SOFTALK

VOLUME 2 SEPTEMBER 1981 $2.00

MEGABUCK APPLES

][at the Top: Apple and Puckett

Exec Hayes

Terminal Data

About Time

Applesoft Compilers


ULTIMA
. . . the ultimate fantasy role-playing game
Designed by Lord British
Available thru your local Computer Store.
A TOP OF THE ORCHARD SOFTWARE PRODUCT
from California Pacific Computer Company

CONTENTS

SOFTALK

Chairman John Haller
Editor Margot Comstock Tommervik
Managing Editor Al Tommervik
Associate Editor Craig Stinson
Art Director Kurt A. Wahlner
Contributing Editors Roger Wagner, Jim Merritt, Peter Olivieri, Greg Tibbetts, Taylor Pohlman, Kenneth A. Landis, Jeff Mazur
Guest Reviewer Greg Voss
Coordinating Editor Melissa Milich
Editorial Assistant Linda Va Salle
Operations Manager Mary Sue Rennells
Production David Hunter
Circulation Kimberly Curling, Ron Rennells
Assistant Roberta Tommervik
Systems William V. R. Smith
Advertising Sales Al Tommervik
(213) 980-5074

Cover by Kurt Wahlner.
Composition by Photographics, Hollywood, California. Printing by Volkmuth Printers, Saint Cloud, Minnesota.
Special equipment courtesy of Computerama, Burbank, California.
Apple is a registered trademark of Apple Computer Inc., Cupertino, California.
UCSD Pascal is a trademark of the University of California at San Diego.
VisiCalc is a trademark of Personal Software, Sunnyvale, California.
SoftCard is a trademark of Microsoft, Bellevue, Washington.

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

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

SUBSCRIPTIONS: Complimentary to all owners of Apple computers in the USA. If you own an Apple but you're not receiving *Softalk*, send your name, address, and Apple serial number with a request for subscription to Softalk Circulation, 11021 Magnolia Boulevard, North Hollywood, CA 91601. *Softalk* is totally independent of Apple Computer Inc.; sending your Apple warranty card to Apple Computer will not inform *Softalk* of your existence. Non-Apple-owner subscriptions for one year: $12. For the convenience of businesses and schools in which several staff members share an Apple and would like individual copies of *Softalk*, multiple subscriptions to the same address are available at lower rates: two through five, $9 each; more than five, $7 each.

BACK ISSUES: $2 through February 1981; $2.50 thereafter. November and December 1980 and January 1981 issues are sold out. March 1981 issue is in short supply. *Softalk* will send you a back issue of your choice free (available issues only) for the name, address, and serial number of each Apple owner you can find who isn't already receiving *Softalk*.

PROBLEMS? If you haven't received your *Softalk* by the 10th of the month or if you have other problems with your subscription, Ron Rennells or Kimberly Curling can help out. Call (213) 980-5099.

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

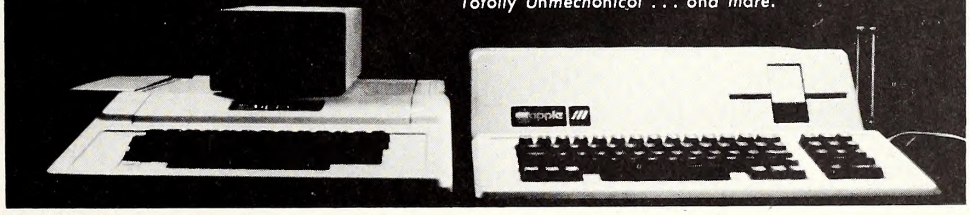
MANUSCRIPTS: All *Softalk* articles are written on assignment. If you are a writer and would like to write for *Softalk*, send statement of your association with Apples to the editor. Queries are also welcome; state the subject you have in mind and the theme of the potential story and briefly outline the direction the article would take. Include a stamped, self-addressed envelope, and keep a copy of whatever you send. Questions about rates for editorial matter will be answered only by telephone.

FEATURES

DEPARTMENTS

PREVIEWS

October Whispering . . . The Astonishing Dolphins: Learning with Apples . . . Exec Softsel: Software Distribution Comes of Age . . . Solar Architecture . . . Lower-Case Chips . . . Homemade Hardware for the Totally Unmechanical . . . and more.



ADVERTISERS INDEX

CONTEST CONTEST

Do you have a favorite kind of puzzle that you wish *Softalk* would use for a contest? Have you ever looked at our contest page and said, I could do a better contest than that?

Well, here's your chance. This month's contest is to invent a contest for *Softalk*. Grand prize is $100 worth of products from *Softalk*'s advertisers plus publication of your contest.

There'll be no runnersup—or many. Any Contest contest entry may be chosen at some point to be *Softalk*'s contest of the month. If and when your contest is chosen and published, you'll win $50. So everyone could win.

Here are the rules:

1. Create a contest that relates to the Apple in some way and that most of *Softalk*'s readers could attempt.
2. Contests should be challenging but solvable.
3. Your entry will be judged on brevity as well as completeness.
4. Winning your contest must not be dependent on earliest postmark or time limit.
5. Judging will be based on ingenuity, fun value of playing, and challenge.

Print out or type your entry just the way you'd like to see it published. Entries must be neat.

Deadline is November 1, 1981.

Send your contest, with an entry form, to Softalk Contest Contest, 11021 Magnolia Boulevard, North Hollywood, CA 91601.

Bonus Contest: Hidden somewhere in the pages of this issue is a major city's city hall. If yours is the first entry from your time zone to find it and send it to us, along with a description of where you found it and what city it represents, you'll win $25 worth of goods from your dealer. ◨

Name: _____

Address: _____

City/State: _____

Your dealer: _____

The prizes I'd like are: _____

_____

Your autograph _____

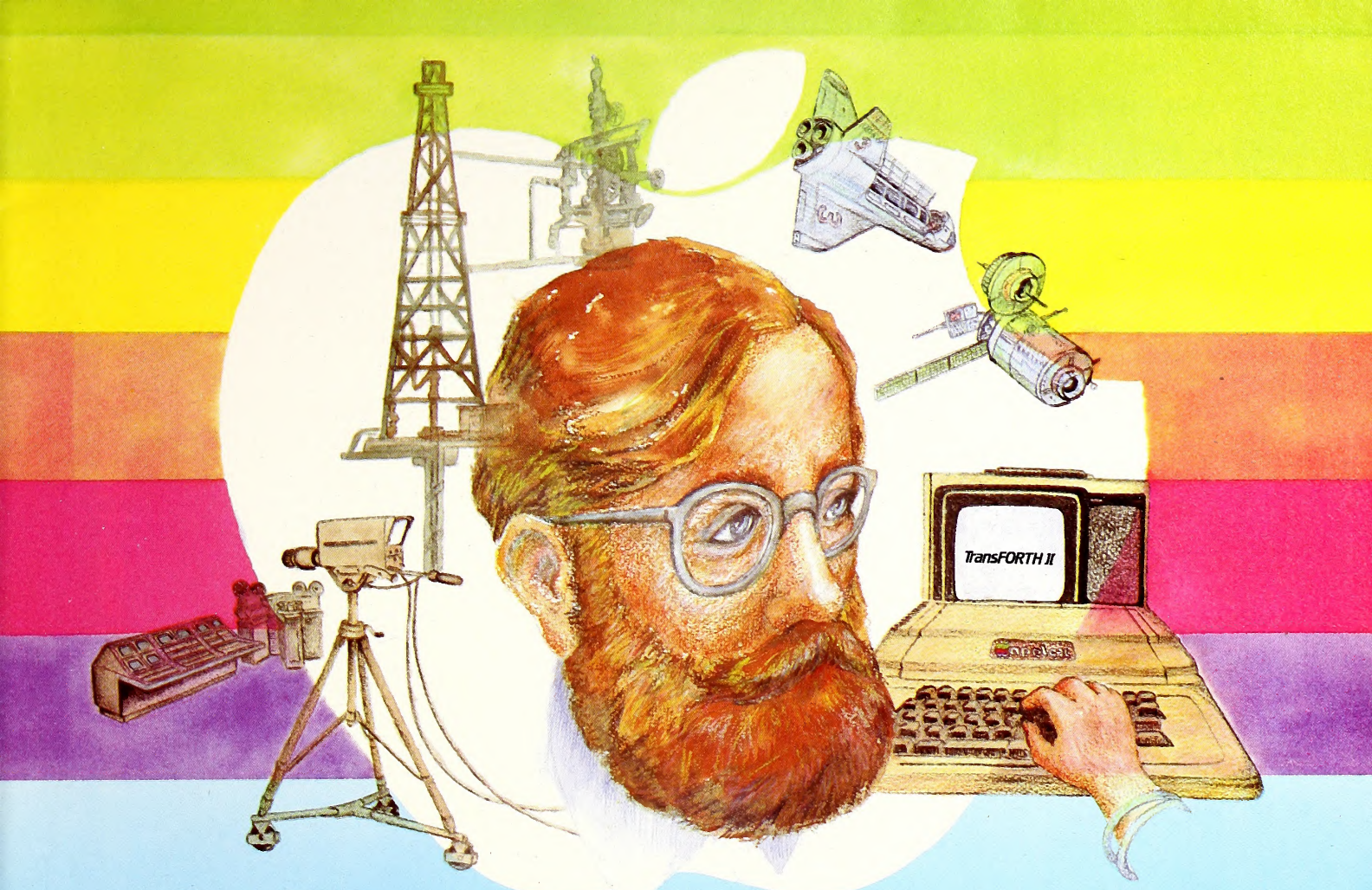
Logicians, Oracles, and Campers; Winners!

The Oracle. On July 30, 1981, Apple Computer Inc. stock closed at 24⅝, just ⅜ less than predicted by Richard Kraemer (Pewaukee, WI), who wins $100 worth of goods from his local dealer. Runnersup were Rick Jones (Spokane, WA), who predicted 23½ as the closing price, and Cliff Josephy (Brookville, NY), who foresaw 25⅜ and 26⅛. Kraemer's win brought his total score in the Oracle race a catchup-playing −48⅜. Jones would need a near miracle to rally from his current −148⅛; but Josephy, who sent numerous entries, hangs right in the running with −26½ on the first of his two runnerup entries and—29¾ on the second. These are Josephy's two best entries overall.

Jim Ganz (West Hartford, CT) now leads in the cumulative standings for the disk drive, with −13. Close behind is Daniel Tobias (Poughkeepsie, NY) with −15⅝. Others staying within a length or two are Gary Kim (Seattle, WA), −23⅜; Douglas Stewart (Cape Elizabeth, ME) and Paul Shanberg (Moraga, CA), each with −23⅝; Bill Pu (Albany, NY), −24⅜; and Thomas Murray (Hayward, CA), −25¾.

The Logical Source. Last minute hysteria is not the appropriate time to prepare a presumably logical puzzle. This is

ONTESTWINNERSCONTESTWINNERSCONT



In Language Lies The Future.

Paul Lutus creates language. Language for the future.

From him we have Apple Writer, Apple World and MusiComp. His software was used by the Viking Mars Lander team and other advanced space programs.

For the past three years he has applied himself to the development of more efficient and powerful computer languages to help him with his work. These new languages of the future are now available for your use exclusively from Insoft.

See the new world of programming from Insoft at your Apple dealer.

Each package introductory price $125.00

insoft

The languages of the future. The tools of today.

259 Barnett Rd. / Unit 3 / Medford, Or 97501 / (503) 779-2465

TransFORTH II

The final "word" in Apple programming. A brilliant extension of FORTH.

- Fully compiled
- Floating point
- Transcendental functions
- Strings and arrays
- Hires, Lores and Turtlegraphics
- Music

Far more compact and approachable than Pascal, TransFORTH II is both recursive and structured and easier to use than BASIC.

Over three years in development, this high level language enables you to program in English with far greater speed and convenience than ever before possible.

TransFORTH II. A transformation in computer programming. Available today.

ALD System II

The Assembly Language Development System. Already the preferred assembler for professional software developers.

- Object files to 18K
- Source files to 37K
- Cursor based screen editor
- Upper and lower case text entry
- Nested macro instructions
- Local, global and universal labels
- Comprehensive error trapping

The enormous file capacity of ALD System II assures you of the ability to develop virtually any imaginable software system. And you can do so with ease since ALD System II edits with the simplicity of a word processor.

ALD System II. Powerful yet convenient. The first choice of professionals.

The best comes last. Both new programs are available for the Apple II or Apple III.

CONTEST WINNERS CONTEST WINNERS CONTEST WINNERS CONTEST WIN

the lesson *Softalk* learned from June's contest. Giving credit where due, Nathaniel Hawthorne—not Edgar Allen Poe—is responsible for *The House of Seven Gables.* Apologies to our astute readers and to our American lit. professor for that slip of the mind. Top row, middle key is clearly 7—another example of our confusion. Clue y should have been $42A, not $42B. Phalanges should indeed have been digits, although many readers seem to have brought an equal lack of knowledge to that one.

Despite all this, most people solved the Logical Source correctly, and many asked for more contests of this nature. That we'll provide—next time without errors. There was also discussion about clue m. It was indeed a somewhat obscure clue, although horseracing draws a larger audience than any other spectator sport in the U.S.A. Any *Daily Racing Form* explains that one length is the equivalent of one-fifth second; thus, the answer to m was two-fifths—which indeed, in the sense of quantity of liquor, would be wealth on skid row or in the Bowery. Read as 40 percent, it's a bargain basement discount. Obscure, perhaps, but logical.

The final answers were Aristotle and A = 107. The answers to Part 3, which several readers requested, were: Q = 4; z = 5; p = 2; x = 64; u = 10; y = 1066; v = 8; n = 3; w = 11 (the eleventh playing card in a suit is the jack); m = 2/5.

Of all who answered both parts of the contest correctly, the random number generator chose Victoria Wallner (Anaheim, CA) as the winner. Wallner chose *Robot War* as her prize and will collect it and more from the Sound Room. A second prize was offered to those who answered Aristotle, whether or not they got the value of A. In this case, the random generator chose D. O. Hamilton (Lorton, VA), who, incidentally, answered both parts of the contest, and who chose Edu-Ware's *Algebra I* as prize. Hamilton's prize will come from Computers Plus in Alexandria, Virginia.

Computer Campers. The essays of the three Computer Camp contest winners appear at the end of this article. Two of the winners have been to camp and loved it. One, *Softalk*'s original winner, has not been to camp. No computer camp is willing to take him.

Todd Kimball has cerebral palsy and depends on a wheelchair for physical mobility. Despite Todd's parents' willingness to send a representative with Todd to assist him, the camps refused. It appears there are rules set down by various councils for the benefit of handicapped people—minimum facilities an organization must provide if it is to accommodate handicapped people. Without these facilities, the camps would have no chance of becoming accredited if they had a handicapped camper. Because they're all new and struggling, they need accreditation and, for the same reasons, they couldn't afford the extra facilities this year. So the amorphous rules to help handicapped people in this case kept one boy from going to camp. Perhaps some well-intentioned people need to reconsider their thinking.

The California Computer Camp at Zaca Lake expects to have the required facilities next year, and Todd Kimball's well-earned campership will be waiting to send him there. After talking with Todd and his family, we're convinced that Todd will have as much to offer Computer Camp as it has to offer him—or more.

Here are the essays from Todd Kimball, age twelve; John Brandstetter, age fourteen; and Greg Galperin, age ten.

I have been to several camps and each has had something fun, but computer camp sounds very special. It combines all my favorite activities into one camp.

Swimming is one of my favorite activities. I'm not very fast, but I can swim long distances. I also like to fish and have had some success with it. I think the reason for this is because I am patient. But most of all I like computers.

When my dad bought an Apple computer, I was very excited! It has been as much fun as I thought it would be. I like to have friends over and challenge them to games of baseball and Piero. It is much faster to do my homework on the computer. I also try to program things on it.

I am interested in learning to program computers. Programming will help me to get a job when I am a little older. For starters, I would just like to teach other kids how to program. Or maybe I could develop a program that would help kids with their school work. Eventually I could find a higher paying job with computers.

The future is in the hands of computers. People will carry pocket-size computers so they can type messages and remember figures and dates. This might lead to better decisions. This device would talk to other computers and order a person's groceries and plan exciting weekends for him. People would be able to get constant news reports or check the bargains in stores from their homes. People in the future will have very rewarding lives.

My computer gives me a good start on the future. I like my Apple and it helps me with many things. I would probably find more ways to use my Apple at this camp. I would enjoy going to this camp because I like being with other kids and making new friends. The kids could probably teach me many things about computers and we could share our experience.

I've been away from home many times and I have never gotten homesick. However at some camps I might get homesick now—though never at Computer Camp. In the past I have had only my parents, sister and dog at home, and I didn't miss them too much. Now that we have added an Apple to our family I really get homesick, but only for my Apple computer.

Todd Kimball

I would like to go to Computer Camp to learn more about computer programming. I am especially interested in computer graphics. I am currently working on a Computer Merit Badge for Boy Scouts and this would give me a great opportunity to finish it. I would also like to learn other computer languages such as COBOL and FORTRAN and PASCAL.

I am interested in learning how to program games. My parents are getting mad at me because I spend my money playing games in the Arcades. If we get any new games, I have to pay half of what they cost since I am the one who does most of the playing. So, it would really be nice if I could learn to make some Hi-Res games myself. I even have

GOTO 56

Where your secret weapon is the fourth dimension...

Sirius Software, Inc. 2011 Arden Way #2 Sacramento, CA 95825 (916) 920-1939

OPEN DISCUSSION

Clarifying Cassette Program
Thanks for printing my program—I hope it is of use to others.

Paul Kearns of Los Angeles wrote me concerning his difficulties with the program. Essentially, program line #5 must have exactly four characters following DATA. As published, it indeed shows

```
5 DATA 500
```

Mr. Kearns followed my instructions, modifying for only fifty lines, and entered

```
5 DATA 50
```

He should have used

```
5 DATA 050
```

It never occurred to me that anyone would need fewer than one hundred or more than nine hundred ninety-nine blocks. Sorry.
George Kinal, Washington, D.C.

New View of Reviews
You are to be complimented on the quality of *Softalk*. I think you have given the lie to the old adage that "you get what you pay for." Your magazine is a freebie to me, as an Apple owner, but I find it as useful as the trade publications that I buy.

There are two software reviews I wish you'd do. One of these should be on Applesoft compilers. I have seen ads recently for two such products at significantly different prices, and it would be useful to know what relation—if any—there is between price and quality for such compilers. I would also like to know something about operating problems—how much memory is required, how long a compilation takes, and what kinds of differences in speed are observed in the compiled programs that are produced.

The other software products that I would like to see reviewed are Applesoft editors. There are several of these on the market, and it would be good to know which are least troublesome to use, etc.

One aspect of your product reviews that I have appreciated is the attempt to take a critical stance. With the large number of new, relatively untested products appearing on the home computer market, it is easy to make quite expensive mistakes. Moreover, manufacturers rarely (in fact, never) tell you what their products can't do, or what compatibility problems their hardware or software are likely to offer. You offer your readers a real service by giving us information about such matters, recognizing the fact that the world is not in Apple pie order.
John Figueras, Victor, NY

Compiled Tip
With all the new compilers available now and all the new users trying them out, there has been a lot of confusion about using assembled program names to mark compiled programs. I've found it works very well to label a compiled program by simply keying in "C. Program Name." It works like a charm.
Steve Hines, The Computer Store, Santa Monica, CA

Overnight Assembler
Here is my story. Some time ago, I wanted to learn assembly language because I heard it made my Applesoft programs run at lightning speed. I spent about $75 on well-known books on the subject. I dragged through all of them and just didn't understand worth a darn.

Well, this week I got out all Roger Wagner's *Softalk* Assembly Lines articles and reviewed each one, one after another. I put them on the Apple, and everything did just what he said it would.

Suddenly, I'm an assembly language programmer! I got it down pat!

I really believe Roger Wagner has made a very significant contribution to our Apple computer industry. Every day brings something new in our field. Roger Wagner's clear, straightforward writing style makes it easy. He doesn't leave anything out that should be there. He gives you a total picture. I reread some of those books and those other guys don't tell the whole story. I appreciate the care and accuracy Wagner takes with information and his empathy with the reader. I'm looking forward to Roger Wagner's future efforts.
Larry Engel, Garden Grove, CA

Translator Needed
I am having a problem, and maybe some of your readers can help.

As the owner of an Apple II, I have compiled some of my very favorite programs and used good parts of each.

For example, I've taken Roger Wagner's *Apple-Doc* and used the feature locating and changing names of variables. I've used Sensible Software's *Super Disk III* to alphabetize my catalog, after using the Beagle Brother's *DOS BOSS* to customize my catalog by putting their name on it, as indicated in their instruction book.

Then, using Apple's *DOS Tool Kit*, I have been able to shorten the programs to load faster and, with Synergistic Software's *Higher Graphics* by Robert Clardy, I've done some beautiful things.

All this I have now combined onto one disk—all these remarkable programs—and now my problem:

This disk is eighteen inches wide because of the stuff on it. How can I get it into my disk drive?

Can any of your readers help?
Paul Raymer, Paul's Electric Computer, Las Vegas, NV

Undoubtedly, Softalk*'s readers will have better solutions, but just to get you started, why not try feeding the whole thing to Programma's* Applesoft Optimizer*? It does wonders in doing away with any extraneous material.*

On second thought, perhaps we should test it out with your letter. . . .

Congregating Around Apple
As a probable subscriber and owner of an Apple microcomputer, I would like to ask you for some assistance. I have been introduced to your magazine in several ways during the last two weeks and think you have the expertise to assist me.

At the present time, I am treasurer of a Lutheran church and school. The job is very time consuming! I have convinced the congregation that a microcomputer would not only help me immensely but also provide other service functions needed by the congregation. I have gone to several Apple dealers in the area who have not really been helpful trying to

Losing Control?

⟨ESC⟩ape

With Products From Sirius Software

Gorgon: The earth has entered a time warp . . . and the battle has just begun.

Orbitron: Action in orbit and attacks from all sides.

Gamma Goblins: Avoid the hyperdermics while transporting blood to Hemophilia.

Both Barrels: A two-game pack including "Duck Hunt" and "High Noon."

Star Cruiser: Save yourself from the swooping aliens.

Pulsar II: Pulsar and Wormwall in a unique combination.

Phantoms Five: Real-time 3-D fighter-bomber challenge.

E-Z Draw: An easy, powerful, and versatile Apple graphics editor, includes the Higher Text generator by Ron and Darrel Aldrich.

Cyber Strike: Challenging galactic excitement for the real arcade gamer.

Autobahn: Hair raising excitement at 200 kilometers per hour.

Epoch: Explosive fast action with new 3-D effects. Complete with high speed docking and a time warp that is sure to give you vertigo.

Pascal Graphics Editor: Easy animation and powerful graphics editing has finally arrived.

Sneakers: Unending variety and fun as you meet the Sneakers, Cyclops, Saucers, Fangs, H-Wings, Meteors, Scrambles, and Scrubs.

Space Eggs: Hatch some fun with the spiders, lips, wolves, and Fuzzballs.

Contact your local computer dealer for more information. Dealer inquiries invited.

Sirius Software, Inc.

2011 Arden Way #2 Sacramento, CA 95825 (916) 920-1939

match a program to my needs. They either have tried to sell me the *Apple Controller* package or several programs from BPI having no printed documentation to describe the program.

My needs are basically to keep up-to-date records of receipts and expenditures. Expenditures include a payroll (eight salaried employees and several paid by the hour) and payment of bills, assigning all to various established accounts. At the moment, all expenditures are paid from a single checkbook with a great deal of time and paperwork.

One Apple dealer did introduce me to a program recently reviewed in *Softalk* (April 1981). The program's name was *Bookeeper* and *Check Writer*. It appeared to have many of the features that I have been looking for. Can you give me any information on similar programs or possibly some designed specifically for church bookkeeping?
Paul Prell, Westchester, IL

Invitation to Information
I'm writing this letter to inform any of your interested readers about a bulletin board in Jacksonville, Florida. Entitled the SEB Bulletin Board, the system is maintained by SEB Computer Consultants. Availability is seven days a week from 6 p.m. to 8 a.m. The access number is (904) 743-7050.

I hope your readers will find this of interest. The Bulletin Board consists of:
Apple II Plus, 48K
Two disk drives
Mountain Clock
Hayes Micromodem II
Software
Sam Bateh, SEB Computer, Jacksonville, FL

When Your Dealer Can't Help
We received a complimentary issue [of *Softalk*] at our local Computerland store. We (eight-year-old daughter and I) read many Apple magazines but find yours to be at the top in this league.

I am interested in obtaining the mailing address of Orange Micro. I would like to order their friction feed kit for my MX-80 printer.

