX
0306: AA           TAX           len to X
0307: A0 00        LDY #0      Y=0
0309: b1 5E        .1 LDA (INDEX),Y get a byte
030b: 09 00        ORA #5B0    set high bit
030d: 99 00 02     STA $200,Y  put in input buffer
0310: C8           INY           next byte
0311: CA           DEX           any left?
0312: D0 F5        BNE .1     loop till done
0314: A2 04        LDX #504   POKE 72,4 [see
0316: 86 48        STX #48    OR page 77a]
0318: 4C 70 FF     JMP $FF70  CALL -144

```

Here's an *Applesoft* program that demonstrates how to use the code and how much faster it is than the traditional Lam routine. In a similar test, I timed the standard Lam routine loading 2,048 bytes into memory. It took 98 seconds. This routine can load 2,048 bytes in one second—an improvement of 98 to 1. Not bad.

```

10 REM LAM A'LA KASHMAREK
100 BELL$=CHR$(7)
110 CS="0300:20 7b DD 20 00 E6 AA A0 00
      b1 5E 09 00 99 00 02 C8 CA
      D0 F5 A2 04 86 48 4C 70 FF
      N 03F6:00 03 N D9C6G"
200 PRINT BELL$: REM standard LAM
210 FOR I=1 TO LEN(CS) :
      POKE 511+I, ASC(MID$(CS,I))+128 :
      NEXT : POKE 72,4 : CALL -144
220 PRINT BELL$: REM Lam a'la Kashmarek
230 & CS
240 PRINT BELL$
250 END

```

So, there you have *Lama'la Kashmarek*. Hardly worth the effort to improve Ruth's routine, which was already faster than a speeding bullet, but like I said, I couldn't resist.

Ken Kashmarek  
Eldridge, Iowa

And in less than half the space—I'm glad you couldn't resist.

To make it perfectly clear, with this technique you don't have to embed the *Monitor* command you want to execute from *Applesoft* in a variable—you can also use a literal, as in &"any monitor command N D9C6G"; or an expression, as in &CS+ "N D9C6G". This is because the first two instructions in the code call *Applesoft* routines that handle references to strings inside programs. These routines convert whatever follows the ampersand into a string length and starting address that is quite easy to deal with.

## Open-Apple



is written, edited, published, and

© Copyright 1986 by  
Tom Weishaar

Most rights reserved. All software published in *Open-Apple* is hereby placed in the public domain and may be copied and distributed without charge (most is available in the MAUG library on CompuServe).

*Open-Apple* is sold in an unprotected format for your convenience. You are encouraged to make back-up archival copies or easy-to-read enlarged copies for your own use without charge. You may also photocopy *Open-Apple* for distribution to others. The distribution fee is 25 cents-per-page per-copy distributed. Please pay fees monthly. Send fee payments and all other correspondence to:

**Open-Apple**  
P.O. Box 7651  
Overland Park, Kans. 66207 U.S.A.

ISSN 0885-4017. Published monthly since January 1985. World-wide prices (in U.S. dollars; airmail delivery included at no additional charge): \$24 for 1 year, \$44 for 2 years, \$60 for 3 years. All back issues are currently available for \$2 each; seven or more from any single volume \$14 (postpaid). Index mailed with the February issue. *Open-Apple* is available on disk for speech synthesizer users from Speech Enterprises, P.O. Box 7986, Houston, Texas 77270 (713-461-1666).

**WARRANTY AND LIMITATION OF LIABILITY:** I warrant that most of the information in *Open-Apple* is useful and correct, although drift and mistakes are included from time to time, usually unintentionally. Unsatisfied subscribers may return issues within 90 days of delivery for a full refund. Please include a note from your parents or children confirming that all archival copies have been destroyed. The unfulfilled portion of any paid subscription will be refunded on request. MY LIABILITY FOR ERRORS AND OMISSIONS IS LIMITED TO THIS PUBLICATION'S PURCHASE PRICE. In no case shall I or my contributors be liable for any incidental or consequential damages, nor for any damages in excess of the fees paid by a subscriber.

*Open-Apple* is neither affiliated with nor responsible for the debts of *Apple Computer, Inc.*; "tinaja questing" is a trademark of Don Lancaster.

Source Mail: TCF238 CompuServe: 70120.202 Tele.: off hook