How about a classified section for small producers of computer hardware and software?
Bob Davidow, Overland Park, KS

Orange Micro is located at 3150 East La Palma, Suite 1, Anaheim, CA 92806.

Plans are in the works to implement a classified section with a difference (touch of mystery here) soon; it won't be confined to Apple product producers.

Thanks to you and your daughter for your support.

I am an Apple user who is looking for eighty-column output for my computer. I recently was looking through the back issues of your magazine when I came across a report on video boards that give the Apple eighty-column output. I decided to do some research on my own on the *Smarterm* from Advanced Logic Systems. Unfortunately, I believe no such company exists because I have not been able to find its address in any of the other magazines that I receive. Please give me their address if you know it! I am desperate!
James Dumesnil, Port Arthur, TX

Although Advanced Logic is the manufacturer of the Smarterm, *Apple is acting as its distributor. Any questions about the* Smarterm *can be directed to Apple Computer Inc., 10260 Bandley Drive, Cupertino, CA 95014.*

Since the publication of the article to which you refer, another eighty-column board has reached the American market: the Vision 80 *from Vista Computing (Irvine, CA). Although Vista holds all rights to manufacturing the* Vision 80 *in America, the original card is the* ZEV 80/24 *card from Zofarry, Sydney, Australia.* ZEVs *are available directly from Australia; we are still waiting for Vista's version. This card was reviewed in the June 1981* Softalk.

I enjoy your magazine very much and find Marketalk and Tradetalk of particular interest. In order to request literature on new products, however, it would be convenient if you would include the producers' complete addresses (and perhaps phone numbers) rather than just the city and state.
Peter Wagner, Lake Katrine, NY

Relatively new Apple owners and any Apple users whose interest is in using the Apple purely as a tool often benefit from solid dealer support and help with learning and problems. If, as soon as they no longer need this support, Apple owners begin shopping by mail to the exclusion of buying from the dealers, those dealers will not be able to afford the time or personnel to provide such service.

In addition, many producers simply are not equipped to handle direct sales.

Thus, Softalk *encourages its readers to buy through their local retailers. As an emphasis to our words, we do not print full addresses of producers.*

For those of you who live too far from any dealer or otherwise require mail-order service, there are advertisements in Softalk *for several retailers who will be glad to serve you—with service as well as products.*

Default Is Whose? I Dunno. He's on Third and. . . .
In the July 1981 Beginners' Corner, you stated that the programs *Copy* and *CopyA* default to copying from slot 6, drive 1, to slot 6, drive 2. In fact, the default source is the slot and drive from which *Copy* or *CopyA* was run, and the default destination is the other drive on the same controller as the source you

GORGON is here...

THE EARTH HAS ENTERED A TIME WARP . . . AND THE BATTLE HAS JUST BEGUN

Sirius Software, Inc. 2011 Arden Way, #225A, Sacramento, CA (916) 920-1939

have selected. I think the column should have explained *Copy*'s defaults this way.

Technically, that isn't quite true. The default source is the default drive as of the execution of *Copy*. If you load *Copy*, then change the default (by using the slot and drive options on catalog, for instance) then the default for *Copy* changes (and the new default must contain *Copy.OBJ0*). I don't think this added complexity would be appropriate in Beginners' Corner.
John W. Baxter, Coronado, CA 92118

Roger's Rooters, Part 11
I have been reading Assembly Lines from the start with great interest. It is nice to have the subject explained relatively clearly. The only real problem I have with it is waiting for the next issue.

The contest winning program by Steven Morris was very interesting. There seems to be a slight mistake in the source listing. The label SCREEN in line 51 should be in line 52. The problem does not occur in the list from memory. Your readers might enjoy making a slight addition to the part of the program dealing with screen switching. The length of time spent in either mode can be put under paddle control. Inserting the following lines:

```
LDX #$00
JSR PREAD
```

between lines 52 and 53 and

```
LDX #$01
JSR PREAD
```

between 53 and 54 will do the trick very nicely. The effect is very interesting. I would like to see something on using hires graphics from machine language before the screen is mapped out. Some help along those lines would be greatly appreciated.
John McKee, San Diego, CA

Against Obsolescence
In the August 1981 *Softalk*, you say, "DOS 3.2 . . . is rapidly becoming obsolete." Being a normal selfish human, it makes me mad to pay $500 for a piece of equipment and have someone make it obsolete in less than two years. Who did it? Who should I be mad at?

Furthermore, how mad should I be? Does it mean that I shouldn't order any software unless the ad specifies thirteen-sector? Or is there something I can buy to make my DOS seem to specify the number of sectors. Surely someone must be producing thirteen-sector software for those of us stuck with an obsolete 3.2 DOS.

I'd appreciate a word or two about this in one of your future most interesting columns. In an era when everyone charges twice what everything is worth, I'm fascinated, and thankful, that such a quality magazine as *Softalk* is free!!
Charles M. Larson, Whittier, CA

The obvious solution—which I suspect is not what you want—is indeed a product you can buy to make your DOS 3.2 handle sixteen-sector software: Apple's DOS 3.3.

It doesn't make much sense to be mad at Apple for discovering a new, more efficient disk storage method or for marketing it. It makes just as little sense to be mad at the numerous people whose enthusiastic acceptance made the product popular. And it makes no more sense to be angry at the software manufacturers for recognizing the widespread popularity of the product and its value and producing software for it. That leaves us fresh out of answers to who you should be mad at.

Most current software either will boot on both DOSes, comes with two versions, or is in Muffinable 3.2. All these will work on your DOS.

Emphasis on Magic Window
I have an Apple II computer and an Epson MX-80 pin feed printer. I have just purchased a *Magic Window* word processor and am having trouble inserting the ESC control characters in the text to control the emphasized printing. The other control characters seem to work just fine. Can you recommend a solution or direct me to someone who might know the solution?
John W. Butler, Apalachin, NY

QUALITY DISK SOFTWARE

BACKED BY ON-GOING APPLICATIONS SUPPORT

APPLE II Ⓐ TRS•80 Ⓣ

HOME FINANCE PAK I: Entire Series $49.95 Ⓐ Ⓣ

CHECK REGISTER AND BUDGET: This comprehensive **CHECKING ACCOUNT MANAGEMENT SYSTEM** not only keeps complete records, it also gives you the analysis and control tools you need to actively manage your account. The system provides routines for **BUDGETING INCOME AND EXPENSE, AUTOMATIC CHECK SEARCH,** and **BANK STATEMENT RECONCILING.** CRT or printer reports are produced for **ACTUAL EXPENSE** vs **BUDGET, CHECK SEARCH DISPLAY RECONCILIATION REPORT** and **CHECK REGISTER DISPLAY** by month. Check entry is prompted by user-defined menus of standard purposes and recipient codes, speeding data entry and reducing disk storage and retrieval time. Six fields of data are stored for each check: amount, check no., date, purpose, recipient and **TAX DEDUCTIBLE REMINDER. CHECK SEARCH** routines allow searching on any of these data fields. Up to 100 checks/mo. storage $39.95

SAVINGS: Account management system for up to 20 separate Savings accounts. Organizes, files and displays deposits, withdrawals and interest earned for each account. $14.95

CREDIT CARD: Get Control of your credit cards with this program. Organizes, stores and displays purchases, payments and service charges for up to 20 separate cards or bank loans. $14.95

UNIVERSAL COMPUTING MACHINE: $49.95 Ⓐ

A user programmable computing system structured around a 50 row x 50 column table. User defines row and column names and equations forming a unique computing machine. Table elements can be multiplied, divided, subtracted or added to any other element. Hundreds of unique computing machines can be defined, used, stored, and recalled, for later use. Excellent for sales forecasts, budgets, inventory lists, income statements, production planning, project cost estimates-in short for any planning, analysis or reporting problem that can by solved with a table.

COLOR CALENDAR: $29.95 Ⓐ

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

BUSINESS SOFTWARE: Entire Series $159.95 Ⓐ Ⓣ

MICROACCOUNTANT: The ideal accounting system for small businesses. Based on classic T-accounts and double-entry booking, this efficient program provides a journal for recording posting and reviewing up to 1,000 transactions per month to any one of 300 accounts. The program produces CRT and printer reports covering:

TRANSACTION JOURNAL	BALANCE SHEET
ACCOUNT LEDGERS	INCOME AND EXPENSE STATEMENT

Includes a short primer on Financial Accounting. (48K) $49.95

UNIVERSAL BUSINESS MACHINE: This program is designed to **SIMPLIFY** and **SAVE TIME** for the serious businessman who must periodically Analyze, Plan and Estimate. The program was created using our Universal Computing Machine and it is programmed to provide the following planning and forecasting tools.

CASH FLOW ANALYSIS	SALES FORECASTER
PROFORMA PROFIT & LOSS	SOURCE AND USE OF FUNDS
PROFORMA BALANCE SHEET	JOB COST ESTIMATOR
REAL ESTATE INVESTMENT	INVENTORY ANALYSIS

Price, including a copy of the Universal Computing Machine $89.95

BUSINESS CHECK REGISTER AND BUDGET: Our Check Register and Budget programs expanded to include up to 50 budgetable items and up to 400 checks per month. Includes bank statement reconciling and automatic check search (48K) $49.95

ELECTRONICS SERIES VOL I & II: Entire Series $259.95

LOGIC SIMULATOR: SAVE TIME AND MONEY. Simulate your digital logic circuits before you build them. **CMOS, TTL,** or whatever, if it's digital logic, this program can handle it. The program is an interactive, menu driven, full-fledged logic simulator capable of simulating the bit-time response of a logic network to user-specified input patterns. It will handle up to 1000 gates, including **NANDS, NORS, INVERTERS, FLIP-FLOPS, SHIFT REGISTERS, COUNTERS** and user-defined **MACROS.** up to 40 user-defined random, or binary input patterns. Accepts network descriptions from keyboard or from **LOGIC DESIGNER** for simulation ..... $159.95 Ⓐ Ⓣ

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

MANUAL AND DEMO DISK: Instruction Manual and demo disk illustrating capabilities of both program (s) ..... $29.95 Ⓐ Ⓣ

ELECTRONIC SERIES VOL III & IV: Entire Series $259.95

CIRCUIT SIMULATOR: Tired of trial & error circuit design? Simulate & debug your designs before you build them! With **CIRCUIT SIMULATOR** you build a model of your circuit using **RESISTORS, CAPACITORS, INDUCTORS, TRANSISTORS, DIODES, VOLTAGE** and **CURRENT SOURCES** and simulate the waveform response to inputs such as **PULSES, SINUSOIDS, SAWTOOTHS,** etc. all fully programmable. The output is displayed as an **OSCILLOSCOPE-STYLE PLOT** of the selected waveforms (Apple only) or as a printed table of voltage vs time. Handles up to 200 notes and up to 20 sources. Requires 48 **RAM** ..... $159.95 Ⓐ Ⓣ

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

MATHEMATICS SERIES: Entire Series $49.95

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

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

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

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

ACTION ADVENTURE GAMES: Entire Series $29.95 Ⓐ

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

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

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

FREE CATALOG-All programs are supplied on disk and run on Apple II w/Disk & Applesoft **ROM** Card & **TRS-80** Level II and require 32K **RAM** unless otherwise noted. Detailed instructions included. Orders shipped within 5 days. Card users include card number. Add $1.50 postage and handling with each order. California residents add 6½% sales tax. Foreign orders add $5.00 postage and handling.

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

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

VISA master charge

Riverbank Software Inc.

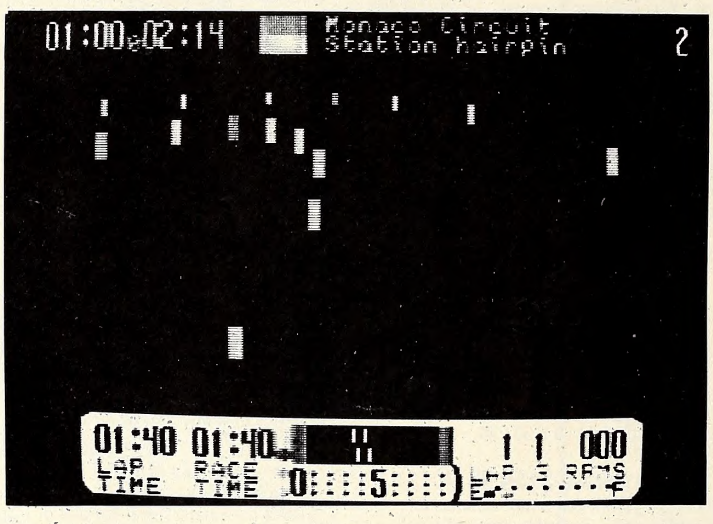
PROUDLY INTRODUCES

INTERNATIONAL GRAN PRIX

an arcade-like race car simulation
by RICHARD ORBAN
author of THREE MILE ISLAND*

REQUIRES APPLE II ★ ★ OR APPLE II PLUS ★ ★ 48K, 13 & 16 SECTOR DISK, PADDLE CONTROL

VISA/MASTERCHARGE
MONEY ORDERS/COD
DEALER/DISTRIBUTOR
INQUIRIES INVITED



$30.00 PER DISK
MD. RESIDENTS ADD 5%
U.S. CURRENCY ONLY

See your local dealer
or call Toll-free
24 hours for
ORDERS ONLY:
1-800-327-9191 ex. 244
(Outside Florida)
INQUIRIES AND FLORIDA
ORDERS: 301-479-1312
SMITH'S LANDING ROAD
POST OFFICE BOX 128
DENTON, MD. 21629

● SPECIAL FEATURES ●

Five GRAND PRIX-style road circuits, including: Oulton Park, Warwick Farm, Karlskoga, and Monaco ● Five speed manual or automatic transmission (with or without cruise control) ● Eight levels of difficulty.

● ADDITIONAL FEATURES ●

Speeds to 198 MPH ● controlled skids ● spinouts ● spectacular crashes ● hair pin turns ● narrow corners ● obstacles ● identified circuit features ● number of laps selection ● flashing last lap indicator ● 'Christmas tree' controlled start ● switch for silent operation ● blue post marks 300' intervals ● best lap/best race times posted ● fully instrumented control panel: lap timer ● race timer ● indicator lights ● edge detectors ● position indicator ● steering indicator ● moving speed tape ● lap counter ● gear and RPM indicators ● operating fuel gauge

RIVERBANK WILL REPLACE DAMAGED DISKS WITHIN 1 YEAR OF PURCHASE. RETURN DISK WITH PROOF OF PURCHASE PLUS FIVE DOLLARS POSTAGE AND HANDLING FOR IMMEDIATE REPLACEMENT
(TEN DOLLARS OVERSEAS)

*TRADEMARK MUSE CO., BALT. MD. ★ ★ APPLE COMPUTER, INC., CUPERTINO, CA.

THE BASIC Solution

By Wm. V. R. Smith

In previous Basic Solutions, we've created several subroutines and a couple of useful programs. Here's a list of the subroutines and the issues they were in:

Dollar Formatter, April 1981
Read Screen and *Autorun Catalog,* May 1981
Data Parser, June 1981
Video Tape Calculator, July 1981
Turn DOS On/Off, August 1981

If you've been a subscriber all along and missed any of these issues, please contact *Softalk* Circulation; if you're new to *Softalk* and would like to obtain these subroutines, most back issues are available for a small fee.

Many readers have responded to Basic Solutions with very interesting ideas for subroutines.

This month's subroutine was created by Clayton Curtis, M.D., of Tucson, Arizona. Curtis's routine solves a problem encountered by most programmers: how to input cursor control commands in a clear and uniform manner. Basic input statements don't accept cursor commands.

The subroutine is invoked by the usual GOSUB mechanism, watches for keyboard activity, and returns with a flag (RV) set to indicate which key was pressed. The set of keys used—return, left arrow, right arrow, and slash—were selected because of their physical orientation on the Apple keyboard and are used in our application to represent the directions up, left, right, and down, respectively. The escape key was chosen to allow a way for the user to specify an unusual condition. If characters other than these are entered, they are returned in a string variable (I$).

Flexibility is increased by the provision of a way to obtain a single character alone (similar to the Applesoft GET command) through the use of a character mode flag (CM), which, if set to 1, causes the subroutine to return a single character instead of waiting for a multicharacter string terminated by a return.

When the subroutine returns, RV will have a value of from 1 through 6, and, if RV = 2, I$ will contain either a single character, if CM = 1, or a string of one or more characters, if CM = 0.

The most convenient way of handling the various cases is through the Applesoft ON . . . GOSUB mechanism (which can be simulated in Integer Basic if desired). For example:

```
100 PRINT "FUNCTION: ";
110 CM=1: GOSUB 1000
120 ON RV GOTO (escape, string, left
    arrow, right arrow, slash)
```

will cause the message FUNCTION: to be printed, then the program will wait until one of the control characters, or any other single character, is pressed, at which point control will return to the main program.

Note that the subroutine doesn't tell the Apple what to do with the character once it has it. Presumably your program will contain routines for each of the alternative key presses. In line 120, you should substitute the starting line number of each of these routines for the appropriate key name as we've given it in parentheses. Drop the parentheses but keep the commas.

The following values are used:

Character Codes		Return Codes (RV)
27	escape	1
—	string	2
08	left arrow	3
21	right arraw	4
13	return	5
47	slash	6

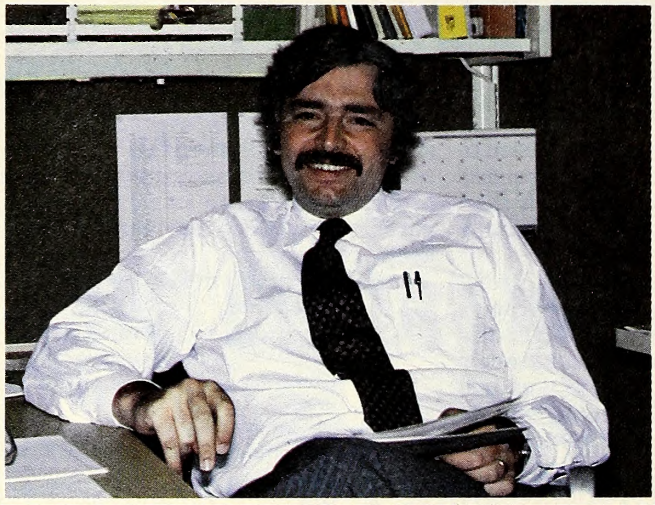
The subroutine itself:

```
1000 REM: KEYBOARD MONITOR
1010 C=PEEK (-16384) : IF C < 128 THEN 1010
1020 C=C-128
1030 RV=1: IF C= 27 THEN 1090
1040 RV=3: IF C=08 THEN 1090
1050 RV=4: IF C=21 THEN 1090
1060 RV=5: IF C=13 THEN 1090
1070 RV=6: IF RV=47 THEN 1090
1080 RV=2: IF NOT CM THEN INPUT "";I$: RETURN
1090 I$=CHR$(C) : POKE -16368,0: RETURN
```

You may add other character codes into this routine if you wish. Many programs require yes and no answers to questions. If you add the character codes for Y and N, the routine will return with the RV set to the proper values.

Your comments and solutions to problems are welcome always. If your solution is printed, you'll receive a ten-dollar credit toward any purchase at your local computer store. Mail your input to Softalk Basic Solution, 11021 Magnolia Boulevard, North Hollywood, CA 91601. ■

Exec Hayes



Exec Heatherington

Exec Sirkis

Exec Nacht

Exec Suite A

Hayes™

Hayes Microcomputer Products Inc.

BY ALLAN TOMMERVIK

Almost all companies have, in their embryonic stage, a founder or founders who combine the roles of philosopher and pragmatist successfully.

The philosopher sees the macropicture, thinking his way through various cerebral mazes to develop the raison d'etre for the company. He views the company's overall position in its marketplace, its producing methods, its products, and its ambience for employees from a relatively lofty plane.

The pragmatist reduces all these cerebral meanderings to practicality. Within the bounds of good business practice and the company's resources, he melds the philosophical to the real, in the process creating a hybrid that represents the best possible synthesis.

Seldom are both traits found in the same person. More often a company will have two or more founders with one providing the philosophical base while others build the practical foundation.

Triple Threat. The company that has two or more founders who can contribute to each function is doubly blessed. So the success of Hayes Microcomputer Products, manufacturer of the Micromodem II, comes as no surprise. It has three founders who actively contribute to both the philosophical and pragmatic bases of the company.

Make no mistake. D. C. Hayes himself, better known around the office as Dennis, was, is, and probably will continue to be the spiritual mentor of the company.

But also practicing as both gurus and businessmen, to varying degrees, are Dale Heatherington and Glenn Sirkis, the other members of the founding Hayes triumvirate.

Actually, to be precise, Hayes and Heatherington founded a firm originally known as D. C. Hayes Associates. Sirkis talked his way into the firm shortly after its founding.

If Hayes Microcomputer should ever achieve its stated goal of reaching the Fortune 1000 list, how these three execs divided up the company turf may become a textbook example of business management practices.

Hayes has applied himself to the manufacturing process in the present and functions as the seer for the group, trying to portend what the future may hold.

Heatherington has settled into the role of technical guru, concerning himself with what can be done with today's technology and how to provide technological solutions to problems that come to his attention.

Sirkis is the purveyor of product, analyzing the marketplace and the company's role in it. His is not a passive, or reactive, role of merchandiser, but rather the role of an active participant in shaping the market.

This trio represents a surprising agglomeration of talent in what has been essentially a one-product company. The company is now beginning to broaden its product line, but how it arrived at this point and the founders' thoughts are as interesting as where the company is going.

The Gleam of Interest Piqued. Dennis C. Hayes, president of the firm, got his start in data communications transmission as a student at Georgia Tech when he landed a co-op student job with AT&T Long Lines in the radio engineering group.

Hayes was an electrical engineering student at the time, but he switched to physics after his AT&T stint. In the wry style that is his fashion, the humor given away by the twinkle in his eye and the crinkle around his eye, Hayes remarks that he saw too many electrical engineers who were married to the technology they had learned in school in the fifties, and he didn't want to be married to the technology of the seventies. So it was the more fundamental physics for which he opted.

From school, Hayes joined Financial Data Sciences, a company making terminals for savings and loan companies. There he was involved in integrating the Intel 4004 chip into the terminals. That early microprocessor enabled FDS to cut the number of parts in the terminals from one hundred seventy-seven to twenty-five. During his stay at FDS, the first data sheets on the 8080 microprocessor were released.

Hayes subsequently joined National Data Corporation, a mainframe service bureau for large data processing users. It was there that he met Heatherington, now Hayes Microcomputer's senior electronic designer, when both were designing and building communications interfaces of variable complexity. These were specialty one-of-a-kind items that would tie two networks together for a dedicated purpose.

A Wedding of Ways. Heatherington arrived at NDC from Southern Tech, where he studied electrical engineering. He stayed in the design group while Hayes moved into operations as a network engineer and later became part of the systems programming group.

But both were becoming interested in a wider spectrum of data processing than just building communications boxes.

Hayes was still intrigued by the specs he had seen on the 8080 chip. And Heatherington took a look at the early entries in the microcomputer marketplace and bought an Imsai.

Hayes wanted to get involved in the revolution he saw coming. But he recognized that he had insufficient resources to develop and support an entire computer system.

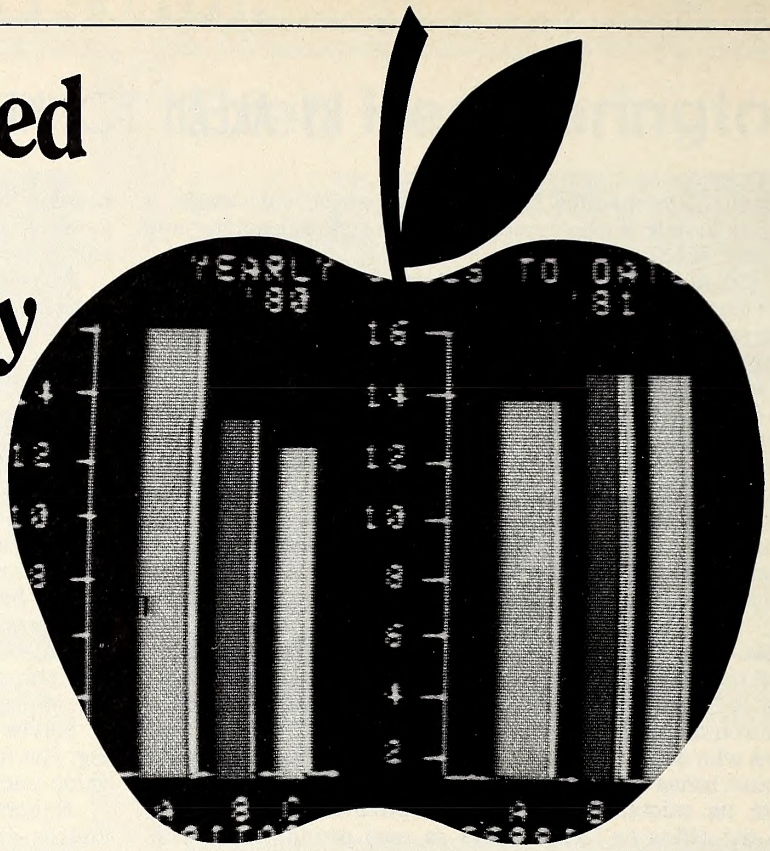
Nevertheless, the entrepreneurial drive remained strong, and he and Heatherington agreed to fall back on what they were already expert in—data transmission—as a means for getting involved in the personal computer revolution.

What they designed was the 80-103, a communications adaptor for S-100 bus computers. The first models were delivered in April 1977.

About that time, Sirkis was looking around for a way to get

How to succeed in business without really trying.

For peanuts.



STC is proud to formally announce four field-proven business packages for APPLE users. And an irresistible offer.

All of these programs are straightforward, easy to use, virtually foolproof, and very attractively priced. The prices quoted include both the programs and easy-to-follow instruction manual.

Accounts Payable

Uses ISAM structured routines and all posting is done in seconds as opposed to hours with other programs. Various reports are generated, including cash requirement journal, check register, open invoice report, Y-T-D payment journal, and aging function to tell how many days left for a discount or an invoice. You also get a complete audit trail on all transactions, a current and Y-T-D analysis of all your accounts and vendors, and a check printed on available forms. **Only $200.**

Payroll

Maintains an entire payroll for up to 125 employees. It provides numerous payroll utilities for quick and easy payroll and check generation. It has built-in Federal and State tax tables and provides some remarkable reporting capabilities. **Only $240.**

Professional Time and Billing

Maintains time and billing data for up to 300 clients. 120 rates (employees) can be on file at any time. Rates may appear as time/expenses/fixed fees. Prints statements on available forms. Generates a variety of transactions and activity reports, and much more. **Only $325.**

Apartment Management

Maintain financial and managerial data for up to 6 separate apartment complexes. Each complex can contain a maximum of 120 units and 8 different types of units. Generates an operating statement which contains YTD rental income, security deposits, pet deposits, pool deposits, misc. deposits, damage amounts, allowances, expenses and a complete breakdown of apt. units by type. Again, some great reports. **Only $325.**

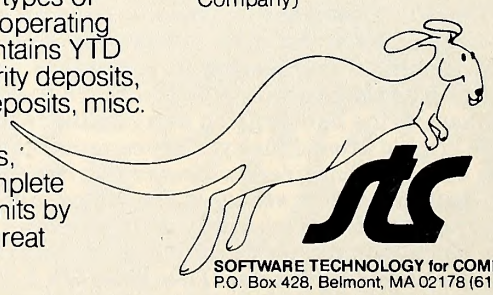
Early Bird Offer

If you purchase any one of these programs from your STC dealer within the next 30 days, we'll let you have our new and quite fantastic Coloring Board package for only $30. This program, normally costing $60, compares with some hardware capabilities costing $800 and more. It allows you to quickly create, store and display all types of graphics and text labeling. For example, a bar-chart like the one shown above for display at sales meetings. In short, it has unlimited useful applications for business.

Act now. Contact your local STC dealer for details on this extremely unusual offer. And be sure to check out the rest of our great line of software packages as well.

Dealer inquiries invited.

(Apple is a trademark of Apple Computer Company)



SOFTWARE TECHNOLOGY for COMPUTERS
P.O. Box 428, Belmont, MA 02178 (617) 923-4334

involved in the microcomputer industry. He thought he recognized all the same elements in the personal computer market that had once fueled the boom in minicomputers.

Seer of Revolution. Sirkis had also attended Georgia Tech, working his way through school at television stations. His undergraduate days were concurrent with Hayes's, and it's possible the two shared some classes; but they never met on campus. Sirkis majored in industrial management, which gave him the ability to perceive certain portents in the microcomputer industry that paralleled the minicomputer boom.

The Data General Nova, first of the minicomputers, had been an 8K system with a teletype paper tape input when it was announced in 1968. The Altair was a 4K computer with paper tape or cassette access when it was announced in 1975.

Sirkis had witnessed the development of additional memory, disk drives, printers, and systems software in the subsequent two years and felt the personal computer market was ready to break out as the minicomputer market had done.

A friend told him about two guys working out of their homes who had a communications device for personal computers. They were so busy that it took Sirkis three months to wangle a meeting with them.

The two guys were Hayes and Heatherington, and Sirkis, although not knowing a thing about modems, did know enough to recognize the care they were taking and quality they were building into each device.

In addition, Sirkis saw the unique niche that D. C. Hayes Associates was carving out for itself. To the extent that there was any tradition at all in the computer industry, peripheral and support companies normally provided such products as printers while communications companies provided data transmission services.

A computer company specializing in communications products seemed to have great promise.

The mix whetted Sirkis's already strong entrepreneurial appetite and he talked himself into a partnership and the job he still holds—vice-president of finance and marketing. As Hayes remembers it, "The first task we assigned him was to figure out how to pay himself . . . his first official action was to cut my pay."

Flowing with the Cash. Sirkis provides another perspective to the story. "We had to cut all the expenses back in order to survive. At first, I didn't get paid at all."

The early manufacturing history of the company is not atypical for peripheral makers. The company would receive mail orders for the product and take that money to the suppliers to get their raw materials.

The material order would be placed on Tuesday and picked up on Friday. Hayes used a bedroom and the dining room table at home as his assembly line.

On Saturday, he would assemble the products and deliver them to Heatherington the next day. Heatherington would test each piece on his Imsai—returning the good ones to Hayes for shipping on Monday and keeping the defective ones to repair.

The 80-103 data communications adaptor was an excellent product, but, by the time Sirkus came onboard, it became clear that superior product alone was not the answer. Such manufacturers of fine personal computers as Imsai, Altair, and Sol were either in financial trouble or already out of business.

The trio made their immediate goal one of survival within the context of delivering quality product. As with most companies in the microcomputer industry, D. C. Hayes started out undercapitalized. For money, they substituted what Sirkus calls "sweat equity."

The company's next product was the one that secured its future and made it a household name in the Apple community—the Micromodem II.

The 80-103 was doing about as well as could be expected, but the sales surge in computers not having an S-100 bus system forced the company to reevaluate its market.

Mixing Apples and Micromodems. After looking at the three most viable possibilities at that time—TRS-80, Apple,

and Pet—they settled on aiming at the Apple market. Apple was a clear choice because of the architecture of the expansion ports and the reliable power supply.

Likewise, Apple had the most suitable dealer and distributor network through which to sell.

Making that decision and implementing it were two altogether different things. At that time, the company had fewer than ten people and practically no cash with which to make the prototype to submit to the Federal Communications Commission for approval.

The cash problem was compounded by Dale Heatherington's design, which called for both a modem and a microcoupler. Heatherington designed the product in two parts with the future in mind.

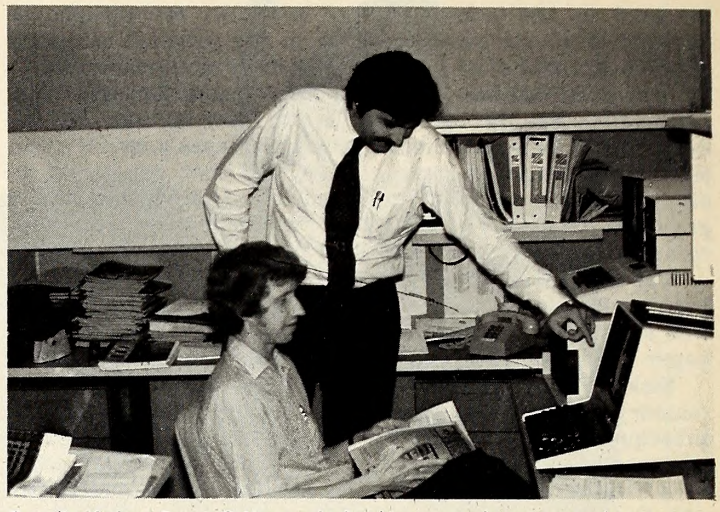
Because it's the microcoupler that physically connects to the telephone system, FCC approval of that device would allow the company to hook other products up to common carrier lines without additional approval.

Heatherington also reasoned that both the telephone lines and the user's personal computer could stand to be protected from each other. Inserting the microcoupler as an interface served this purpose.

A Scary Rumor. While the success of Heatherington's design is now history, the first days of the product were not without some suspense. Soon after the company made the firm decision to proceed with the product, the rumor reached them that Apple would soon be manufacturing and marketing their own modem device.

The Hayes folk value their Atlanta location for various reasons, not the least of which is their distance from all the hubbub of the silicon gulch. By the time rumors filter through the system to Atlanta, they are usually easily confirmed or denied.

But the rumor that they might soon to be head-and-head with Apple caused a certain amount of consternation. Whatever event caused that particular rumor to circulate, history



Softalk photo

Founding fathers: Dennis C. Hayes, who lent his name to the company, observes some of the technical wizardry of Dale Heatherington, who's designed all the Hayes products. Hayes's bedroom was the company's first manufacturing facility and Heatherington's personal Imsai served as the company's first quality control machine.

records that Apple did not choose to bring out a modem at that time.

Distance from Apple's headquarters in Cupertino, California, and the timing of the decision to manufacture an Apple-specific product also raised some difficulties. At the time the company dedicated itself to the Micromodem II, no disk operating system yet existed for the Apple.

As DOS was implemented and other changes were taking place, Heatherington was kept scrambling to maintain the pace.

In September 1978, the company sent out photos and fact

sheets on the Micromodem to selected dealers. They received their first thirty orders and a call from Apple founder Steve Jobs that opened the lines of communication between the two companies.

The thirty orders were harbingers of times to come—nearly from the moment of those first orders, the company has had a backlog of orders. While the backlog maintained constant stress on their manufacturing facilities, it also made possible the unusual corporate structure still in existence at Hayes: the company has no sales force.

It has been essentially the Micromodem II that has fueled the growth of the company. A year ago, the company had fewer than twenty-five employees; today there are more than eighty employees. In addition, the company has expanded its product line with the addition of other communications transmission equipment, and it has announced a new series of non-communications products, first item of which will be available this fall.

In the communications transmission area, Hayes has replaced the 80-103 with the Micromodem 100 for computers with the S-100 bus structure, and this spring it announced the Smartmodem, a modem using the RS-232 interface.

Stacking the Deck. The Smartmodem was also the first of a line of products the company intends to introduce under the umbrella marketing name of the Hayes Stack. All the products will be packaged the same way to allow them to be stacked one upon the other next to your personal computer. Already announced as the second product for the stack is the Chronograph, a calendar/clock module.

Growth and expanding product lines have caused Hayes's exec trio to question just about everything about their business, from its technology to its management.

Neither Hayes nor Sirkis sees any inherent dangers in rapid growth, although both emphasize that they prefer a slower, more controlled growth. In Hayes's words, "Growth as fast as Apple grew is a little scary."

In terms of technology, Sirkis thinks in a slightly more macro sense than Hayes and Heatherington. He points out that the minis of today are more powerful than the mainframes of fifteen years ago and that the micros of today are more powerful than the mainframes of thirty years ago and the minis of fifteen years ago. He stops there and leaves the conclusion of that analysis hanging, but the clear implication is that technology has not been frozen and further advances are to come.

On another level, Sirkis points out that the actual cash outlay for data processing, whether in business or at home, never goes down. He observes that computers are getting "bigger and smaller, cheaper and more expensive."

Reality gives lie to the thought that his words form a paradox: computers today are bigger in that individual units are carrying more memory than ever before, while at the same time they are packaged in smaller and smaller cases.

Likewise, the cost for processing an individual item of data continues to decrease as computers become more efficient and more powerful, but the overall cash outlay for these more efficient machines is higher than for the less efficient ones that preceded them.

The Large and the Small of It. While Sirkis is voicing these macroindustry thoughts, Heatherington contemplates the technological possibilities. He's not at all sure what technological breakthroughs can occur to help the personal computer user.

Among other things, he believes that the 300 baud transmission rate, approximately thirty characters per second, may be the maximum speed for reliable transmission by low-cost communication transfer devices. And, even at higher speeds with more sophisticated equipment that contains the most advanced protocols for error checking, data transmission reliability is only as good as the phone lines over which the data is transmitted.

To a certain degree then, progress for the hobbyist and home computer user will depend not on the efforts of such folks as those at Hayes, but on advances in the laboratories of West-

ern Electric and other telephone company research subsidiaries.

Both Heatherington and Hayes see a potential for improved transmission through better data compression techniques. And Heatherington points out that some local networks, such as Ethernet, are now transmitting ten million bits of information per second.

Unlike other computer peripheral manufacturers, Hayes is unable to export their products because communications regulations and standards vary with each country. That selfsame difference may have insulated the company from such outside competitors as the Japanese in the past, but Hayes feels his company is ready for the challenge, should it ever come.

Ideas from Abroad Are Actually on Return Trip. Hayes takes Japanese manufacturing technology so seriously that he's been studying their style and methods "since before it was fashionable." Among his thoughts on the subject are that the Japanese have no secrets that can't be duplicated by astute observers.

He feels that many of the best of the Japanese skills are distilled American methods, and if American manufacturers will just be willing to adopt those ideas that originated here and were exported, they will become more competitive.

Hayes Microcomputer has emphasized automation as a means of keeping up with the Joneses. Automation allows each employee to maximize productivity, which has more than just profitability implications at Hayes. But, in terms of competitiveness, the high productivity of the company's manufacturing staff makes Hayes believe that they'll remain competitive with any manufacturer—foreign or domestic.

Hayes is so serious about tracking Japanese manufacturing that he's studying the language at two learning sessions a week. He shares with many other observers great admiration for the quality control circle method.

Quality control is a subject of importance to Hayes because the company has the most liberal warranty program in the Apple marketplace—two years. Sirkis says that, to this date, the company hasn't even charged for repairs on product that was past warranty, although he concedes that policy may change.

The company's product is so reliable, thanks to their quality control procedures, that even though there are more than twenty-five thousand Micromodems installed, it takes only one individual to handle repairs.

The company takes pride in turning those repairs around in twenty-four hours, usually by replacing the offending piece. Hayes notes that few users become so emotionally attached to their circuit boards that they object to the substitution.

Even with the care and expense spent on quality control, automation has enabled the company finally to erase their backlog of orders.

Making Work a Pleasure. While that's a primary benefit of automation, Hayes sees a byproduct of automation as equally important. That's the fact that increased productivity enables the company to share the benefits with their employees—improving their lifestyle as a direct result.

So important is this goal of making the company a pleasant place to work that the first top exec into the company in the past three years, Art Nacht, who joined the company earlier this year, has been charged first with organizing the employee benefit program, even though employment and employee benefits represent only a partial list of his duties.

Nacht is also responsible for other administrative functions and will develop a software support capability for Hayes products.

While Nacht is a latecomer to Hayes Microcomputer, he's no stranger. He followed Sirkis as manager of the Georgia Tech radio station and carries both the bachelor's and the master's degrees in industrial management.

He had been actively involved with the MIN company in the design and development of a software package aimed at apartment owners, *The Landlord*, prior to his association with Hayes Microcomputer.

While Nacht's versatility enhances the company's executive abilities, it is a telling comment on the sincerity of Hayes's desire to create a pleasurable working environment that Nacht's time has been mostly spent in that area to date.

Hayes takes pleasure in seeing new employees take hold and, over the course of their first year, integrate themselves into the company. A major portion of Nacht's effort has been in facilitating that process for the company's employees.

As with most other companies in the microcomputer industry, Hayes Microcomputer is actively bidding for creative talent. Nacht is presently looking for both hardware design engineers and software programmers to complement current company activities.

The addition of Nacht as software honcho does not signify a change in attitude by the company. In the past, they have left applications software support to software publishers, and neither Heatherington nor Hayes sees any significant change in that policy.

But the company has published a Pascal software support disk and can probably be expected to fill some other voids if outside software sources fail to do so.

A Place in the Sun. There are those who feel that modems used in conjunction with personal computers have the potential to change society radically, permitting individuals to work, shop, bank, and vote from their homes.

The implications of such changes to society are far ranging, impacting on where we live, how we govern ourselves, who we work for, when we work, how we work, and what we work at, what we buy, and how we buy it.

Even such seemingly comprehensive tomes as Alvin Toffler's *Third Wave* underestimate the upheavals that could be caused if the capabilities of modems and personal computers are pursued to their logical conclusion.

Whether the change will be revolutionary or evolutionary, Hayes Microcomputer will be serving the cause with a reliable line of the products that make it all possible. ■

ASCII EXPRESS II

by BILL BLUE

Described in INFOWORLD as "The finest program for Apple data communications...," ASCII EXPRESS II allows your Apple to communicate with virtually any computer with dial-up access.

Written in Applesoft and Machine language, Ascii Express II includes everything you'd expect in a complete communications package. It features a variety of powerful features including full support of upper/lower case, autodial and answer capabilities (when used with the Hayes Micromodem), and file oriented upload/download facilities.

A built-in line editor gives full editing functions, and programmable keyboard MACROS reduce complicated log-in procedures to a few simple keystrokes.

Downloaded files may be printed while being received, saved to disk, or printed later when offline. The copy mode allows everything shown on the screen to be saved in the large (20K) buffer.

Ascii Express II works with the Hayes Micromodem II, Apple communications card, the CCS Asynchronous Serial card, SSM-AIO Board, Lynx Telephone Linkage System, and many other communications devices.

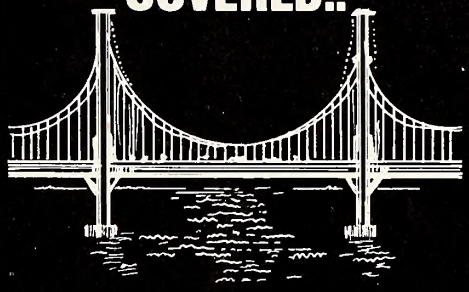
Uses include:

- Send/receive letters/files from networks like the SOURCE, MICRONET, or other bulletin board type systems.
- Transferring program files between Apples, an Apple and a TRS-80, PET, etc.
- Use the Apple as a terminal to a mainframe at a remote location with the added advantage of being able to process data at the Apple before or after transfer.
- Minimize on-line costs by quickly transferring files and other data.

System requirements include a 48K Apple with Applesoft in ROM or the Language Card, a disk drive, and one of the above communications devices. A lower case display board is recommended, but not required.

Price: $64.95

COMMUNICATIONS GAP??? WE'VE GOT YOU COVERED!!



S.D.S. offers a complete selection of communications software to meet almost every user's need. We think you'll find the three programs described here the best available anywhere — and we back that claim with our unconditional guarantee of satisfaction or your money back.

To find out more about these programs, send for more information, or see your local Apple dealer.

ONLINE

by BILL BLUE

ONLINE is a completely secure dial-up system that allows you to call your Apple computer while you're away from home. It also provides for up to 50 optional user accounts.

ONLINE includes a versatile mail system and built-in line editor with provisions for uploading and downloading programs and files.

Its many applications include use by businesses for 24 hour answering of field representatives inquiries, taking orders or advertising for your company. It can also be used by clubs or groups for posting announcements, or transferring files.

Requires 48K Apple with Hayes Micromodem and DOS 3.3.

Price: $89.95

(Calif. res. add 6% ON ALL PRICES)

*Prices effective until 10/1/81.

SDS

southwestern data systems

P.O. BOX 582-S • SANTEE, CA 92071

(714) 562-3670

Z-TERM

by BILL BLUE

The Rolls-Royce of communications software. You may find cheaper programs but you'll never find one better. Not only does it provide everything Ascii Express II does, but then some. Designed for the CP/M environment using the Z-80 Softcard, Z-TERM permits a number of features not available elsewhere.

- Receives up to 41K of data at a time. Can send files of any size.
- Auto save mode sends XOFF character when buffer is full, and resumes (with operator prompting) after save.
- Terminal emulation allows you to define the type of terminal your Apple (equipped with 80 column board) will display as.
- Entirely in machine language for maximum speed and power.
- Complete programmable keyboard macros.
- Can produce entire ASCII character set including break.

Z-TERM fully supports the Hayes Micromodem, Apple Communications card, SSM-AIO board, CCS Asynchronous Serial Card, Lynx Communications system and others! Fully supports the local Apple 40 column screen, external terminals, and all 80 column boards interchangeably and with NO configuration necessary!

If you have a Z-80 card, you owe it to yourself to check this one out before you buy any communications software. If you don't have the Z-80 Softcard, you may want to get one just to run this package!

*Note: CP/M and Apple DOS files are not directly compatable.

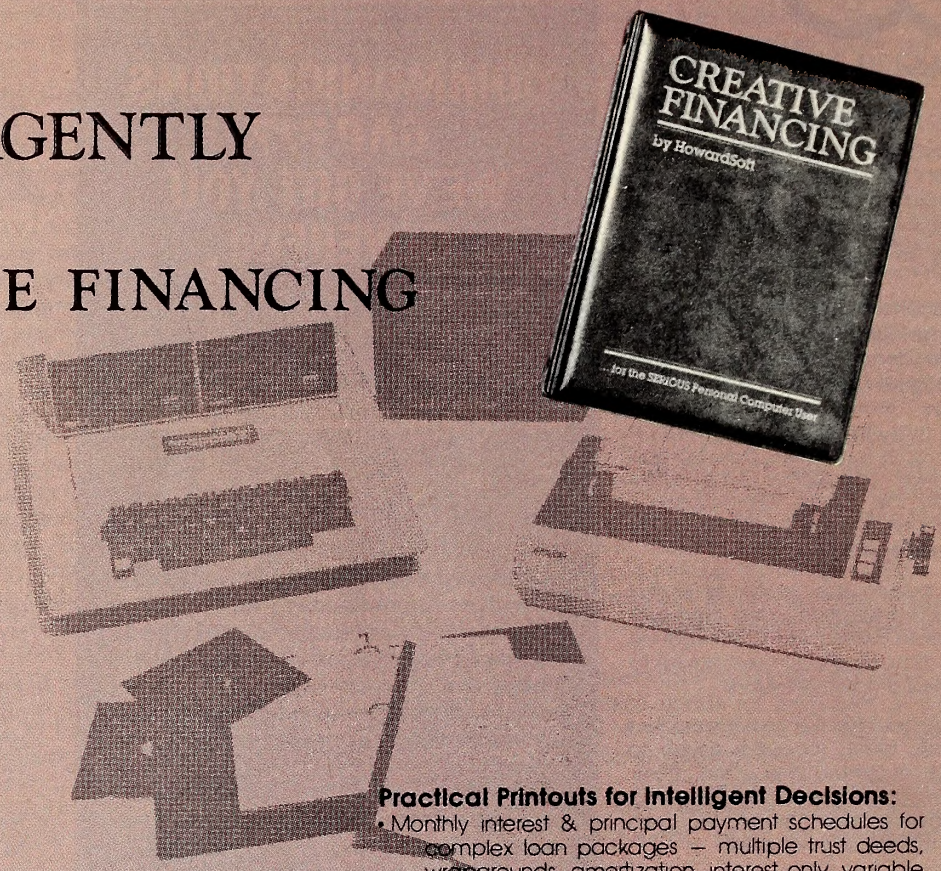
Price: $99.95

INVEST INTELLIGENTLY

with

CREATIVE FINANCING

by HowardSoft™



Practical Printouts for Intelligent Decisions:

- Monthly interest & principal payment schedules for complex loan packages – multiple trust deeds, wraparounds, amortization, interest only, variable interest rates, delayed payments, balloons, rebates, mid-year starts, add-on interest, simple interest.
- Annual depreciation schedules for complex component depreciation – straight line, sum of year's digit, declining balance, automatic conversion to straight line, tax credits, recapture of excess depreciation.
- Tabular projections of year-end on-sale profits – internal rate of return, return on investment, return on prior year's equity, net present value, future value
- Tabular projections of cash flows – debt service, tax credits, leases, tax payments, expenses

Analyses for Today's Complex Transactions:

- Picks up where our Real Estate Analyzer left off.
- Equivalent compound annual interest rate of loan package
- Yield to maturity for discounted bonds
- Present value of complex annuity schedules
- Value of ownership vs. sale-leaseback.
- Profit comparisons for dissimilar investments
- Risk evaluation through "what if" comparisons

Friendly Software for Efficient Operation:

- Extensive documentation, complete with tutorials and sample analyses for the financial novice.
- Error-tolerant and trouble-free for use by the computer novice.
- Flexible, professional, paginated report generation for impressive reports on all printers.
- For all Apple Computers with 48K, Applesoft, and one 16-sector disk drive.

Get an Edge with CREATIVE FINANCING:

This sophisticated but friendly software clears the fog created by today's creative loan packages, complex tax laws, runaway inflation, and cash-draining investments. Whether you invest in bonds, stocks, income property, or trust deeds, or you just want to determine the true cost of your home mortgage or consumer loan, this package will aid you in intelligent decision-making. Realistically see the direct impact of inflation, time value of money, taxes, and cash flow timing, and play "what if" with the confidence that all important factors are properly handled. It is now possible in just a few minutes to objectively compare dissimilar loans and investments with the realism formerly achieved only by the few financial wizards who could complete analyses of this type in a few hours. This package is INDISPENSIBLE FOR ANYONE WHO INVESTS WITH BORROWED MONEY.

NOW AT YOUR DEALER FOR $150.

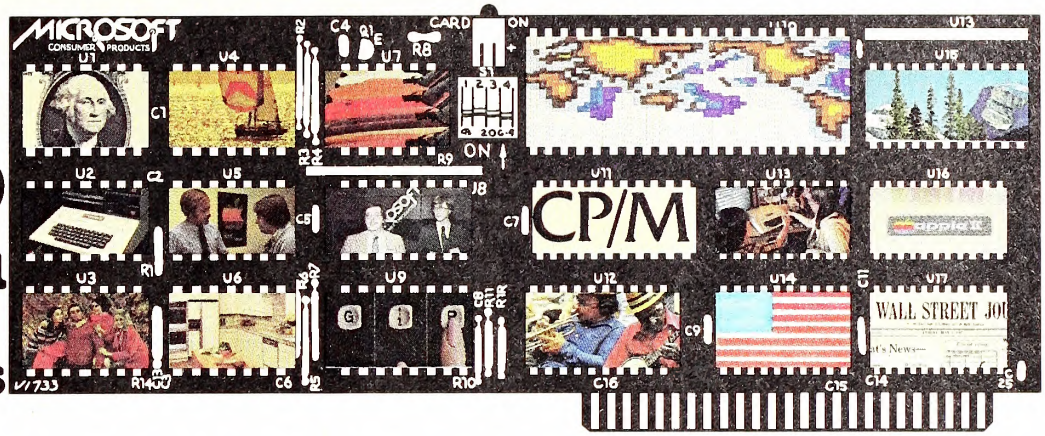
Also ask about our Tax Preparer and Real Estate Analyzer software.

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

6713 Vista del Mar | La Jolla, CA 92037 | (714) 454-5079

SOFTCARD Symposium

by Greg Tibbetts



In last month's column we talked about the SoftCard and CP/M in a general way to give those of you who are unfamiliar with the product some idea of what it is. This month we'll concentrate on the Basics provided with SoftCard, since these are the languages most of you will be working with for your applications programs. We'll pay special attention to the differences between the SoftCard Basics, MBasic and GBasic, and the Apple's normal Basics, Integer and Applesoft.

The Basic shipped with SoftCard comes in two forms: MBasic and GBasic. Both of these are versions of Microsoft Basic-80, release five, with MBasic being very close to standard Basic-80 and GBasic containing the provisions of MBasic plus most of the hi-res graphics capabilities you're used to with Applesoft.

Although it may seem redundant to supply two versions, it was done with good reason. Essentially, both are provided so users can choose the tradeoff (between hi-res graphics and extra memory) that best suits their programming goals.

The need for a tradeoff is due to the methods used by Apple to implement hi-res graphics. To make a hi-res display, the Apple stores the data for the display in RAM in an area called a *picture buffer*. Although there are two such areas available in Applesoft, GBasic uses only the primary picture buffer, located between 6502 addresses 2000H and 3FFFH (Z-80 addresses 1000H and 2FFFH).

Under Apple DOS and Applesoft, which are both located high in memory, this area lies within the user program space. Although it may sometimes be inconvenient for the programmer to arrange his programs and select the proper picture buffer to avoid overlaying them, it generally isn't too difficult.

Under CP/M, however, Basic, like most other programs, resides in low RAM memory beginning at Z-80 address 0100H (6502 address 1100H). Since the core of Basic-80 is slightly more than 23K in length, this combination means it would extend over the area needed for the picture buffer. To resolve this problem, Microsoft split Basic-80 into two parts around the buffer, and, although the actual code added to Basic-80 to produce hi-res graphics represented only about a 4 percent increase in its size, the need to protect the picture buffer made this an 8.8K total size increase. Therefore, using MBasic, the user gains nearly 9K of program space.

After hearing this explanation, many users want to know why Basic-80 is so much larger than the Apple Basics in the first place. The reasons lie in the additional features and in the different method by which it supplies the features they share. This, in turn, is a function of the environment for which they were developed. Both Applesoft and Integer Basic were originally developed with the assumption that they were to go into specific and limited areas of memory reserved for firmware in ROM. They were also designed for nondisk systems and, consequently, have no disk I/O routines as part of their structure. The routines necessary for such I/O activity are contained within Apple DOS, which intercepts such commands before Basic sees them.

Basic-80, on the other hand, was developed from the beginning as a RAM-resident language, so the only limitations on its size, and, therefore on its features, were the total RAM available in the system and the needs of the user for program space. Consequently, Basic-80 was designed to emulate the Basics found on larger machines, to supply as many of those features as possible, and to conform to the ANSI standard for the Basic language. Also, since it is RAM-based, it has changed to reflect its environment with the addition of disk I/O routines as the environment itself changed. Let's take a look now at some specific differences between the Apple Basics and Basic-80.

First, when either of the two forms of Basic-80 are executed, by typing MBasic or GBasic, several options are offered that affect Basic-80's operation. These are discussed in the beginning of the second chapter of the *SoftCard Manual*, Volume II. In comparison with Apple's Basics, the /F: option is similar to issuing the command MAXFILES from within Apple DOS. Essentially, its only purpose is to tell Basic how many sequential or random disk files your program expects to have open at any one time so that the necessary number of buffers will be maintained to permit blocks of data to be loaded from disk or sent to disk. Each allocated buffer consumes a significant amount of memory and, consequently, the number used should be held to a minimum. Default if no /F: is specified is three open files.

The /M: option is identical to issuing the Applesoft HIMEM: command in that it sets the top of available space for Basic to use (that is, the string storage area).

The last option for initialization, /S:, is used to specify the maximum size of individual records used in random access disk files. This has no analogous command in Apple's Basics. It does not mean that all records used in the program must be this size, only that this is the maximum they can be. What this command actually does is set the size of the buffer we discussed under the /F: option. A base amount of 166 bytes is allocated for each buffer plus the number of bytes specified with the /S: option. This is useful if you plan to have large numbers of files open, with relatively short record lengths. Setting /S: to the smaller record length will save memory for program use. The default here is 128 bytes, the size of a logical disk record under CP/M.

Language differences can be classified as additional features or features used differently. We'll look at the additional features first.

The new direct commands (commands typed at the keyboard for immediate execution) are: AUTO; EDIT; LLIST; MERGE; RENUM; RESET; and SYSTEM. New indirect commands (commands normally used only in program statements) are BEEP; CHAIN; COMMON; DEFINT/SNG/DBL/STR; ELSE; ERASE; ERR/ERL; ERROR; FIELD; LINE INPUT/#; L/PRINT/USING; LSET/RSET; OPEN; OPTION BASE; RANDOMIZE; SWAP; WHILE/WEND; and WIDTH. While indirect commands are normally used only in program statements, Basic-80 will allow them to be used as direct commands and will even allow multistatement lines to be executed in this manner. For example,

```
For I=1 To 2000:Print "This is a direct command sequence";:Next I:Lprint "All done":For I=1 To 600:Next I:Home
```

is a perfectly valid direct command to Basic-80. Many of the

normally direct commands can also be used in indirect mode (within programs); however, their use is not so straightforward and can occasionally have bizarre and undesirable side effects.

Of the direct commands, RESET and SYSTEM apply only to Basics that operate within the CP/M environment. SYSTEM is the command that essentially halts Basic and returns you to the CCP, which is the interactive mode of CP/M itself. RESET is the command used to tell CP/M that you have switched diskettes in one of the drives and will write to it. RESET is required since CP/M keeps certain diskette directory information in memory and needs to know when diskettes have been changed.

Of the rest, AUTO, EDIT, MERGE, and RENUM are used primarily for program entry and editing. AUTO works the same way as Integer Basic's AUTO command and is well documented in the manual. RENUM is likewise well documented and simple to use. MERGE and EDIT, however, have caused some users difficulty; consequently, we'll take a look at them.

MERGE is a command that some users have confused with the optional MERGE feature used with the indirect command CHAIN. As a direct command, MERGE is really very simple. Its sole purpose is to insert a program on disk into the program in memory, and thereby combine the two into a single program in memory. Basic lines will be sorted so that they are in correct ascending numerical order, and if any lines in the program on disk have the same line number as an existing line in memory, the line from the disk program will replace the line in memory.

The only requirement for a successful MERGE is that the program on disk must have been saved in ASCII format—that is, the program was saved with the command SAVE "PROGNAME.BAS",A. MERGE is most useful when combined with RENUM as a means of including often-used subroutines from a collection of such routines on disk. By structuring your programs so that certain types of subroutines always have the same line numbers, your time spent typing can be greatly reduced by keeping these files in ASCII format on the disk and merging them in where appropriate.

EDIT is the command that allows Basic lines in memory to be altered character by character. In this way, it's like a simple text editor, and many of its subcommands are similar. Although most users are familiar with the often used subcommands, D-elete, I-nsert, L-ist, and so on, most don't make full use of the EDIT features.

The various less common subcommands in EDIT mode can be real timesavers during debugging sessions and are well worth the trouble to get to know. The best and only way to learn them fully is to spend some time actually using them. It's recommended that you enter a simple two-line or three-line program and then, using the manual, try all the various options to become familiar with their functions. Practice until full use of the subcommands is second nature; it'll be time well spent in the long run.

The final direct command, LLIST, is used in place of PR#x <RETURN> LIST, to output a program listing to a printer connected to the system.

According to user response, the indirect commands seem to generate more confusion than their direct counterparts. This is probably because they're more complex, for the most part, allowing more options and providing wider flexibility. Of these commands, BEEP, ERR/ERL, ERROR, LPRINT, LPRINT USING, and SWAP are all well documented and seem to cause little trouble among users. BEEP, of course, is the way in which the Apple speaker is accessed without having to write an assembly language program or do a series of peeks or pokes. Although no attempt was made to ensure adherence to the chromatic scale when BEEP was implemented, a passable scale can be worked out by varying values for pitch and duration, the two parameters that can be supplied.

ERR, ERL, and ERROR are very useful when combined with ON ERROR GOTO to ensure that the user of your program gets a proper error message and that your program recovers properly during error conditions. It's often useful to define your own error codes for occasions when users of your programs can legally perform an operation in Basic but the operation is one you wish to prevent (see example two under the ERROR command in the *SoftCard Manual*). ERROR is also useful during debugging to test error handling subroutines safely by simulating those errors that can potentially damage data files or otherwise cause nasty side effects.

LPRINT and LPRINT USING are the two commands that control output to the system printer of data generated by the program. This makes them similar to LLIST because they replace the PR#x PRINT PR#0 that is required when printing hard copy under the Apple Basics. Although the commands themselves are relatively straightforward, some users have been dismayed by the fact that unlike Applesoft and Integer Basic, Basic-80 does not explicitly allow using the same print statements for both screen and printer output of data.

For those of you who wish to print conditionally to one or the other based on the way an operator answers a question or on whether the system has a line printer connected, this is a fairly trivial process. Essentially, it involves switching vector addresses for console output and list device output, thereby fooling CP/M into thinking it's outputting to the console when it's actually outputting to the list device and vice versa.

This is possible because CP/M uses a vector table in memory to tell it where to output characters based on the type of output it has. There are, in fact, two console output vectors, and the choice of which one to use is controlled by a single memory location called the IO byte. By installing the address of the list device in the secondary console output vector in the beginning of your program, you can switch outputs merely by changing the IO byte with pokes and peeks. This is usually done with a setup section (lines 10 through 30) and two subroutines (lines 40 and 50), as shown:

```
10 CONOUT=&HF388:Rem Console Output Vector #2
20 LSTVEC=&HF392:Rem List Output Vector #1
```

```
30 Poke CONOUT, Peek (LSTVEC) :Poke CONOUT+1,
Peek (LSTVEC+1) :Goto 60
40 Poke 3, Peek (3) or 3:Return:Rem Switch Vectors
50 Poke 3, Peek (3) and &HFC:Return:Rem Switch Back
60 Rem Continuation of program
```

These subroutines can then be called as needed based on the truth or falsity of a variable, called *noprntr,* for example. Set-up is as follows:

```
100 Input "Do you want output directed to the system
printer, (Y/N)";A$
200 If A$="Y" Then Print "Set paper at top-of-form and
turn printer on" Else NOPRNTR=-1
```

Finally, when you are ready to print your reports, decide whether to switch vectors based on the logical value of noprntr as follows;

```
1000 If Not NOPRNTR Then Gosub 40
1010 Rem Series of print statements for generating report
2000 If Not NOPRNTR Then Gosub 50
```

Although Basic (or CP/M, for that matter) cannot explicitly tell if a printer is connected to the system, during the initial cold boot process it does look in each of the Apple slots for peripheral cards that it can recognize so these cards can be properly initialized. For the cards it does recognize, it places a value associated with the card type in a table in memory called the *slottypes table.* The location in the table for slot 1 (where any printer card would be) is at Z-80 address F3B9H. Examining this location with the peek command will return a single digit value. If the card in slot 1 is an Apple parallel printer card (or any card impersonating one), the value will be 5.

Although you can make your decision on whether or not to switch vectors based on the value in the table, this is somewhat dangerous since the fact that the card is installed is no guarantee that the printer is connected or functional. If you wish to do so anyway, this information should allow you to.

Finally, SWAP is a command that looks simple but can make a big difference in the speed of execution of your programs during sorts, especially during string sorts. When string assignment statements are executed, Basic-80 does not differentiate between strings that have existed before and entirely new ones. In both cases, it writes the string data into memory rewriting it if it was already there, as in the case of A$=B$. Then it updates the string variable table, setting up a pointer to the new string entry either in the old table entry for the variable or in a completely new entry if the variable didn't exist previously.

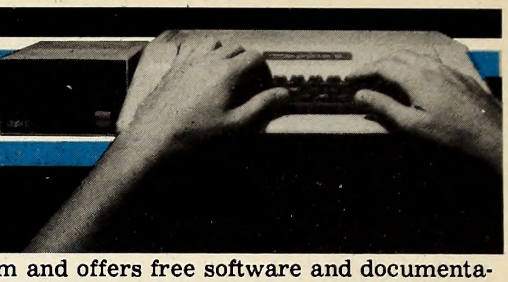
Although this is reasonably efficient for routine assignments, it should be obvious that in sorts and other algorithms in which two values may be swapped hundreds or thousands of times, all this rewriting of strings consumes lots of time. SWAP, however, allows Basic simply to switch the pointers in the variable table between two variable entries. It also results in shorter code, as you can see by comparing the program segments:

```
100 IF A$(I)<=B$(I) THEN 140
110 TEMP$=A$(I)
120 A$(I)=B$(I)
130 B$(I)=TEMP$
140 NEXT I
```

```
100 IF A$(I)>B$(I) THEN
        SWAP A$(I),B$(I)
110 NEXT I
```

We've examined the simpler indirect commands; we'll wait until next month to examine the more complex ones. We'll also explore the collection of commands that are simply implemented under Basic-80 differently from the way they're implemented under Apple's Basics. Also next month, we'll get into the specifics of *APDOS,* the Apple DOS to CP/M transfer program, and some of the ways to speed conversion of Applesoft or other Basic programs to programs executable under Basic-80. It really isn't as difficult as it looks in most cases. We'll look, too, at file handling commands and the differences in file structures. ■

TRADETALK



☐ **Southwestern Publishing** (Cincinnati, OH) and **Apple Computer** (Cupertino, CA) have joined in a cooperative effort to develop business educational software, marking Southwestern's first entry into the personal computer education market. According to Southwestern president **William Wanous** and Apple's director of education marketing, **Greg Smith,** the two organizations had been aware for some time of the ever-growing demand for computers in businesses and in schools. Southwestern, a division of Scott Foresman, has scheduled Spring 1982 for release of a number of business educational programs exclusively designed for Apple.

☐ **Touche Ross** (New York, NY) an international accounting and consulting firm, has installed sixty-five Apple II Plus systems in its major offices throughout the country. Touche Ross bought its first Apple last November for experimental use in the tax department of its New York City headquarters. In May, Touche Ross's managing partner and CEO, Russel E. Palmer, announced that Apples—equipped with disk drives, language cards, monitors, and *VisiCalc*s—were to grace each of the firm's larger American offices. Touche Ross is looking forward to the Apple increasing productivity, lowering costs, and replacing time-sharing services.

☐ **Edu-Ware Services** (Canoga Park, CA), makers of *Algebra I* and other self-paced educational programs, is seeking test sites for evaluation of its special learning need software. Director of research and development **Sherwin Steffin** asks administrators and instructors who use micros in primary and secondary school teaching to participate in the software evaluation program and offers free software and documentation in exchange for critiques.

☐ **Marshall Graham,** president of **The Source Telecomputing Corporation** (McLean, VA), reports that his staff has grown from fifty-seven to ninety-six employees in the past hundred days and expects the number of subscribers to the Source—the majority of whom are Apple owners—to more than double in the next year. The Source, a subsidiary of the Reader's Digest Association, also announces the addition of three key officers: **A. Martin Clark,** vice-president, marketing; **Aksel Olesen,** vice-president, technical development; and **Michael J. Rawl,** manager of advertising and public relations.

☐ **Broderbund Software,** producers of *Snoggle,* the *Galactic Empire Saga, Alien Rain,* and the *Run Time Payroll,* are pulling up stakes in Eugene, Oregon, to move to the heart of the microcomputing community. Broderbund will now be located at 2 Vista Wood Way, San Rafael, CA 94901. Founders **Doug** and **Gary Carlston** expect their San Francisco Bay area location to facilitate import and export activities.

☐ Growing pains may be inevitable when a corporation like the **Biomation Operation of Gould Inc.'s Instruments Division** (Santa Clara, CA), maker of the model *K101-D* logic analyzer, expands to the point of needing a large new facility. But the pains shouldn't last long when employees settle in to their 87,600-square-foot quarters, which will house logic analyzer and waveform digitizer manufacturing. The building, which has a target completion date in early 1982, will be adjacent to the company's present quarters.

☐ **Mel Wong,** president of **Artificial Intelligence Design Specialists** (San Francisco, CA) opened one of the city's first microcomputer dealerships in 1977, and by 1978, the company was dealing in Apples exclusively. Last month, AIDS cut the ribbon on what Wong believes is the largest microcomputer store in the world. The second AIDS shop, located at 271 Sutter Street, San Francisco, deals only in Apples and their accessories, including printers, monitors, and stands. Apple executives were among the hundreds of well-wishers who saw the unveiling of *Omninet*, an office of the future featuring several Apples interfaced with a central storage system by **Corvus.**

☐ **Nahib's** (Evanston, IL), a Chicago-based computer dealer and distributor, announced that it will oversee Midwest distribution of the **Microsoft** (Bellevue, WA) line of Apple and CP/M software and hardware.

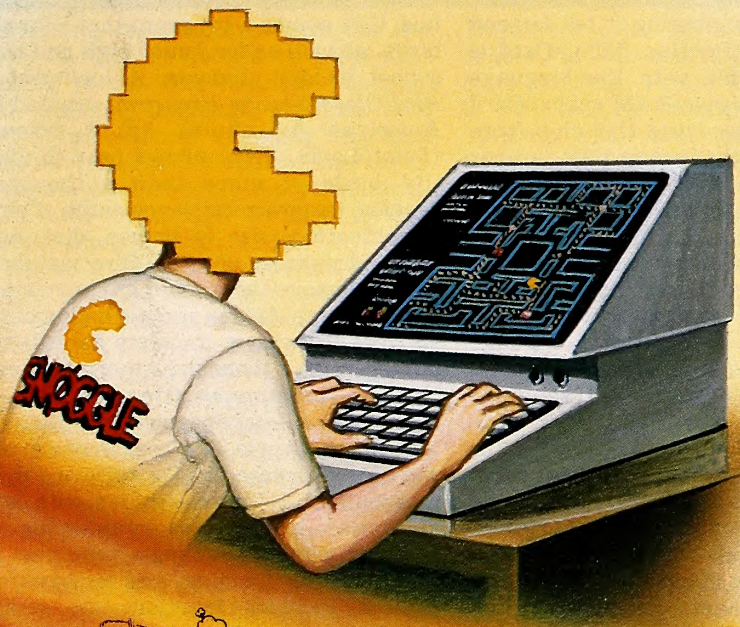
☐ It was a major escalation of already hot-and-heavy war gaming when *Origins VII,* the seventh annual **Academy of Adventure Gaming Arts and Design** convention in San Mateo, California, opened its competition for excellence in game design awards to computer games. First-year nod for highest achievement went to **Automated Simulations** (Mountain View, CA) for its fantasy role-playing game *Temple of Apshai,* which made several appearances this year in the *Softalk* Top Thirty.

☐ **Design Technology,** (Baltimore, MD), new specialty manufacturers of furniture and fixtures for microcomputer users, announces its first-year officers as president **Jack Harvey,** vice-president of operations and personnel **Betsy Sherriff,** vice-president of finance **Andre Vernot,** and vice-president of planning and development **Jim Salmons.** According to Salmons, the company uses computer-assisted manufacturing to provide low-cost, high-quality products. Design Technology also plans to create learning centers for schools.

☐ **California Computer Systems** (Sunnyvale, CA) will distribute **Graham Dorian Software** with CCS line of S-100 systems.

☐ **David J. Wertzberger** is the new director of marketing and sales for **SSM Microcomputer Products** (San Jose, CA). ◨

HI-RES MASTERPIECES



SNOGGLE

You are Snoggle, fleeing through a maze of ghosts who will eat you if they catch you. You have to be quick, you need to be bold, to master the eight levels of this fast-action puzzler. From the Japanese masters of hi-res animated graphics. Apple II Integer/Plus, 48K **disk. $32.95 Now for keyboard and joystick**

ALIEN TYPHOON

In brilliantly colored array, the Aliens swoop down from all sides in dazzling, swift attacks to do battle upon the lone defender. Keyboard or paddle controlled. Twice as many aliens, twice as fast, as Alien Rain. Apple II Integer/Plus, 48K disk — $24.95

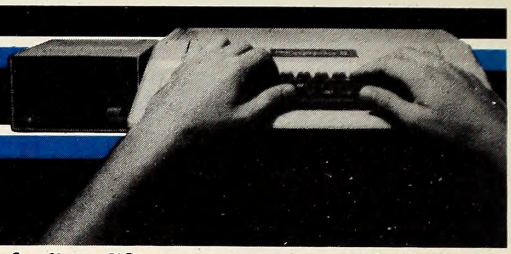
NEW!

SPACE WARRIOR

You are the space warrior, attacked on all sides by multi-colored waves of drone ram ships. You must destroy them all before they pierce your shields and take you to pieces. Hi-res animation from the author of Planetoids! Apple II Integer/Plus, 32K disk — $24.95
16K cassette — $19.95

Brøderbund Software **#2 Vista Wood Way, San Rafael, California, 94901 (415) 456-6424**

MARKETALK News



Unless otherwise noted, all Marketalk News products can be assumed to run on both the Apple II and the Apple II Plus and to require 48K and one disk drive. The requirement for ROM Applesoft can be met by RAM Applesoft in a language card.

☐ *Attendance Reporting System,* from **Educational Services Management Corporation** (Raleigh, NC), has applicability for primary or secondary schools of any size. Reports include daily attendance, individual student history, and homeroom rosters; summary reports include total absences, total unexcused absences, and total excused absences. System is designed for clerical staffs with no prior computer experience. Completely self-contained with built-in functions for such operations as disk formatting, compressing, and copying for backup. 64K, DOS 3.3, two disk drives, printer. $550. Demonstration version consisting of user's manual, program, and data disk available for $35, which can be applied to purchase price.

☐ A *64K RAM Card* is available from **Great Lakes Digital Resources** (Detroit, MI), which is distributing it for **Legend Industries, Ltd.** (Pontiac, MD). Card is directly compatible with the language card, doubling its own RAM space. Card is installed by removing 4116 chip from motherboard and inserting DIP jumper into the vacant socket. The 4116 is reinserted into the 64K card. $349.

☐ *Microcomputers and the CPA Practice,* a two-day executive seminar, shows how to apply profitably the latest advances in small computer technology. Covers hardware components, language and operating systems, typical financial systems, communications, and the marketplace. Seminars will be held in various locations throughout the country. For more information, contact Margaret Sharpe, P.O. Box 2866, 3651 Cedar Lane SW, Roanoke, VA 24001.

☐ Home and small business users need no programming knowledge to operate *Data Base Application Series with Filewhiz* from **Softhouse** (Rochester, MN). Data management system creates files according to specifications; built-in command processor eases editing; search feature and arithmetic features work with subfiles; text features aid in hard copy generation. *Filewhiz* is also for programmers—comprehensive manual explains how to access files for use in Basic programs. DOS 3.2 or 3.3. Printer compatible but not required. $39.95.

☐ Two new products from **LJK Enterprises** (Saint Louis, MO). *Lower Case Character Generator* provides lowercase mode via EPROM. $34.95. *LJK Disk Utility* is a menu-driven program that converts any text, binary, or source file into any of the others. User is provided with constant update of length of file and amount of memory remaining. Other functions include change drive and slot number, lock, unlock, and delete any file. $29.95.

☐ Keep track of your money—or lack of it—with *The Accountant,* a financial data base system for the home or office from **Decision Support Software** (McLean, VA). Dr. Ernest Forman of George Washington University, who developed the program, says it will reveal financial strengths and weaknesses. User will know where assets are, how much they are appreciating, how much user can afford to invest, and how much to keep liquid. Natural dialogue and user-oriented features enable *The Accountant* to offer advantages of an automated double-entry system to both accountants and those who know nothing about debit and credit. $89.95.

☐ From **Imprint Editions Ltd.** (Fort Collins, CO) comes a program that's for the birds, as well as for junior high and high school biology students. *Aviculturist II Bird Classification Program,* created by **American Avicultural Art & Science** (Saint Louis, MO), allows user to identify birds of more than a thousand species. Program retrieves name of bird, classification, size in inches, diet, nest site and number of eggs. Bird watchers can also store their findings, to retrieve and compare. $35 for sample listing of 125 entries; $50 with a complete data listing of more than a thousand American birds.

☐ **Scott Instruments** (Denton, TX), introduces the *Vet/2* speech recognition device for the Apple II. Consists of a hardware preprocessor and a software driving routine. Software resides with host computer, preprocessor analyzes an acoustic signal. Package provides all functions for training and recognition. System includes functions for saving vocabulary sets to disk, calling vocabulary sets from disk, and selecting subvocabularies. *Keyvet* feature makes voice input appear to the Apple as though it came from the keyboard—voice and keyboard can be used interchangeably. With *Keyvet*, the user can run existing software with voice without modifying the software. $895.

☐ After eighteen months of research and improvement, **Sirius Software** (Sacramento, CA) introduces *The Pascal Graphics Editor.* Complete package creates fonts, pictures, and images within the Pascal environment. Produced in a form compatible with the Pascal Drawblock procedures. Can be used directly for display or animation with included utilities. Any portion of the graphics screen may be defined as a Drawblock array and moved, rotated, inverted, crunched, or expanded either horizontally or vertically. Editor includes simple commands for dots, line, rectangles, or parallelograms, ellipses, circles, and color fill routines. $114.95.

☐ A version of the standard language of business software, *CIS Cobol with Forms-2,* has been developed for the Apple II by **Micro Focus** (Santa Clara, CA). Cobol, known for being easy to read and write and easy to learn, is used extensively in business applications. Features include dynamic call/cancel (for large programs) fast indexed sequential, interactive debug, fips flagging for easy conversion. CIS Cobol, $850; with *Forms-2,* $1,050.

☐ *Spellbound with Binder Mechanics* from **Micro-Media** (Arlington Heights, IL) attacks a list of seventy-five fre-

quently misspelled words. Companion programs give students extra drill in finding and correcting spelling and usage errors. *Binder Mechanics* allows instructors to create sentence drills up to two hundred twenty-five characters in length. $19.95.

☐ *Micro-DSS/Finance* from **Addison-Wesley** (Reading, MA) provides the financial manager/analyst with the majority of features needed by most time-sharing users. This financial modeling and planning language provides automatic calculations including depreciation, internal rate of return, and break-even analysis. Flexible report writer allows user to design reports to own specifications. Pascal. $1,500.

☐ Apple owners with inferiority complexes because they don't know how to play adventure games will soon have *The Birth of the Phoenix* from **Phoenix Software** (Lake Zurich, IL), a tutorial designed to teach novice players the intricacies of mapping mazes, finding secret doors, and mining treasures. Handbook included. $14.95. For those who are already adventure gamers, Phoenix's *Adventure in Time*, is rated for class four players. Go back in time to thwart a master criminal and save the world. $29.95. Both programs available September 15.

☐ Version 6 of *Micropayroll*, from **Alternative Software** (Cherry Hill, NJ), handles 199 employees on a single or dual-disk drive system. Program calculates just about any combination of pay situations ranging from biweekly to semimonthly for salaried or commissioned employees. Program also handles simple and complex formula tax deductions, Social Security, and other deductions. $349.

☐ Professional tax practitioners can counsel clients with help from *Individual Tax Plan* from **Aardvark Software** (Milwaukee, WI). Program isolates the tax effects in items of income or expense, then performs tax planning calculations within seconds, according to Aardvark. Computations include federal tax liability, and capital gain deductions. $250.

☐ *PenUltimate*, by **Computer Solutions** (Mansfield, Queensland, Australia), is designed to be easy—but its repertoire of functions is vast. You can print many copies and insert specific information into each—plus lots more. Fast recording of documents. Compatible with wide variety of printers. $295.

☐ Appropriately, *First Edition* is the first of a series of integrated software packages planned by **PRP Programming** (Sykesville, MD). Provides general purpose data entry, edit, and storage capabilities. May be used with DOS 3.3 and printer. Package may be customized through the addition of user-designed Basic or assembly language modules or through a series of *Softkits* soon to be released by company. Disk, $70; cassette, $75.

☐ **Clone Software** announces a new disk-based, line-oriented text editor and 6502 assembler. Text editor is a separate program that can be used for purposes other than editing source programs; assembler helps reduce debugging time through cross-reference feature. $39.95.

☐ Many educational programs will be available this month from **THESIS** (Garden City, MI). *Fishing for Homonyms, Word-Scramble, Word-Mate,* and *Preschool Fun* will be available on both cassette and disk. Prices under $25.

☐ *VisiNews*, a monthly newsletter on *VisiCalc* and other business planning tools, from **Twenty-First Century Media** (Northport, NY), has temporarily suspended publishing until a new publisher can be found. Subscription money will be returned, and no new subscriptions will be taken until further notice. Persons interested in subscribing when the publication resumes should send an inquiry to *VisiNews*, c/o Wes Thomas, 606 Fifth Avenue, East Northport, NY 11731. The address printed in the August issue of *Softalk* for *VisiNews* is no longer in use.

☐ *Mychess* is now available from **Datasoft** (Northridge, CA). Winner among micros at the World Computer Chess Championships in Linz, Austria. Has nine levels of play. Requires Z-80 Softcard. Five-inch disk, $34.95; eight-inch disk, $49.95.

☐ *Space Adventure* is the latest hi-res color graphics game from **Sierra Soft-**

ware (Las Vegas, NV). Journey through the universe aboard the space cruiser *Callisto*, encountering aliens and space phenomena. Object is to solve the puzzles and accumulate as many points as possible. $29.95.

☐ **Novation** (Tarzana, CA), introduces a multicapability modem that can communicate on a variety of networks and operate at various baud rates and functional modes. Seven LSI modem modules are available, including *Phone Line Interface* with auto or manual answer, pulse dialing control, and multitiming function; and *Deaf Modem Module*, which provides interface for communicating with the TTY network for the deaf. Single unit pricing begins at $99.

☐ *Power Text* word processor for Pascal users from **Beaman Porter** (Harrison, NY), allows user to produce perfectly formatted letters, memos, reports, presentations, and manuscripts without formatting the copy as it is typed. Company says system prints document exactly as the user wants it. Full insert, delete, and global search and replace capabilities. Presently configured for Qume, Diablo, and Epson printers. Language System, Pascal. $199.

☐ Remember *PUP-I*? Well, now *PUP-II* has joined the litter from **Gryphon** (Silver Springs, MD): User-friendly *Pascal Utility Package* is designed specifically for the Apple Pascal 1.1 environment. Moves Pascal text files to a Basic disk and displays and modifies any byte from a Pascal or Basic disk. With *PUP-II* and its complementary package, *PUP-I*, edit your Basic programs with the Pascal editor, use Basic communications programs to transfer Pascal files, and vice versa. Features 40/80 column formats, upper and lower case, single or multidrive operations. Easy to use for Pascal beginners. $29.95.

☐ *The Coloney Intelligent Learning Carrel*, from **Coloney Productions** (Tallahassee, FL), is a complete interactive learning system housed in a customized learning station. Carrel combines the features of computer-assisted instruction with the impact of interactive video. Contains Bell & Howell Apple, TV monitor, videodisc player, and interface package. $6,500.

☐ *Basic Mailer* from **Artsci Inc.** (North Hollywood, CA) is a mailing list merge system designed to take *Magic Window* documents. Replaces names, addresses, or any other section of a document with individual data, creating customized letters, statements, invoices. Prints on letterhead (single sheets) or tractor feed paper. DOS 3.3. $69.95.

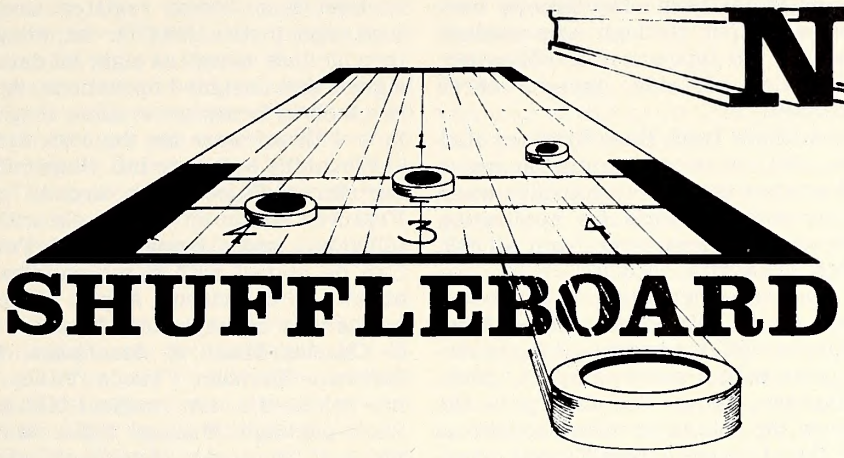
☐ Two boot or not to boot? There's no question with *Two Boot*, the new double-DOS bootswitch available from **Output** (Plymouth, MI). Flipping a rear-panel switch will allow booting either DOS 3.2 or 3.3. Printed circuit board rides piggyback on disk controller card and is small enough to permit full use of slot 7. Dealer inquiries welcome. $24.95.

☐ New version of *Snoggle,* the arcade game from **Broderbund** (San Rafael, CA), offers joystick option for control. Same game. $32.95. Original version can be updated for $7.95 plus $1.00 shipping and handling.

☐ **Minnesota Software** (White Bear Lake, MN) introduces *Apple Lowercase Module.* Requires no cutting or soldering, compatible with all programs that allow lower case, displays lower case with true descenders. $39.95. $49.95 for Apples with RAM configuration blocks.

☐ **Eastern House Software** (Winston-Salem, NC) has released their unbundled *MAE Software Development* system. Software developers can transform several different mirocomputers into a development system using this package. Package eliminates need for software developers to relearn a new set of syntaxes or commands because this software works similarly on several different systems. Includes the only 6502 UC Macro Assembler. Designed to aid program-

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

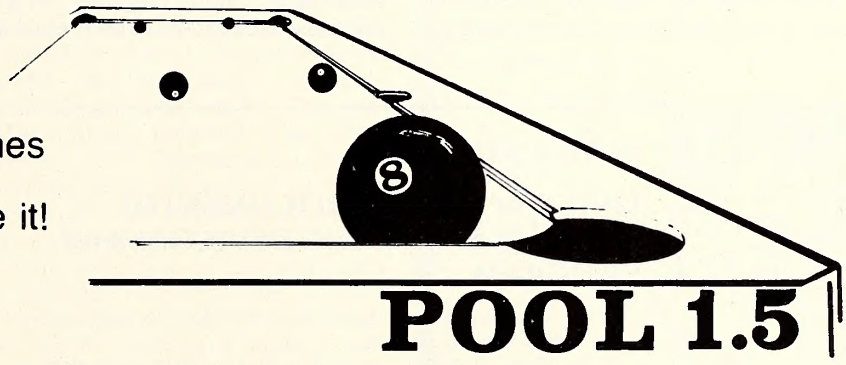


- *Real time HIRES Color Graphics*
- *Play Against your APPLE or another opponent*
- *An accurate and challenging game following in the tradition of Pool 1.5*

— *Another first from IDSI*

POOL 1.5 features

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



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

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

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

MasterCard OR VISA

We accept
Visa, MasterCard,
Check or Money Order.

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

mer productivity via text editing capabilities and other programmer development utilities in package. Main features include: macros, conditional assembly, interactive assembly, up to thirty-one characters per label, string search and/or replace. Other utilities provided depending on version such as word processor, forward/reverse scrolling, tape interface. $169.95.

☐ **Avery Label** (Azusa, CA) a division of Avery International, now has customized computer labels for small businesses. Customized shipping, addressing, price marking, cassette, and other labels can be ordered in quantities as small as five thousand with delivery in less than four weeks. Prices vary.

☐ **International Computer Systems** (Coral Gables, FL), has created *Inventory Control System,* a package that integrates sales, purchasing, accounting (accounts receivable), and generates reports for each category. ICS keeps track of 26,000 items allowing you to keep track at point of sale, have access to physical inventory lists, turnover reports, reorder reports, item profitability, and price lists. Also provides access to sales journal, sales records by customer, receivables by invoice number or customer, and commissioned report, as well as daily cash receipts for nightly deposits. $1,500. *General Ledger* program in Pascal also from company provides journal entries restricted only by size of your hard disk. Allows three levels of sub accounts and use of fifteen digit numbers. Ideal for foreign markets, the program can either display the screens in English or Spanish.

☐ *Dynatyper,* a typewriter interface system from **Rochester Data** (Rochester, NY), generates hard copy directly from computer output through any electric typewriter. No typewriter modification, low profile, portable, weighs three pounds. $499.

☐ From **Micro Data Base Systems** (Lafayette, IN), decision pamphlet suggests and describes features to consider when choosing software tools for application system development. Free from MDBS, Box 248, Lafayette, IN 47902.

☐ In the new category of computer games, *Castle Wolfenstein* from **Muse** (Baltimore, MD) combines an arcade-type game with a more complex adventure/fantasy game. Scenario puts the player in the role of an Allied soldier as World War II rages across Europe. Player is captured but must find the Nazi war plans and escape from the castle. Sound effects include guns firing and Nazis shouting—in German. $29.95.

☐ *Attendance Tracking System* from **Edu-Comp, Inc.** (Renton, WA) automates school attendance record keeping with daily absence reports, cumulative absence reports, parent notice reports, and so on. $995.

☐ The *Apple 9 Processor Board* from **Southern Semiconductor** (Norcross, GA) plugs into any slot in either the Apple II or Apple II Plus. Provides access to 6809E microprocessor in addition to the 6502. 6809E runs at full speed, two accumulator/four index register architecture, and instructions for handling sixteen bit data as well as eight-bit data with signed and unsigned operations. Apple 9 has built-in firmware to allow it to interface with software for the 6502. $379.

☐ **Piccadilly Software Inc.** (Summit, NJ) introduces *Falcons,* an arcade game. Features action on five levels with "intelligent" space beings called Falcons. Can be played with game paddles, joystick, or keyboard. Apple III option works only on keyboard. $29.95.

☐ **Charles Mann & Associates, Micro Software Division** (Yucca Valley, CA) has released a new medical billing software package. *Medical Office Management II* manages physician's appointments, private patient billing, third-party claim for preparation, diagnostic and treatment records, patient report preparation, mailing lists, referral letters, and general word processing. Single CPU license is $959.95. Appointment, private patient billing, and claim for preparation elements are available separately for $449.95. Medical records element, including versions supporting the Corvus systems hard disk, is also available separately. System documentation manuals are available on a preview basis for $50.

☐ From **Omega MicroWare** (Chicago, IL), a new approach to disk and memory management, *The Inspector,* allows user to perform useful alterations to disks as well as memory. Can also repair blown disks, map disk space, and can search and edit a portion of or the entire RAM memory. $49.95. Company also offers a new price for the *RAMEX 16* memory expansion board. Allows memory expansion and will hold several languages. $129.95. Omega also offers a new version of *Locksmith,* a program that will back up disks. $99.95; updates on previous versions, $20.

☐ *Clean Cycle Kit* from **Inmac** (Santa Clara, CA) helps prevent data errors by removing oxide particles that accumulate during normal operation. Cleaner resembles a standard disk and is inserted into the drive the same way, allowing access to the head without disassembly. Inner polyester scrubber and a cleaning solution remove contaminants while lubricating the head to minimize friction and head wear. Each kit good for at least sixty applications. Prices start at $59 for each kit. Discount on purchases of three or more.

☐ A mainframe capability to users of microcomputers available soon from **Datalink Microcomputer Systems Limited** (Bristol, England). *Micro Planner* is de-

signed to plan and control time, resources, and progress involved in any complex project such as construction, organization of events, new product launches, and so on. Company says this type of modeling and control has only been available on mainframe computers with the disadvantages of cost (ten times micro cost at least), access and operational complexity. Uses technique of Network Analysis (or PERT), the program in conjunction with an Apple II.

☐ A plastic laminated card with most of the Apple programming commands and error messages has been published by **Irv Brechner Enterprises** (West Orange, NJ). The *Apple InfoCard* places the most commonly used commands and error messages in an easy-to-read format for quick reference. $3.95.

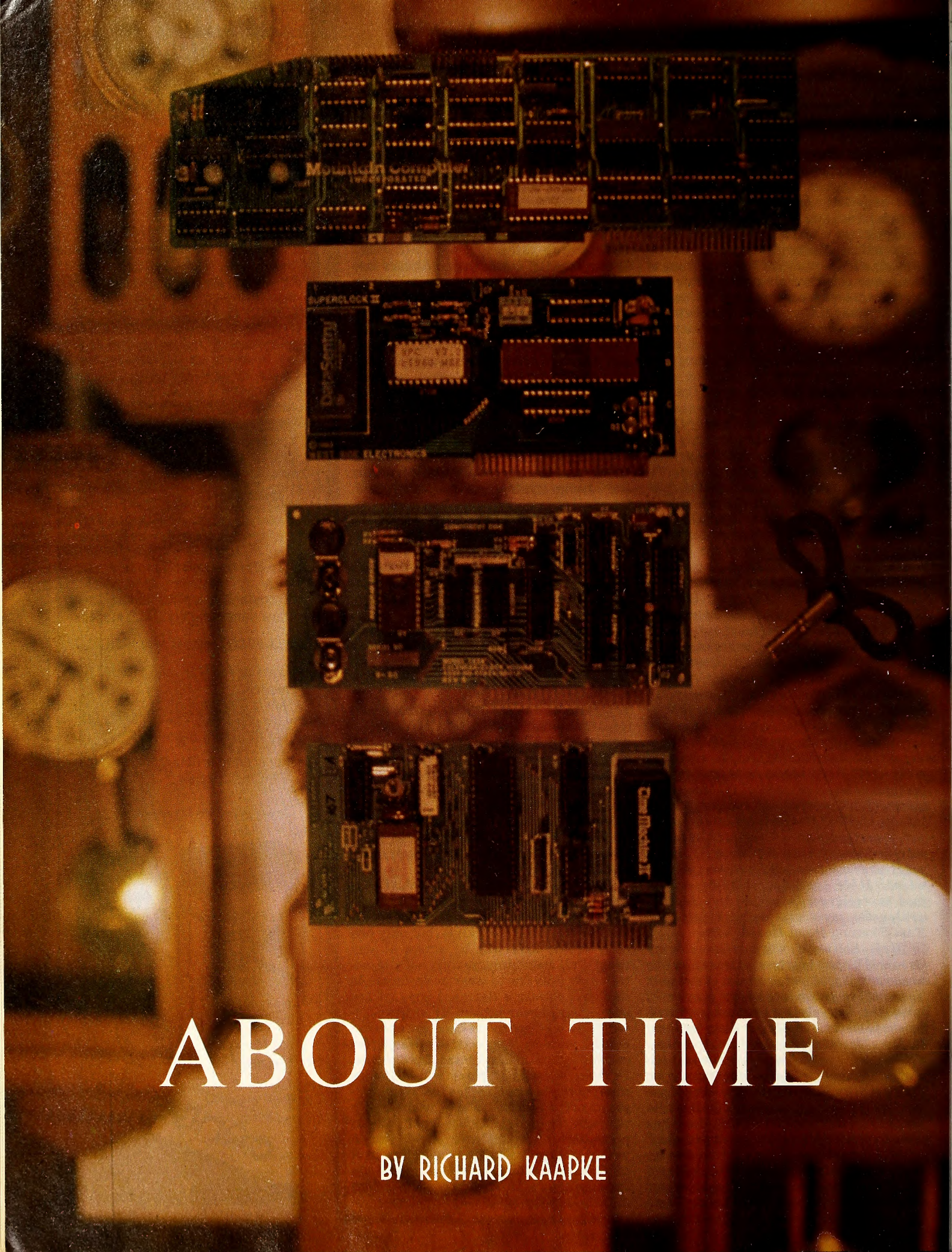
☐ A slew of new products from **Computer Station** (Saint Louis, MO). *Enhanced Graphic Software* for the Epson MX-80. Driver allows user to dump the contents of the hi-res pages out to the new printer to obtain hard copy graphics. $44.95. *Ultra Hi-Res Graphics* is a plotting program. $49.95. *The Boot Button* allows users who've upgraded to 3.3 DOS (sixteen-sector) to boot a disk in 3.2 (thirteen-sector) with the push of the button. $34.95. *Enhanced Graphic Software* for the Centronics 739. Driver allows user to dump contents of hi-res pages to the new Centronics printer to obtain hard copy graphics. $44.95. *Macro-Sced* version 2.0 now offers the following enhancements: search and replace, use of lower-case adaptor, expanded macro table—over 256 bytes. $49.95. Registered owners of the original version may upgrade to Version 2.0 for $15. Plus, another enhancement to Computer Station's *Fast Floating Point Board.* Any Apple equipped with a 16K RAM board or language system can take advantage of the increased speed of transcendental function calculations of FFP Board in existing Applesoft Basic programs. User installs patch supplied by company for using FPBasic, patched version called *Applefast* is loaded into RAM card upon booting. *Applefast* is designed to utilize the AM9511 arithmetic processor as configured on Fast Floating Point Board. $475.

☐ **Krell Software Corporation** (Stony Brook, NY) has four new releases for the academic Apple owner. *Isaac Newton* is an inductive game that allows players to intervene actively by proposing experiments to determine if new data conform to the "laws of nature" in question. $24.95. *Odyssey In Time*, a historical adventure game, includes challenges for the time traveller who yearns to visit the eras of Alexander the Great and Attila the Hun, to name a few. $39.95. Krell has also expanded its *College Board SAT 81/82 Preparation Series* from five to twenty-five programs for the college-bound student. Includes preparation for the word relationship, vocabulary, sentence completion, and mathematic portions of the college board examinations. $229.95. *The Competency Exam Preparation Series* consists of simulated examination modules, diagnostic package, and complete set of instructional programs. Designed to teach concepts and operations, provide drill and practice, and to attain higher achievement levels. CEPS is for the individual student or classroom use. $1,299. Separate mathematics and verbal packages are $799 each.

☐ Don't be left out in the cold—now two can play at this game. **Automated Simulations** (Mountain View, CA) has added a two-player option to *Invasion Orion.* Formerly only a solitaire space battle game, now a game in which each player can control up to nine spaceships armed with destructor beams, missiles, and torpedoes. Expert player can make a challenging game against a human novice by allowing his opponent more powerful ships. *Invasion Orion* has ten scenarios. Also included is a second program that allows players to create more scenarios and design their own starships. 32K. $24.95.

☐ Keep tabs on your neighbors in the computer community with two new directories. *The Community Computerist's Directory* is labeled the "Who's Who of People and Computers." National paper data base in phone book format, it's dedicated to all computer users; published semiannually, updated quarterly. Single issue $3.50. One-year $10 subscriptions includes free white page listing. *The International Microcomputer Software Directory* has 5,000 programs and 15,000 entries that are continually updated with information collected from all parts of the world. Includes system classification, subject classification, and software house classification. $28.95 (plus $2.95 postage and handling). System-specific directories extracted and cross-referenced from the main data base available for $14.95 (plus $1.95 postage and handling). Both from **Alternet** (Forestville, CA).

☐ **Diversified Software Research** (Rockford, IL) announces a system for making disks uncopyable. *Protect-O-Disk* allows user to make backups, but only one copy of the disk will be active at any time. Full backup capability—if the active copy fails due to wear or damage—is used to activate a dormant copy. If the protection is ever broken by pirates, Diversified will try to produce an updated secure version. Protects Applesoft, assembly language, and Pascal programs. Protect-O-Disk duplicator will make as many pirate-proof copies of your program or programs as you wish. For a one-time charge of $1,500, each title is protected by duplicator disk. For a yearly charge of $12,000, all titles in a publisher's catalog can be protected via the company's *Master Disk* service that makes duplicator disks for each program. ■



ABOUT TIME

BY RICHARD KAAPKE

What are computer clocks? And what good are they? What features should you look for in a clock?

These are questions often asked about computer clocks. For those of you who are considering buying a computer clock—and for those who are just curious—this general review will cover enough ground to make you well-informed and (perhaps) determine which computer clock you'd like to use in your Apple computer.

World Without Time. First let's consider real-time. *Real-time* is a term used to denote the passage of time in our world. Our Apple computers don't realize what time or even what day it is. There's no built-in circuit in an Apple that's analogous to a wrist watch for humans. An Apple can perform operations measured in millionths of a second, but nowhere is there an automatic clock/calendar function to add up these little passages of time and relate them to time in the real world.

We all have to conform to business hours, movie schedules, kilowatt-hour rates, and various time-measured fees. Our Apples can help us cope with real-time hassles if we buy them a relatively inexpensive present—a real-time clock. The prices of real-time clock boards (referred to as clock cards, clock boards, real-time clocks, or real-time clock cards) range from that of a digital watch to that of a twenty-one-jewel Swiss job. These clock boards essentially add the same circuitry that's inside a digital watch to your Apple. And, just as the price of a watch rises with the number of features and functions it offers, so does the price of a clock card.

Some of you might wonder, "Why do I need a clock card? The computer could always ask me for the time or date. . . ." This is true; however, there are cases in which this is inconvenient or impossible for the computer.

Our computer, for example, is busy in the laboratory taking measurements and doing calculations. We don't want the computer to stop in the middle of our calculations to ask for the time; rather, our real-time clock card will interrupt the computer at desired times, automatically putting on hold what it was doing until it has serviced the interruption.

Marking Time. This example describes a scientific application of an Apple computer. The terms *interrupt*, *service*, and *hold* are watchwords in many real-time applications. A real-time clock can function as an alarm clock to the Apple. To put it simply, most real-time clocks can set off an alarm for the computer at a specified time, and some even allow the specification of a date along with the time. When the alarm goes off, the Apple gets a special signal called an *interrupt request*. The Apple treats this like a tap on the shoulder and will promptly drop everything to service the interrupt; that is to say, whatever was happening before the interrupt occurred will be put on hold, and a special program—whatever you want the computer to do when the alarm goes off—will run. When this program ends, the interrupt has been serviced, and the Apple will return to what it was doing before it was interrupted.

Many real-time clocks keep time to a very precise level—often to or beyond a thousandth of a second *(millisecond)*. We humans can seldom see a need to measure time to anything less than a second, but an interrupt occurring ten or a hundred times a second would require this. We would quickly lose our minds interrupted this often, but the Apple won't. In fact, being interrupted a thousand times a second is not taxing on the computer—assuming, of course, the task assigned in the space of time is not too great.

Some of the video games you can buy for your Apple read the game paddle controllers and/or keyboard hundreds of times a second. These programs do this in a *loop* where the controllers or the keyboard are checked this often. With a real-time clock, you don't have to make your program read information in these small time intervals. Rather, the clock can interrupt the computer just in time to take the next reading.

World Enough. Some real-time clocks also do other things for you. Two of the three reviewed in this article do more than tell your Apple the date and time. Added features include built-in interfaces for printers and other accessories and appliance controller interfaces. (For those of you picturing someone's head stuck inside a computer, the term *interface* simply means a connection method to put computers and add-on devices together; printers, telephone hookups, and light dimmers can all be accessories for your Apple, but these require interfaces to allow the computer to control them.)

Educators and scientists can benefit from real-time clocks. The time it takes to respond to a question, the time it takes a rat to run a maze, and other measurements like this are often termed *measurements of response time*. Response times can be measured down to the millisecond—helpful information in determining when the student or subject took just a little more time than normal to finish.

Business software is also enhanced by real-time clocks. Instead of asking for the date or time, a program requiring this information can read it from a clock card and affix the information to invoices, transactions, or whatever. Some creative geniuses can use real-time clocks to see how long various parts of programs take to run, down to the thousandth of a second. Some folks are just impressed to the teeth with statistics.

Real-time clocks can also make your computer do two (or more) things apparently at once. Assume you have two things you'd like to do: (1) print out the term paper you just finished writing with your word processor, and (2) play a game as a pastime while the printer is running. Without a real-time clock, chances are that you'll never be able to do this. With a real-time clock, it's pretty easy. Just use those fast interrupts. Most printers print at less than two hundred characters a second, many at less than a hundred. The program that prints can be put in one spot in memory, and the game program in another. The game program would be interrupted a hundred or so times a second, at which time the next character would go to the printer, then go back to the game program. Since the action of sending a character to the printer can be done in virtually no time at all, the game would be slowed down a very small—almost unnoticeable—amount. Most of the time spent printing is spent waiting for the printer to be ready for the next character or line to be accepted. This time can be put to other uses, provided something can remind the computer to check up on the printer once in a while.

In summary, real-time clocks help you by supplying information for your personal and professional records, assist entertainment and professional programs with precise time measures, and provide interrupts to enable two things to get done at once, as well as perform other special functions.

Mountain Time. First and fanciest is the Mountain Computer CPS multifunction card. Isn't the name impressive enough to speak for the product? The CPS card (CPS stands for *clock*, *parallel*, and *serial*—the three functions of the card) contains a real-time clock that keeps time to one second precision (no millisecond measurement), plus two interfaces—one for allowing serial data input and output, and another for parallel data output.

For those of you frowning at that last statement, serial data input and output is the way an Apple and a telephone hookup talk to each other. Telephone hookups, or modems, are devices that send information back and forth over the same telephone lines we use every day. Modems send tones rather than words, and these tones represent the ones and zeros most all computers know and love. With the CPS card, you can type a message on your Apple and have another computer at the other end of the telephone line receive the message, often keystroke by keystroke.

Some printers are also serial data devices; that is, they like to get the ones and zeros over the same connection system that a modem does. Therefore, the serial interface capability of the CPS card can be put to more than one use. Serial data is sent bit by bit; that is, a single character like the letter *A* will take no less than ten ones and zeros following each other in formation. The pattern of these ones and zeros follows a settable rate of speed and format. The CPS allows you to set the rate of speed and format so that the Apple can talk to virtually any other computer or device that uses serial data. Remember that the serial interface of the CPS card is input as well as out-

put, so the Apple itself can receive serial data coming from most any source, computer, teletype, or otherwise.

The parallel interface portion of the CPS card is output only—in this case, eleven wires are used to send the data, one wire for each bit of data. (This time, only eight bits, ones and zeros, need to be sent to make a character. The other three are for signal ground, *strobe*, and *acknowledge*.) Serial data goes at a constant rate; parallel data travels as fast as possible.

Parallel interfaces work like this: the Apple sets each of the eight wires to voltages representing the ones and zeros in the character it wishes to send, each wire representing a specific place in the binary number system both the Apple and the device it's sending to understand. Once these voltages are set, the Apple sends a *pulse*, a quick on-off (or off-on) signal, down another wire, called the *strobe*. The printer or device receiving the data must tell the computer that it accepted the character and is ready for another by similar on-off or off-on switching of yet another wire, called the *acknowledge*. When the Apple receives the acknowledge signal, it knows it can send another character at its earliest convenience. All this sounds troublesome, but it often results in the data being transferred much faster than is possible with serial data lines.

The Phantom Time. The CPS card can do something no other card does (at least at the time this article goes to press): it can generate phantom slot assignments. If this sounds mysterious to you, read on!

Phantom slot assignments are a new and unique ability of a circuit card to perform the job of one, two, or more others. The Apple usually exercises an interface (such as the serial or parallel ones on the CPS card) by using a PR# or IN# instruction. The PR# and IN# instructions essentially tell the computer to print or input its information to or from the device connected to the interface in one of the slots in the Apple. (Is this starting to sound like a song—"A wart on the nose of a frog on a bump on a log in the hole in the bottom of the sea . . ."?)

Remember, an interface is necessary for a device to be controlled or inspected by the Apple. The interface must be plugged into one of the slots found inside the Apple. The slots are numbered 0 through 7. A PR# instruction in Basic will cause subsequent print instructions to the device connected to the interface in slot 7. A special case exists, however: a PR#0 statement does not send printout to an interface plugged into slot 0; rather, it tells the computer to start printing to the screen again.

The phantom strikes again! A CPS card can actually tell if you issued a PR#7, and even if the CPS card is not plugged into slot 7, it can still act as though it were! (Note, this does require the CPS to be plugged into some slot, however.) A diskette with software lets you define who's who in the Apple II—you can set up the CPS card so that slot 1 is your parallel output, slot 2 is your serial input/output slot, and the clock can be read by accessing slot 4—all this and still have the CPS card in slot 5, 6, or 7. For those of you who can't make up your minds, the CPS card can also let you define more than one slot as having the same function, so either a PR#1 or PR#7 can be used to turn on your parallel printer. Note that you shouldn't reassign a slot that is already in use unless it's the one the CPS is plugged in.

The CPS card worked quite well in the test Apple. The operating manual that comes with the card is recommended only to those better-seasoned people who've had some experience with technical manuals, although the installation and set-up instructions are step-by-step and the software supplied with the card is easy to use. The CPS card does not generate interrupts at all; therefore it should be purchased by those wanting more out of a clock than just the date and time. At $239, the CPS card is a good value, provided you need all three functions. In comparison to other clocks, the CPS clock does not provide the interrupt capabilities of the others.

Storm Season. The Thunderclock Plus is a real-time clock card capable of interrupts and returning time and date information in a number of formats, one of which is compatible with the Mountain Computer CPS Multifunction card. The Mountain Computer time format is a standard that most clocks are meant to follow. Most programs written to read the time will accept this standard.

The Thunderclock can interrupt your Apple at rates of 64, 256, or 2,048 times a second—a fair range of selections. The ability to act like an alarm clock that causes the computer to do something special at a certain time is possible, using simple programming.

The Thunderclock works with an interface for light dimmers and other appliance controllers, using the popular BSR X-10 appliance control system that Sears, Radio Shack, and other department stores are selling. Appliances such as coffee pots, electric fans, and lamps must be plugged into *appliance controller modules* or *lamp controller modules*, respectively. These and a BSR X-10 Ultrasonic Command console must be purchased separately.

The BSR X-10 Ultrasonic Command console is smaller than a three-by-five file box and has a small calculatorlike keyboard built into it. A power cord extends from the back of the console into any standard American AC outlet; the controller modules are half the size of the command console and act as go-betweens that switch the power flow on and off between the appliances and the outlets. Lamp controllers do this and also act as dimmers where the power output can be varied from 100 percent down to 2 percent or so. With the X-10 Interface Accessory for the Thunderclock, you can have your Apple computer completely controlling your appliances; it's possible to schedule for weekly, monthly, or even yearly periods. Daily and even random events are available, too. The interface accessory looks like a plastic hose from a fish tank, but it carries the ultrasonic (above the range of human hearing) signals that the command console understands. Using the *Scheduler* software included with the X-10 interface accessory, you can set up a schedule like this:

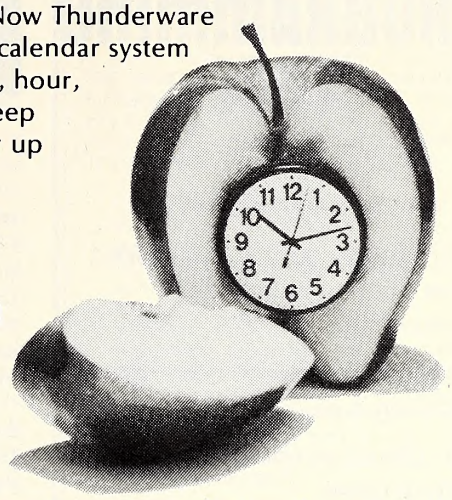
Turn on the lights some indeterminate period between 6:00 and 7:00 p.m. in the house; lights off between 9:30 and 11:00 p.m. (also indeterminate time), except Saturdays, when they'll go off between 11:30 p.m. and 1:00 a.m.

The Best of Times. This is useful in crime prevention at

GOTO 55

PUT YOUR APPLE TO WORK FOR YOU!
WITH THE THUNDERCLOCK PLUS™

As an APPLE user you already know all the things your APPLE can do. Now Thunderware expands that list with the THUNDERCLOCK PLUS, the complete clock/calendar system for your APPLE! Your programs can read the month, date, day-of-week, hour, minute, and second in any of APPLE'S languages. On-board batteries keep your THUNDERCLOCK running accurately when your APPLE is off - for up to 4 years before battery replacement. But that's just the beginning.



The THUNDERCLOCK PLUS is the most useful and versatile peripheral you can put in your APPLE. It can keep your disk files organized by time-and-date-stamping them, it enhances the usability of many of the new business/professional software packages for accounting, filing, and time management, and it can remotely control lights and appliances for security or display purposes in your business or home.

SOFTWARE PRODUCT COMPATIBILITY

Many of today's important software packages for data-base management, business applications, communications, and time management are designed to use the THUNDERCLOCK PLUS. If you have or plan to purchase any of these packages, a THUNDERCLOCK will greatly enhance their usefulness.

•VISIDEX* (Personal Software) •DB MASTER and MICRO-MEMO (Stoneware) •MICRO-COURIER and MICRO-TELEGRAM (Microcom) •THE CASHIER and THE STORE MANAGER (High Technology) •BUSINESS PLUS and NET-WORKS (Advanced Data Systems) ...and many others!

```
DISK VOLUME 254

 *A 006 HELLO                 07/07 16:37
 *A 006 CLOCK                 06/08 09:07
 *A 004 FRAME                 06/08 09:08
 *A 004 DISK INFO             06/17 16:13
 *B 003 BACKOFF               06/17 16:13
 *B 005 SCREEN                07/24 17:32
 *B 002 TCPUTIL               06/17 16:13
 *B 004 SDTIME.O              06/17 16:13
 *A 007 ADIGCLK               05/19 08:05
 *A 011 SET TIME              06/08 09:08
 *I 009 IDIGCLK               05/19 08:05
 *A 007 TIME                  06/08 09:08
 *A 003 SLOTFINDER            07/07 16:56
 *A 014 DEMO                  06/17 16:14
```

THUNDERWARE'S DOS-DATER

Our new DOS-DATER software upgrades the regular DOS on your disks so that DOS will use the THUNDERCLOCK to time-and-date-stamp disk files. Every time a program is saved or a file is modified, the current date and time to the minute are stored in the CATALOG with the file's name. You can tell at a glance when a program was saved or when any file was last modified. And this time/date stamping feature is completely automatic. That means any program which uses DOS will time/date stamp its files!

REMOTE CONTROL

Add Thunderware's X-10 INTERFACE OPTION to your THUNDERCLOCK PLUS and your APPLE can control lights and appliances through your BSR X-10 Home Control System on your pre-defined schedules. Our powerful SCHEDULER software allows you to create and modify schedules easily and execute them in the 'background', while using your APPLE for other tasks in the 'foreground'. Use your APPLE for energy management, display and security lighting, or laboratory/process control.

Our PASCAL Software lets you use all the THUNDERCLOCK'S features in PASCAL and sets the F)iler date whenever you boot.

You get all this versatility in just one peripheral system. Backed by a full one year warranty. See your APPLE dealer for a demonstration, or contact us for more information. We'll give your APPLE the best time around!

Suggested retail prices:

THUNDERCLOCK PLUS	$139
X-10 INTERFACE OPTION	$49
PASCAL SOFTWARE DISK	$29
DOS-DATER/DEMO DISK	$29
MANUALS ONLY, each	$5

Distributed by Apple Computer, Inc. and Computerland Corp.

THUNDERWARE, INC.
P.O. BOX 13322
Oakland, CA 94661
(415)-652-1737

*Requires software supplied on DOS-DATER/DEMO disk.

THE PASCAL PATH

By Jim Merritt

Tools of the Craft, Part 3: Control Flow

Before launching into the long-promised discussion of *control-flow* (followed immediately by *decision*), I must acquaint you with a very important part of the Pascal language: *comparison operators.* At first, I'll just tell you what they are. *Why* they're important will be made clear in the larger discussion.

Last month, I talked about expressions and operators, including those for Boolean values. Assuming you worked your way through last month's material, completing all exercises and quizzes, you should already be reasonably familiar with Boolean expressions. The previous discussion was deliberately incomplete, however, in failing to mention *comparison operators.* In Pascal, you may compare datum values for equality or various shades of inequality by using an appropriate comparison operator, chosen from the chart below. As you can see in figure 1, these new operators act like those you already know. Each takes an operand on either side of it, and produces a single—Boolean—result.

Pascal permits comparisons between any two data objects whose types are *related.* Sensibly, if the data types of the two objects are identical, they are, of course, related. Thus, Integers can be compared with Integers, Reals with Reals, Chars with Chars, and Booleans with Booleans. However, two separate and distinct data types may be related, if they are subrange types that share a common base type. Remember the subrange type HalfAsBig that has popped up now and then in previous discussions?

```
TYPE
  HalfAsBig=
    0 .. 255;
```

Integer is HalfAsBig's base type. In other words, HalfAsBig was derived from Integer, and so these two types are related. Therefore, Pascal allows comparisons between values of type HalfAsBig and values of type Integer.

You may also compare Real and Integer data (as well as Real values and values of subrange types whose base type is Integer). Real and Integer are defined to be related. I've already mentioned that Apple Pascal's internal representation for Integers is markedly different from that for Reals. This would seem to preclude direct comparison between values of these two types, and so it does. When faced with a comparison between an Integer value and a Real one, Apple Pascal automatically converts the Integer to a Real before comparing the two.

Of course, comparisons between totally unrelated types are not permitted, and the compiler treats them as syntax errors (error number 129, to be specific). You cannot, for example, compare Boolean and Char values, any more than you can assign a Boolean value to a Char variable (or vice versa).

Here is an example program, which you should attempt to compile but need not execute. The program contains only assignments and no WriteLn statements, so executing it produces no tangible results. It is intended only to show you that the compiler will indeed catch attempts to compare data of unrelated types. At least one of the assignments will be noted as a syntax error by the compiler. Before compiling the program, try to find the illegal comparison(s). The compiler will let you know if you guessed correctly.

```
PROGRAM
  Comparison;
CONST
  Blank=        ' ';
TYPE
  HalfAsBig=
    0 .. 225;
  UpperCase=
    'A'..'Z';
VAR
  IntVal
    :Integer;
  HalfVal
    :HalfAsBig;
  CharVal
    :Char;
```

Pascal Comparison Operators each produces Boolean result

Operation	Explanation
DataVal1 = DataVal2	True if DataVal1 and DataVal2 are equal
DataVal1 <= DataVal2	True if DataVal1 is less than or equal to DataVal2
DataVal1 < DataVal2	True if DataVal1 is less than DataVal2
DataVal1 > DataVal2	True if DataVal1 is greater than DataVal2
DataVal1 >= DataVal2	True if DataVal1 is greater than or equal to DataVal2
DataVal1 <> DataVal2	True if DataVal1 and DataVal2 are not equal

Figure 1.

```
    CapChar
      :UpperCase;
    BoolVal,
    CompResult
      :Boolean;
  BEGIN
    HalfVal := 128;
    IntVal := 45;
    CharVal := '*';
    CapChar := 'A';
    BoolVal := False;
    CompResult := HalfVal < 2000;
    CompResult := HalfVal < IntVal;
    CompResult := 35 <> 35.0;
    CompResult := IntVal < 100.0;
    CompResult := 'A' >= CharVal;
    CompResult := CharVal > CapChar;
    CompResult := Blank <= CapChar;
    CompResult := IntVal >= CharVal;
    CompResult := BoolVal < True;
    CompResult := False = 'F';
  END.
```

In "Comparison," the first assignments initialize the program's variables, so that subsequent comparisons involve known values. Each assignment to the Boolean variable CompResult fills the variable with the value of the Boolean expression that follows the assignment symbol (:=). In every case, of course, the Boolean expression just happens to be a comparison.

Precedence Revisited. Now that you've been exposed to Boolean operators, let's expand last month's chart of Pascal operator precedence to include them:

Apple Pascal Operator Precedence (Version 2)

NOT (Highest precedence)
*, /, DIV, MOD, AND
+, −, OR
=, <=, <, >=, >, <>

The relational operators have the lowest precedence of all. This can lead to some sticky situations. For example, what do you think Pascal will do with the following assignment (assuming Result and BoolVal are Boolean variables, and IntVal1 and IntVal2 are Integer values)?

```
  Result :=
    IntVal1 < IntVal2 AND BoolVal;
```

Let's use parentheses to clarify our discussion of this expression. You might feel that what the programmer intends is

```
  Result :=
    (IntVal1 < IntVal2) AND BoolVal;
```

But does the Pascal compiler actually see things that way? No, according to the rules of precedence, it does not. Instead, the compiler treats the expression as if it were actually written like this:

```
  Result :=
    IntVal1 < (IntVal2 AND BoolVal);
```

This last interpretation of the expression leads to type conflicts. First of all, AND must have two Boolean values for its operands, but IntVal2 is an Integer. Second, even if this AND operation were permissible, it would yield a Boolean value, which cannot be compared with the Integer value IntVal1.

As you'll see, comparisons and Boolean operators are often intermingled within complicated Boolean expressions. Accordingly, problems of precedence, like the one shown, crop up frequently. To avoid any such difficulties, I find that it's always best to enclose comparisons within parentheses, as in assignment two, thus ensuring that the compiler sees things my way. (In fact, in my programs, I overuse parentheses and almost never trust the evaluation of expressions to Pascal's precedence rules. This is not an example of paranoia on my part, only megalomania—I like to be in control.)

Control Flow. *Linear Flow.* All the programs we've examined to this point have been no more than simple lists of statements. The computer executes one statement in the list, proceeds to the next in line, executes it, and so on, until it encounters the end of the program. There's no skipping around and no doubling back.

Whenever a given statement is being executed, programmers say that *control* has passed to that statement. I often use the analogy of an automobile racetrack to describe program execution. In this analogy, *control* corresponds to the racing car itself and each statement in a program to a checkpoint along the track. When the car hurtles by a checkpoint, this is equivalent to the execution of a single program statement. The kind of path followed by control in its travels through the program is called its characteristic *control flow.*

Using the auto racing analogy, the program that consists only of a simple list of statements is like a dragstrip. That is, the car—control—starts at one end of the strip, travels in a straight line, passing each checkpoint in succession, and finally finishes at the end of the strip, some distance away from the starting point. This drag strip type of control flow is more properly described as *linear* or *sequential.* Linear execution—doing one thing after another—is the simplest and most common example of control flow. When you group one or more statements in a simple list, without including any special instructions that modify the flow of control, you can count on Pascal to attend to each item in sequence.

Looping. It is impractical for the Indianapolis 500 to be run as a drag race. For one thing, I don't think it's possible to lay 500 miles of straight, level, uninterrupted pavement anywhere on the continent. For another, too many pit-stop crews and locations would be required. Finally, there's room for a lot of climatic variation in 500 miles, and the drivers might find themselves having to face wind, rain, sleet, and even snow along the route. (In fact the presence of any such condition anywhere along the track could be grounds for postponing the race. Consequently, the race might never be held, since inclement weather could visit a dif-

ferent stretch of the track every day of the year!)

Of course, for reasons of economy and convenience (not to mention sheer feasibility), the Indy 500 is held on a 2.5-mile oval track, and the drivers travel around and around until they've gone 500 miles (200 laps).

Let's say we want to write a program that displays the distance traveled by an Indy 500 racer, one mile at a time. As you've seen in previous examples, the WriteLn statement can be used to display numbers. So, for every mile traveled by our hypothetical driver, the computer should execute a WriteLn statement that displays the proper, current count.

The only means you now have to display the numbers from 1 to 500 is to write the program in "drag strip style," like so:

```
PROGRAM
  Indy500;
BEGIN
  WriteLn(1);
  WriteLn(2);
  WriteLn(3);
  .
  .
  .
  and so on ...
  .
  .
  .
  WriteLn(500);
END.
```

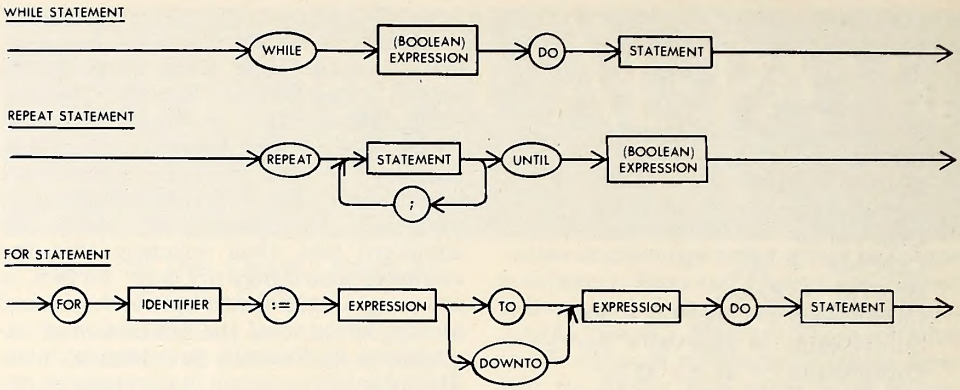


Figure 2.
Apple Pascal Looping Statements.

Clearly, using linear control flow in a program of this sort is as ridiculous a prospect as trying to run the Indy 500 as a drag race. Fortunately, it turns out that you can build an oval track into this kind of program and so force control to flow in a circular fashion through a group of statements over and over again as many times as are necessary to get your job done. Repetitive (or iterative) control flow is commonly known as *looping*.

How a Loop Works. The key elements of a loop are its *body* and its *termination condition.* The body consists of at least one statement (often more), and is executed exactly once for each repetition of the loop. For every iteration, the computer checks to see if the termination condition has been met. When it has, repetitions cease and control passes, in linear fashion, to the instruction (if any) that follows the loop.

Pascal provides three explicit methods of looping, each of which is more appropriate than the others in certain situations. Figure 2 shows the railroad syntax diagrams for the three types of loops. Two of these, the WHILE loop and the FOR loop, imply that the computer checks whether the termination condition holds before the body is ever executed. Thus, if the termination condition holds at the time control first passes to the loop, the loop body will never be executed! The last, the REPEAT-UNTIL loop, involves checking of the termination condition after the body has executed. Consequently, the body of any RE-

PEAT-UNTIL loop must execute at least once. Notice that each loop is considered to be a single statement in itself, even though it must also contain at least one statement as its body.

Before us is the task of displaying a mile-by-mile count of the distance driven during the Indianapolis 500 race. You've seen how such a program *could* be written using 500 separate WriteLn statements, each of which will be executed exactly once. Pascal's looping facilities permit you to write merely one WriteLn statement that will be executed 500 times. Which kind of loop should you use to solve this problem? As a matter of fact, you can use any of them, and I'll demonstrate this by presenting three different Indy500 programs, all of which produce identical results.

How Is a Termination Condition Expressed? Loop termination conditions in Pascal can be either True or False. The implication here is that a termination condition is nothing more than a Boolean expression, and so it is (as you can see by referring to the syntax diagrams). We'll need to keep track of the miles driven for each loop iteration. A perfect way to do this is to use a variable as a counter. Since we'll be counting in whole miles, we can use an Integer variable as the counter—call it *Distance*.

The loop termination condition may be expressed in terms of the current value of Distance. When the Distance traveled is 500 (miles), the program (and the race) should end. In other words, the race car should drive another mile *until* the five hundredth mile has been completed. The use of the word *until* in expressing the termination condition is a tip-off that we might as well start with the REPEAT-UNTIL loop.

REPEAT ... UNTIL ... Let me start by sketching a skeleton for our first version of the Indy500 program:

```
PROGRAM
 Indy500a;

 VAR
  Distance
   :Integer;
BEGIN
 REPEAT
 UNTIL ;
END.
```

The body of the loop—the part that does the real work during each cycle—will be inserted between the REPEAT and UNTIL keywords. The termination condition will be inserted between the UNTIL keyword and the subsequent semicolon. We've already said that Distance will hold the number of miles traveled at any point during the race, and that the race should end when that Distance is 500. The Boolean expression that tells whether or not the Distance is yet 500 is simply formulated:

```
Distance = 500
```

Now, let's tackle the body of the loop. One thing is clear: we'll want to see the value of Distance for every iteration. A simple WriteLn statement will take care of this:

```
WriteLn(Distance);
```

Adding these two parts to our skeleton gives

```
PROGRAM
 Indy500a;

 CONST
  EndOfRace= 500

 VAR
  Distance
   :Integer;
BEGIN
 REPEAT

  WriteLn(Distance);
 UNTIL Distance = EndOfRace;
END.
```

A Word About Generalization. You'll notice that I have chosen to define a constant, EndOfRace, equate it with 500, and use it within the program itself, as opposed to using the literal constant 500. By doing this, I have generalized the instructions so that they can work for a race of any length. To change the length of the race, I need only modify the definition of EndOfRace, then recompile. I won't ever have to tamper with the instructions themselves again.

What I have done here, with the constant EndOfRace, is an example of thinking ahead, of trying to write one's programs so that they can, with little or no modification, solve an entire class of similar problems, rather than only one or two specific ones. Despite the pleasure I get out of it, the fact remains that most programming is a lot of hard work. I am truly a lazy individual and prefer to make a minimum amount of work carry me the maximum distance possible. Within reason, the more general I can make a program, the more often I can reuse it when working on other projects, thus saving myself much time and effort. As we travel along the Pascal Path, I will stop at appropriate points to show you other techniques you can use to promote generalization in your programs.

Back at the Track. . . . The Indy500a program looks complete at this point. It has a one-statement body that seems to do all we require and a termination condition that appears to be sufficient. The fearless may wish to compile the program at this point and try to execute it. However, before you do, why not act as if you're the computer and try to execute the program in your mind, perhaps using a scratch pad to keep track of the changes in Distance. If you look carefully at the program, at least one thing should become apparent to you: Distance, the key to the termination of the loop, *never changes*! That is, the value Distance contains upon entering the loop is the value it will hold no matter how many loop cycles ensue. The only way to change the value of Distance is to assign

a new one to it, and there are no assignment statements in Indy500a.

Look even more closely, and you'll see that we can't even say what value Distance has when it enters the loop, since no initial value is ever assigned to the variable. All we can assume is that Distance contains some "random garbage" value. If that value is not exactly 500, then the loop will continue forever, and the same number (whatever it may be) will be displayed endlessly on the console screen. In the unlikely event that the random value contained in Distance starts out being 500, then the loop will cycle once and quit. In summary, the Indy-500a program, as shown, very probably contains a so-called endless loop.

To avoid the endless loop behavior, we must add at least two more elements to the program: (1) an assignment statement—to be executed *before* the loop—that gives Distance some appropriate initial value; (2) another assignment statement *within the loop body* that modifies Distance in such a way that it eventually will equal 500, thus causing termination of the loop.

The two important questions raised by these requirements are (1) what is an appropriate initial value for Distance and (2) where and how should Distance be modified, so that looping will eventually end? For answers, let's return to the race analogy. It makes sense that any statements executed before the loop correspond to the time before the race itself. That is, the car has traveled no distance whatsoever. Distance traveled before the loop is entered is therefore 0, which appears to be the correct choice for Distance's initial value.

Since we want to chart the racer's progress mile by mile, it seems that we wish to increment Distance by exactly one for each loop iteration. The assignment statement that does this is:

Distance := Distance + 1;

In plain English, this assignment means "Distance gets Distance + 1," or, "Put the current value contained in Distance, plus one, back into Distance." As for placement within the body, the assignment may either precede or follow the WriteLn statement. If the assignment statement precedes the WriteLn, the numbers displayed on your screen will range from 1 to 500. If it follows the WriteLn, you'll see the numbers 0 to 499 displayed. Do you understand why this is so?

Exercise. Try executing the program in your mind for both possible placements of the assignment statement (before and after the WriteLn). Assume EndOfRace is equal to 5, just to keep things simple. This experience should provide the insight you need to grasp the reason why one placement of the assignment results in display of the numbers 1 to EndOfRace, while the other causes display of 0 to EndOfRace−1.

Personally, I think it is more in keeping with the spirit of the problem to show the numbers 1 to 500, rather than 0 to 499, so in my final version of the Indy500a program, which follows, the assignment that increments Distance precedes the WriteLn.

```
PROGRAM
 Indy500a;

CONST
 EndOfRace= 500;

VAR
 Distance
  :Integer;
BEGIN
 Distance := 0;
 REPEAT
  Distance := Distance + 1;
  WriteLn(Distance);


 UNTIL Distance = EndOfRace;
END.
```

The time has come for you to compile and execute Indy500. You might also want to experiment with alternate versions that use different values of EndOfRace.

Next time, we'll rewrite Indy500 to use both a WHILE loop and a FOR loop, learn about Pascal's concept of a compound statement, and start exploring *decision.* ■

THE THIRD BASIC

by Taylor Pohlman

Welcome to a series of articles on Apple III Business Basic, the powerful new cousin to Applesoft, the extended Basic that many of you know and love on the Apple II.

My goal in this series is to make Business Basic a useful, familiar tool for you. To do this, I'll pass along ideas that will help make the task of creating applications programs simpler and more efficient. Because Business Basic and the Apple III itself are new to many of you, we'll relate programming hints and techniques for Business Basic to the more familiar environment of Applesoft. To get the most out of this series, you should be fairly familiar with Basic language commands and keywords and be able to create simple programs. Without those skills as a starting point, this series would quickly grow into the equivalent of a serialization of *War and Peace*.

If you are not that familiar with Basic, your best bet is to start with the *Applesoft Tutorial* manual. If you have an Apple III, simply boot the emulation mode disk, select the Applesoft option, and insert the DOS 3.3 Master Diskette. Presto, you are now in Applesoft and can follow the *Tutorial*'s instructions to get up to speed in Basic. Once you are familiar with Basic and its syntax (a word you are guaranteed to encounter in learning the language), you'll be ready to rip through these articles.

If you are already familiar with Applesoft or another Basic, you should be ready to dig right in to Business Basic. The series will assume that you have an Apple III in front of you to try out all the things we'll discuss. For those of you in that fortunate position, the fun is just starting.

Many of you have an Apple II and are wondering if you need a III for that big new application or as an office complement to your Apple II at home. For you, this series should reveal the power of the Apple III and its relationship to the II. Hopefully, that will help you make your decision. Others of you will just be wondering what all the fuss is about, and for you we wish happy reading. No matter what your situation, you should be able to gain an understanding of the power of Business Basic and pick up some hints you can use in programming.

In any case, we welcome your comments, suggestions, gripes, or whatever concerning this column and Business Basic in general. If you've written interesting routines you'd like to share, have converted programs from another variety of Basic, or simply would like to do a core dump about your favorite subject, write to me in care of *Softalk*. Items of general interest will find their way into these pages, ensuring immortality for both of us.

One last comment should be made, especially to those who aren't business programmers. Why is Business Basic named Business Basic? As any product manager will tell you, dreaming up a product name ranks with dodging trolley cars and escaping from Alcatraz on the all-time "must do" list. Thus it was with Business Basic. Certainly it's true that scientists, engineers, educators, hobbyists, and lots more of you who are writing nonbusiness applications will find just what you need in Business Basic. As you stick with us in this and coming articles, however, you'll see that many of Business Basic's most powerful features were specifically designed to meet the needs of business applications and permit the easy conversion of programs written in other business-oriented Basic dialects.

One of the other things we'll do along our way is to show how syntax in some of these other Basics can be translated to Business Basic. This will help you use the many reference and tutorial manuals on the market that use examples from other versions of Basic. We'll include tips on converting from Basic dialects found on minicomputers and mainframes.

Well, so much for preparation. Now let's get a look at this dragon we're about to slay.

Setting the Stage. Like any other sophisticated computer system, the Apple III takes a layered approach to the operating system, languages, and utilities that animate its hardware. The term *layered* refers to the several levels of software that insulate users from needing to know exact details of the hardware on which their programs are running.

Apple III's operating system is known as SOS (pronounced "sauce"), which stands for Sophisticated Operating System. The origin of the name is curious. Several years ago in the development of the Apple III, the project was given the code name of Sara, named after the daughter of one of its inventors. Thus SOS originally stood for "Sara Operating System." When the time came to make it an official product, the name SOS stuck, so the marketing department had to come up with another word starting with S that made sense. That's how Apple III's operating system became "Sophisticated." As we explore more of SOS's capabilities, we hope you'll agree that it deserves the name.

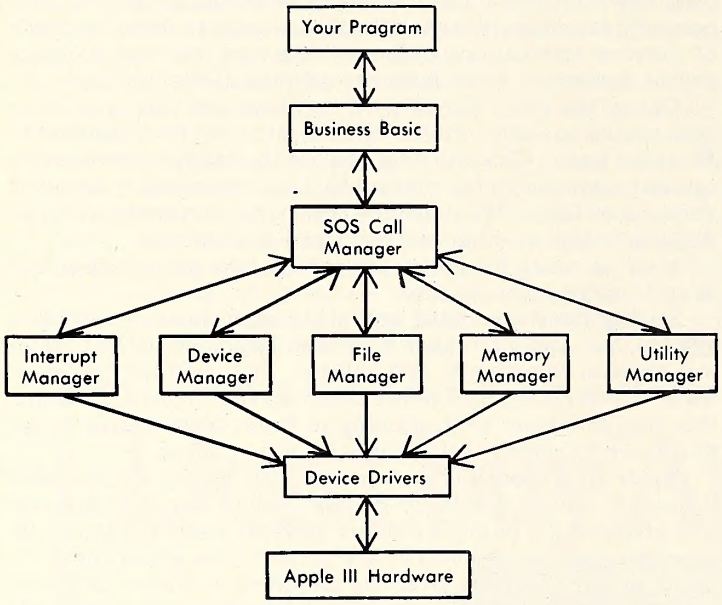
SOS's layered approach to system control makes it more than just a disk manager (like DOS) or an I/O convention (as are IN# and PR#). SOS truly manages all of the Apple III system resources to simplify a programmer's life.

In Apple III's SOS, the lowest level of software is the hardware driver. The term *driver* may seem strange, but it's very logical. Just as the driver of a car has to know the operational details of what's being driven, so the Apple III drivers need specific information about how the device is connected, what its features are, how it's controlled, and what information must pass back and forth between the device and the next highest level in the system. The beauty of this scheme is that the driver can be known by some generic name (like ".PRINTER" or ".TCLOCK") so that the operating system and Business Basic can use the device without being concerned about all the specific information that the driver must know. For example, you don't need to know anything about transmissions and turn signals to take a cab across Manhattan (a paid-up insurance policy will suffice). To extend the metaphor even further, you don't even have to know what taxi company to use; they all work pretty much the same.

In the same way, SOS can reference a ".PRINTER" for you, which may be a Centronics, a Silentype, an Epson, a Qume, or any of numerous other printers, connected via parallel, joystick, or serial ports. The higher you get in the operating system layers, the less specific you must be about the resources you use since SOS knows about all the devices you've configured on your system. Facilities are also provided to allow managing devices on a demand basis (that is, when they signal that they want to do something, called an *interrupt*). This feature makes it possible to request that more than one device be active at a time. To do that on the Apple II takes some pretty sophisticated programming.

Because of SOS's structured, layered nature, activities like reading from a remote computer while writing a message to disk and printing out a report become almost trivial.

We'll look at more about that later. It's sufficient for now that your program runs in Basic, which runs on SOS, which controls the hardware drivers, which accomplish the input/output to receive and deliver data for the system's devices (including a device called ".CONSOLE," which is the keyboard and screen). The structure looks something like this:



As you can see, each layer depends on the one below for services. Since the way the layers communicate is standardized on the Apple III, it's possible to make substantial changes to the hardware and even to some parts of the operating system without changing the way Business Basic operates. This ensures that your programs will continue to work, even if we make changes later. Designing operating systems this way takes longer and makes them larger, but, in the long run, the benefits are enormous.

Getting Started. Since booting a disk is worth a thousand "you're gonna love its," let's get started by trying some things out. Just put the Business Basic disk in the built-in drive and press reset while holding down the control key (called "control-reset" from here on).

The first thing you may see is a slight flicker as the onboard diagnostics check out the Apple III circuitry. Next is the SOS display screen, which indicates that the operating system has been loaded into memory. SOS's next task is to load the language from the boot disk. Since this is the Business Basic disk, that language is loaded and the hello program is automatically run (just like DOS on the Apple II).

You'll note that the final thing to appear is the right parenthesis ")". This is the Basic prompt, meaning that Business Basic is ready for a command.

At this point, enjoy yourself for a minute by typing:

)PRINT FRE

You 128K Apple III owners will notice that you've got more than 70K of user space for programs and data. We'll find some fun things to do with all that room later. The line shown also illustrates another convention we'll be using throughout these articles. What you type will always be underlined to distinguish your commands from what the computer outputs to you.

There are several items of interest in the display of the catalog. First, in the upper lefthand corner of the printout is the name Basic. This can vary from disk to disk and is called the *volume name.* SOS identifies the diskette you're referencing by a scheme called the *Pathname*. The highest level of the Pathname is the volume name, with any subdirectories mentioned next and the actual file name last (lowest) in the hierarchy. More on the subject appears in the *Apple III Owner's Guide* and *Business Basic* manual under "Pathnames."

The next thing to notice is the column on file type. The type SYSTEM is obvious; that's SOS and Basic, the system software. Notice that Basic is named "SOS INTERP," because on this diskette, it is the interpreter (control program) currently configured to run on SOS. Notice also that the "BLKS" column shows the space occupied on the disk in *blocks.* There are 512 bytes in each block. The next columns, alas, alack, are only relevant to those of you who have working clock chips. The files in the Business Basic disk directory will be marked with the date and time of their origin, but, without a system clock, the files you create will not. The final column, EOF, lists the exact number of bytes occupied by the file.

Now back to the TYPE column for a minute. It's easy to figure out that file type BASIC stands for a Basic program (like TIMESET). What does PASCOD stand for? Right, it's a Pascal code file, in this particular case created by the Pascal system's assembler. As you might have guessed (if you've been reading your Basic manual), the INV suffix on those files is a way of indicating that these files are set up as Basic *Invokable Modules.* We'll explore these in more detail later, but for now just remember that Basic uses assembly language routines through a mechanism called *Invoke and Perform.* There are some definite rules to follow in setting up these modules, which we won't go into now. However, there's no reason why we can't start using these capabilities right away! Hang on for a short exercise in using the SOS file system, and we'll give READCRT. INV a workout.

To get a glimpse of how Business Basic works with SOS to manage system resources through files, let's take a simple example that doesn't require the disk or a printer. Basic tells SOS that it wants to use a file by means of the OPEN command and assigns a number for later reference to the file. On the Apple III, of course, everything is treated as a file, even the keyboard and display. As we said earlier, the keyboard/display device is referred to as .CONSOLE. Note that the names for all character devices—devices that transmit one character at a time—start with a period. Type in the following so we can experiment (as Dr. Frankenstein said to Igor):

)10 OPEN#1, ". console" (This sets up a file number for Basic to use in communicating to the console.)

Note that you're already communicating to and from the console. That's because the console is the "default" I/O device. Statement 10 establishes a second path by which to communicate to the same device.

)20 INPUT a$ (This is the good old ordinary input to the default input device.)

)30 PRINT a$ (Again, default output device is the screen.)

)40 PRINT#1; a$ (Now we print to the screen again, this time through the console file previously opened.)

)50 INPUT#1; a$ (This time we input from the keyboard, using the console file.)

)60 PRINT a$ (Print to default screen.)

)70 PRINT#1; a$ (Print the same quantity to the console file.)

)80 END

Now if you LIST and RUN the result, it should look something like this:

```
)list
10  OPEN#1,".console"
20  INPUT o$
30  PRINT a$
40  PRINT#1;a$
50  INPUT#1;o$
60  PRINT o$
70  PRINT#1;o$
80  END

)run
?hello default console
hello default console
hello default console
hello console os o filehello console os o file
hello console as o file
```

A couple of interesting things are apparent here. First, although the first three lines work exactly as you would expect, the next three lines of output are a little different. The default console prints the question mark, as it should, but on line 4 of the output there is no question mark or prompt for input at all. This is because SOS is treating the console as a general input file and therefore can't know that it can accept characters printed to it. It just does a read to the device and waits for an end of record character (in this case a carriage return). The second unusual thing is also on line 4—the PRINT command in statement 60 prints right at the end of the input string (unlike line 2). The same reason applies since the carriage return you typed and the subsequent line feed the system generates for the default console are suppressed for an input file device. But line 5 is printed separately, since the PRINT command in statement 60 outputs a carriage return and line feed.

In this same way, every device connected to the Apple III is available as a file. The ability to address the console devices separately will come in handy in some future articles.

Having experimented a little with files, let's use one of those invokable modules we mentioned earlier and the OPEN statement to do something useful. This is a handy utility to use to make printouts of the screen when something strange or wonderful happens.

In this example, I'm assuming that your printer is a Silentype. Since SOS doesn't care what device it writes to, you may substitute any output file name in line 100, even a disk text file.

```
)new
)100 OPEN#1,".silentype"
)110 INVOKE"reodcrt.inv"
)120 FOR vertical=1 to 23
)130 VPOS=vertical
)140 FOR horizontal=1 to 80
)150 HPOS=horizontal
)160 PERFORM readc('value%)
)170 PRINT#1;CHR$(value%);
)180 NEXT horizontol
)190 PRINT#1
)200 NEXT vertical
)1000 VPOS=23:HPOS=1
)1010 CLOSE
)1020 END
```

Listing this program should show:

```
100   OPEN#1,".silentype"
110   INVOKE"reodcrt.inv"
120   FOR verticol=1 TO 23
130     VPOS=vertical
140     FOR horizontol=1 to 80
150       HPOS=horizontol
160       PERFORM reodc((a volue%)
170       PRINT#1;CHR$(volue%);
180       NEXT horizontol
190     PRINT#1
200     NEXT verticol
1000  VPOS=23:HPOS=1
1010  END
```

Notice that this reveals another nice feature of Business Basic: it automatically indents FOR-NEXT loops for clarity. Ever been jealous of those pretty Pascal listings? Business Basic to the rescue!

On a more serious note, let's look at what this program does. After OPENing the appropriate file in line 100, Basic is told to INVOKE the file readcrt. inv. Readcrt. inv is an assembly language routine that looks at the current position of the cursor. The cursor position is defined by the current values of the Basic reserved variables HPOS and VPOS. Readcrt. inv then modifies the value of the variable "value%" to contain the decimal value of the ASCII character at that location. The INVOKE command tells Business Basic to find a place for readcrt.inv in memory and to set up a table of all its PERFORMable routines. You can INVOKE any number of modules, and Basic will always ensure that they are located in noninterfering areas of memory.

Line 120 sets up a loop that will scan the vertical lines of the screen. Line 140 sets up the inner loop which will look at every horizontal character position on that line. The routine in readcrt. inv is then called using the PERFORM command. Isn't this easier than a bunch of pokes and a call? Line 170 prints the character equivalent to file one, our output file, and then takes a look at the next position. Line 190 makes sure we print a carriage return at the end of each output line (since that character isn't physically on the screen). After that, line 200 starts scanning the next line. Lines 1000 through 1020 set the cursor at the screen bottom, close the output file, and end.

Now for the fun. Run this program and you'll get an exact copy of the first twenty-three lines of the screen on your output file. By putting in an INPUT statement to ask for the file name and then OPENing the resultant string variable as the file name in line 100, you can decide at the time you run where you want the copy to go. Use this program to document all the strange and wonderful things you find in Business Basic as you really begin to explore the language. But first, be sure to save the program to an initialized diskette!

Well, that's it for now. Until next time, happy coding with the most powerful BASIC around.

BEGINNERS' CORNER

BY CRAIG STINSON

Last month, when we were discussing the internal layout of the Apple, we talked about two kinds of memory—read-only memory (ROM) and random access memory (RAM). The first of these, ROM, stores permanent instructions essential to the operation of the computer, material that the 6502 must have access to whenever the power is on. RAM, on the other hand, is a working area for programs and the data they use and generate. The contents of RAM are lost as soon as the power is turned off.

Memories Are Made of This. What about permanent storage of material that isn't essential to the operation of the computer but is nonetheless vital to you—things like your payroll records, a document you create on a text processor, or perhaps a game you program? That's what the disk drive is for.

The disk is a form of external, or auxiliary, memory. There are actually several kinds of external memory devices that can be attached to the Apple. The five-inch floppy disk drive is the one that most Apple owners use, so we'll consider it first and then mention some of the alternatives.

The five-inch floppy disk—sometimes called a diskette or a mini-floppy—is so named to distinguish it from the rigid, or hard, disk typically used in larger computer systems. Hard disks are made of metal; floppies are mylar, coated with a magnetic oxide.

A computer's memory is essentially a big lineup of on-off switches. In one position, a switch means zero to the computer, and in the other it means one. With the help of various codes, the computer translates data and progams into zeros and ones and stores them in its memory.

Because the molecules of the oxide coating on a floppy disk are like little bar magnets that can be oriented in either of two directions, they can be used to represent data to the computer. A disk maintains its pattern of magnetic orientations—its constellation of zeros and ones—until it's brought into contact with a magnetic field, so it's suitable as a permanent storage device—presuming, of course, that you don't set it on your television set or wave a magnet over it.

The disk drive—the one sold by Apple Computer is officially called the *Disk II*—is an example of an input/output, or *I/O*, device. A drive that's physically hooked up to the computer is said to be *on-line*; likewise, data on a disk in such a drive is also on-line. By contrast, a disk sitting on a shelf somewhere is said to be off-line.

Program-to-Program Service. The floppy disk bears some resemblance to a phonograph record. When the drive is turned on, it spins the disk at a certain precise speed, much the way a turntable spins a record. A read or write head then moves along the surface of the disk until it finds the appropriate place to read or inscribe.

GLOSSARY

Basics—a disk that gives DOS 3.3 owners access to files created under DOS 3.2.
boot—short for bootstrap.
bootstrap—to load the first few instructions of a routine, from hardware. In Apple context, the term is usually associated with DOS; when you type PR#6 or turn on the machine with a disk in the drive, you are booting DOS. This operation is also commonly and erroneously called booting the disk. The disk in drive one when DOS is booted is called the boot disk.
DOS—acronym for Disk Operating System. DOS tells the computer how to use the disk drive.
FID—a program on the System Master disk that, among other things, transfers DOS 3.3 files from one disk to another.
greeting program—the program with which you initialize a disk. When you boot DOS, one of the first things the computer does is execute the greeting program on the disk in drive one.
Hello—the conventional name for greeting programs on the Apple.
initialize—a term that means different things in different contexts. One meaning is to prepare a floppy disk to receive data by implanting DOS and laying out the track and sector structure.
input/output device—an instrument external to the computer that provides data to and receives it from the computer.
Master Create—a program on the System Master disk that converts a slave disk to a master disk.
master disk—a disk whose DOS is not dependent upon the computer's memory size.
Muffin—a program on the System Master that takes information from a DOS 3.2 disk and reformats it so it can be saved on a DOS 3.3 disk.
off-line—a term describing equipment or data that is not directly connected to the computer.
on-line—a term describing equipment or data that is hooked up to the computer.
revision level—a number that tells the user which version of a program is being used. The 3.3 in DOS 3.3 is the operating system's revision level.
sector—a unit of data on the floppy disk. Disks initialized under DOS 3.3 are formatted into 560 sectors, each of which holds 256 bytes.
slave disk—disk whose DOS is memory-size dependent. A slave disk initialized on an Apple with 48K will not boot on a 32K machine.
track—one of the concentric rings into which DOS formats the floppy disk. Each track on a DOS 3.3 disk is divided into sixteen sectors.

Like a phonograph record, the disk is a random access device. This means you can get to any part of it directly without having to take in everything in sequence from one end or the other. As such, a disk offers certain advantages over sequential access devices like magnetic tape or paper tape.

That's about as far as the similarity goes. Information on a record is stored along a continuous groove. The floppy disk, in contrast, organizes information into discrete concentric rings, called *tracks*. Each ring is broken up into discrete portions called *sectors*. Each sector will hold a packet of data—256 bytes on disks used by the Apple. And the data does not usually flow continuously from one sector to its physical neighbor.

None of this organization of the disk into tracks and sectors is inherent in the physical structure of the disk. A brand-new blank disk is just oxide-coated mylar in a cardboard container. All kinds of schemes could be devised for organizing data on the disk, and, unfortunately, different kinds of computers do indeed use different formatting schemes. This is one reason why your new Apple game isn't likely to run on your friend's Brand X computer.

Teaching Your Apple To Say Hi. Before you can do anything with a blank disk on the Apple, the disk must be initialized. Here's one way to do this. First, turn on your machine with the System Master disk in the drive (drive one if you have two or more drives). Then type NEW to clear the program area of RAM (typing FP or INT will accomplish the same thing). At this point, you'll need to key in a short program in whatever brand of Basic you're working in. This program will be known henceforth as the greeting program for the disk you're about to initialize.

Greeting program is just a generic term to describe the program with which you initialize a disk. The actual name of the program—the one that will subsequently appear in the disk's catalog—can be anything you choose. It's customary to name the program Hello, but you can call it George if you like. Or Zdravstvitye. Any name that starts with a letter, has no more than thirty characters, and has no commas is legal.

Because so many Apple owners name their greeting programs Hello, you'll often encounter the term *hello program*, used as a synonym for greeting program.

You don't have to know anything about programming to write a greeting program. You can type in something as simple as 10 PRINT "THIS DISK BELONGS TO GEORGE." Or just 10 PRINT. The program doesn't need to do anything special; the Apple just has to have some kind of program to put on the disk when you initialize it. Page 14 of Apple's current *DOS Manual* (the one with the orange cover illustration is current) will tell you more about how to write a greeting program.

After you've got your program, replace the master in the disk drive with a blank disk. Then type INIT HELLO, if you're going to call the program Hello. If you want to give it some other name, type INIT followed by the name you choose. When you hit return, the red light will come on, the drive will emit some pleasant purring noises, and in thirty seconds or so you'll have a new disk to work with.

The Apple Adds Its Monogram. Two things happen when you type INIT. First, the computer formats the disk into the track and sector structure we described. It does this by writing information at the appropriate places to identify each sector on the disk. If the disk was not blank when you typed INIT, whatever information was there is erased during the formatting process. Second, the computer fills two entire tracks with a set of instructions known as DOS.

DOS stands for Disk Operating System. Simply stated, DOS is the information the 6502 requires to perform any operation involving the disk. When you type CATALOG, for example, you are invoking a DOS command.

The Disk Operating System also governs the structure of data on the disk. DOS was introduced by Apple in 1978 and has undergone several revisions since then. The most recent version, called DOS 3.3, was released in August 1980.

The major difference between earlier versions of DOS and DOS 3.3 is that the latter divides a track into sixteen sectors and the former only thirteen. This is a significant enhance-

ment since it permits you to put almost 25 percent more data on a disk. However, data that was stored on disk using DOS 3.2 cannot be read by an Apple that's expecting to find sixteen sectors on a track.

The Ins and Outs of Different DOSes. Fortunately, there's a way around this problem. If you have DOS 3.3, you should have, in addition to your System Master, a disk called Basics. If you want to read a thirteen-sector disk, all you need to do is put Basics in the drive and turn the machine on (or type PR#6 and return, if your machine is already on). By doing this, you tell the computer that you want to work temporarily in the older operating system. A message will appear on your screen, telling you to insert your thirteen-sector disk and hit return. Then, until you turn your machine off and on again, or type PR#6, you'll be operating under DOS 3.2.

Doing Away with the Slave System. Here's a postscript about initializing disks. When you type INIT, as we said, one of the things that happens is that the set of codes called DOS gets written onto two tracks of your new disk. Included with the operating system is information concerning the amount of random access memory in your computer. Your machine knows whether it has 32K, 48K, or whatever, and it plants that information onto your new disk right along with the rest of DOS.

The significance of this fact for you is that a disk created on, say, a 32K Apple will henceforth only be useful to an Apple with at least 32K. Your disk is at this point officially termed a *slave* disk. You can convert it to a *master* disk—which will then run on an Apple with any amount of memory—by means of a program on the System Master called *Master Create*. Pages 44 through 46 of the *DOS Manual* tell how to do this.

The terms *slave disk* and *master disk* are a little misleading, perhaps—especially since they can be confused with System Master, which is the name of a particular disk—but we seem to be stuck with them. If you have an Apple with less than 48K of RAM and you think you might at some point add more memory, it would be a good idea to use the *Master Create* program whenever you initialize a disk. The reason for doing this is that, even though it will run on a machine with more memory than the one on which it was initialized, a slave disk will cause the larger machine to ignore its extra memory. In other words, if you make a slave disk on a 32K Apple and then use it on a 48K Apple, everything will work fine, but the 48K machine will behave, for the moment, as though it only had 32K.

On Boots and Straps and Such. When you turn on the Apple or when you type PR#6 return, one of the first things that happens after the disk drive turns on is that the computer reads those two tracks containing DOS. All the information that makes up DOS gets loaded into a certain area of random access memory. Thenceforth, when you type a DOS command like CATALOG, RENAME, or DELETE, the computer will know what to do. It won't have to go to the disk to find out what your command means. In fact, since DOS contains all the information the computer needs to interact with the disk, if DOS isn't already stored in random access memory, the computer won't even know how to go to the disk.

This raises an interesting question. How does the computer know how to get DOS off the disk in the first place? A bit of coding in a read-only memory chip on the controller card tells it how to do that. The procedure—reading DOS from the disk into random access memory—is called *booting* DOS, or *booting the disk*. You can do it either by turning on the machine with a disk in the drive or by typing PR#6 (assuming your controller card is in slot 6) and hitting return.

Boot is short for *bootstrap*, a common computer term that's roughly equivalent to "starting from nothing." A bootstrap loader is a bit of hardware that contains the first few instructions of a routine, which are then used to load the rest of the routine from somewhere else. The read-only memory on the controller card is an example of a bootstrap loader.

A Thrifty Alternative. Now then, about those alternatives to the floppy disk. A less expensive—and less convenient—form of external storage is magnetic tape. The Apple has a pair of jacks, next to the video jack, into which you can plug a cassette tape recorder. You can then save data on tape in much the same manner as you would record music or speech.

The big advantage of the tape over disk is cost. You don't need a high-quality recorder or high-quality tape, and you don't need to plug in any $500 interface cards.

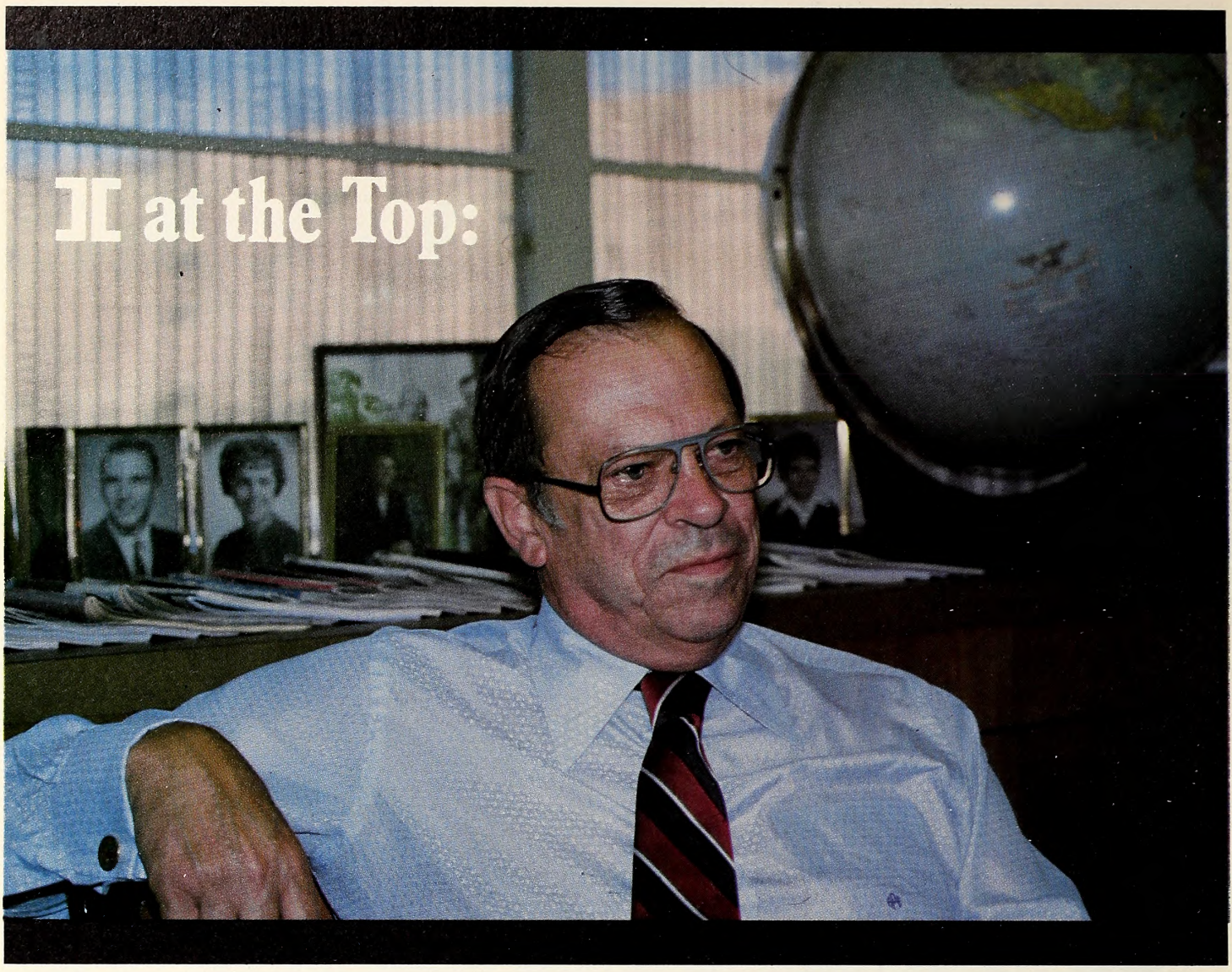
The disadvantages are several. Sending data to the recorder and loading it back into the computer are relatively slow and clumsy operations, compared to doing the same things with a disk drive. For one thing, you have to set the volume and tone controls to just the right levels for the data to be transmitted properly. Furthermore, since tape is a sequential access device, if you have five programs on a cassette and you want to load the third, you've got to advance to the right spot yourself; you can't just tell the Apple to go get Number Three.

As if these limitations were not enough, there is more to the price of not having that expensive card in slot 6. You're very restricted in the ways you can interact with the recorder during the running of a program. Certain kinds of data can be stored or retrieved while a program is running, but other kinds cannot.

Nevertheless, you may occasionally have reason to use tape even if you do have a disk drive. The medium itself is less expensive than disk, which means that byte for byte it's a cheaper form of storage. Tape can be a practical way to archive extensive data files that you're not planning to use again for a while. When you do need to look at that data again, you can load it into the computer and save it back out onto a floppy disk for convenience.

An Efficient Expense. So much for tape. A high-cost alternative to the floppy disk is a rigid disk system. The most obvious advantage to this approach is that it allows you to keep a great deal more information on-line at any given time. If you need to sort or rearrange something like a large mailing list or if you have several Apples needing to use the same information, a hard disk system may be a reasonable investment. Rigid disks also have advantages over floppies in terms of speed and security (see the article on page 88). ■


ODYSSEY:
The Compleat Apventure
Synergistic Software
5221 120th Ave. S.E.
Bellevue, Washington 98006
A colorful high-res graphics adventure on land and at sea, in dungeons and castles; different each time you play. Plan your strategy, solve the puzzles, and survive the endless variety of hazards and foes to reap your rewards. Odyssey is an ever fresh mythic adventure.
Apple II, DOS 3.2 or 3.3, Integer — $30.00.



The Chairman and His Apple

BY MELISSA MILICH

Dr. Allen E. Puckett, chairman of the board and chief executive officer of Hughes Aircraft, sat back with his arms folded as his twelve-year-old son took control of the jet fighter plane.

"I'm taxiing out on the runway," Jimmy Puckett said excitedly. "My vertical altimeter's building up."

"Aren't you supposed to get on the center of the line when you're taking off?" Puckett senior asked, mildly concerned.

"We're up. Don't worry, but now I can't lose any altitude. This button is to climb, this button is to land, and this button is to declare war."

"Don't push that one."

"I won't do that till the last minute. Oh crud, the enemy aircraft is firing at me!!!"

The top executive from the defense aircraft plant, one of the chief suppliers to the Pentagon, remained unruffled. "You can run away from him, can't you?"

But the plane had already gone into a nose dive and crashed. Puckett senior shrugged his shoulders and chuckled. "Well, Jimmy, you're not ready for the real thing yet."

Coming Home to Apple. Although Allen Puckett probably could identify with some aspects of SubLogic's computer game, *Flight Simulator,* not all the problems at southern California based Hughes Aircraft are so easy to deal with. Jimmy has a lot more leeway with his computer game—he can even make his airplane fly backward if needed. Unfortunately, life at Hughes Aircraft, with more than fifty-five thousand employees working with billions of dollars in government defense contracts, is a bit more structured.

As the top executive officer of Hughes, Puckett has to do a lot to keep the company functioning. His days are filled with meetings, giving reports, analyzing material. At the end of the day, he escapes to his Pacific Palisades, California, home, takes off his jacket, and relaxes in front of his Apple.

The microcomputer serves a dual purpose in Puckett's life. Not only can he unwind by playing games with his son, he is also keeping up with business. Mainframes play a major role

Softolk photos

at Hughes, and Puckett says he wants to understand the massive amount of computer power in the company.

According to Puckett, the Apple is not greatly different from the big machines. "Understanding the flow of information is the important thing."

Experts Led the Way—Now He Does. Puckett says he wanted his own microcomputer at home so he could take time in the evening to absorb the processes, since he was always too busy at work.

Before he made his decision on which microcomputer to buy, Puckett checked with a number of professional programmers who he reports were "five to one in favor of the Apple."

That was two years ago, and he doesn't regret his decision. On the contrary. "Dr. Puckett loves his Apple," confides his private secretary, Julie Maccabee.

Puckett is no stranger to the computer world. With his engineering background, he was first introduced to what he fondly recalls as "a mechanical monster" desktop calculator in the late 1930s. "It was pretty crude and pretty clumsy. It just cranked away all day, multiplying and dividing." Yet it was a forerunner of the computer.

From monster to micro, things have come a long way. "The Apple is self-contained, easy to use, has more than enough power, and is not limited in speed or memory," Puckett says with executive authority.

His being so outspokenly in favor of the Apple has naturally convinced some of Puckett's colleagues also to learn at home. He made a convert just recently, as a matter of fact. "Let's just say he [the new Apple owner] got tired of hearing me talk about it," he says a bit sheepishly.

"But it always tickles me to think of somebody working all day on an AMDAHL or an IBM 3341 and then going home and playing with the Apple."

Joy of Programming Wins Over Games. In spite of his stint with *Fight Simulator,* Puckett says he does not play many games. "I get trapped into playing them once in a while, but I prefer to mess around with my own programming."

Puckett is a self-taught programmer and started with the manual that came with his Apple computer. He has since devised his own programs to keep track of his private tax records and stock transactions as well as his own mailing list.

The latter is a simple program, says Puckett, suited to his own purposes, with address classifications and affiliation codes. At one point, he had the computer print out his Christmas card mailing labels, but then executive Puckett decided that would be too impersonal and dumped them.

Puckett is partial to the program he developed to keep track of the yachts in the Mazatlan Regatta (See *Softalk,* April 1981). The program monitors the times and standings for the entrants in the yearly two-week race from Los Angeles to Mazatlan. Last November, the program was run in an Apple computer aboard a boat piloted by Donald Gumpertz, chief escort and communications post official for the regatta. Gumpertz, who's a friend of Puckett's, broadcast the results day-by-day to keep the contestants apprised of their position in the long race.

Puckett, who entered his own yacht, *Blackbird,* in the regatta, says keeping track of your own and your competitor's boats makes for a more interesting race. "It keeps up the morale, especially if you're doing well," he says. "Otherwise,

THE DAKIN5 DIFFERENCE

Proven Performance with 7,000 **Controller®** Packages now being used by satisfied customers.

Specially designed for users who have no prior computer or accounting knowledge.

Fast, technical support for you through nation wide, toll-free lines to our Customer Service Department.

Simple, all programs are menu driven with built in fail safe protection that allows error free operations.

A total system approach with add-on packages that allow you to meet all your business needs.

There is other business application software on the market all claiming it is the best. But, only the Dakin5 Controller Package has recently won the International Computer Programs (ICP) Award for one million dollars in sales the first year on the market.

The new Controller® 1.1 is a comprehensive accounting system comprised of Accounts Payable, Accounts Receivable and General Ledger. It has improved processing speed, requires less disk swapping and is printer independent. It's fail safe and error free, which makes The Controller ideal for the businessman who wants maximum output with no worry. The Controller package shortens bookkeeping time, allows for easy tracing of transactions thanks to audit trails, and warns of data entry errors via "audio beeps". The Controller also protects the user from losing important business information by automatically making a back-up copy of all operational data after posting. The Controller prints statements and checks and generates all reports necessary for the running of a successful business. The reports are so complete it's more like having an accountant 24 hours a day than a computer system. All of these features make The Controller software the best accounting package for your business.

What makes The Controller Package even better is **The Analyzer®** Package. The Controller software was developed to serve as an operational aid.

The Analyzer is a simple to operate management tool which utilizes The Controller as a data base. Little or no operator time is required and information is produced automatically that would be virtually impossible to generate manually. The Analyzer produces 16 decision making reports which allow for the comparison of business activities on a monthly, quarterly, and yearly basis. These include The Cash Flow Report which projects the amount of money available for the next 4 months, allowing for realistic decisions for the use of available cash. Because all financial information used by The Analyzer is produced by The Controller, tighter control can be exercised and immediate decisions can be made if the results are not as planned.

Dakin5 isn't stopping with these two products designed for the businessman. The following products will be available in early Fall, and will provide you with a complete integrated business system.

- **The Business Bookkeeping System™**
- **The Depreciation Planner™** Package
- **The Visualizer™** Package
- **The Budget Planner™** Package

Dakin5 is committed to helping you solve the right problems. That's the Dakin5 difference.

For more information see your local Apple Retailer or contact Dakin5 Corporation at P.O. Box 21187, Denver, CO 80221, Phone: 800 525-0463.

DAKIN5 CORPORATION
"...committed to solving the right problems."

a week goes by with no way of telling what's going on."

A modified version of Puckett's regatta program, written by Larry Somers, was used recently in the Transpac race from Los Angeles to Honolulu.

Like Ducks to Water. Puckett is not presently working on any new programs, but he's revising many of those he's already done. His son Jimmy, who'll enter sixth grade in the fall, is also learning programming. He attends a private school in Pacific Palisades where students have the option of taking a computer course. The school uses Apples, and Jimmy is learning Basic.

"Kids aren't scared of the computer because it's fundamentally pretty easy and they dive right in," notes Puckett. "A lot of adults grew up in a slide rule world and still reject them. But computers are as natural to kids as milk and cookies."

Puckett has also managed to get his wife Marilyn involved with the Apple, primarily in word processing. Their Apple sits in a corner of their living room on an oak stand that Puckett built himself.

The Hughes executive sees the microcomputer as playing a more and more important role in the home in the future, and he predicts it's going to become "as standard as a TV set."

Future Promises Quick News, Easy Chores. Although microcomputers are already playing a role in home security and home management, the "big future," according to Puckett, is the home information center tied into a central communication center with a data base covering everything from airline schedules to the Dow Jones stock report.

"This little Apple can be plugged into a telephone, and its user will be able to call up darn near everything for whatever is needed," says Puckett, sounding rather like a proud parent.

Puckett says he goes along with the experts who liken the electronic revolution of the 1980s to the Industrial Revolution.

"The Industrial Revolution was a way to extend muscle power, an extension of human energy into mechanical energy," explains Puckett. "It was the big thing in the 1800s. The computer explosion will be the big thing in the last half of the 1900s."

The electronic revolution can best be summed up by what Puckett calls the electronic triangle: data processing, data storage, and communication.

"Group all three and you have a totally new capability in excess of human capability. We're moving into an age where information flow is a big thing."

Laps Ahead in Progress Race. "Computers are changing the face of society," he continued. "This is where the action is—the most rapid advances in technology.

"I don't mean the rest of the world is standing still, they've made some tremendous leaps in areas like genetic engineering, but computer technology is growing faster than anything."

And just how far will it go? Does Puckett see a future in which businesspeople won't even have to go to the office—but will report instead to their computers at home?

Not at Hughes Aircraft, he asserts. "I wouldn't be surprised seeing the Apple showing up in the company, but a computer cannot replace the hands-on work that's necessary here.

"The team effort is fundamental at Hughes. We have to have our employees get together and talk about projects and equipment they're working on.

"Whether they agree or disagree, whatever the case, computers will never replace that human interaction."

Instead, Puckett suggests that computers will replace or speed up a considerable amount of paperwork and give individuals more time to pursue important things—including fun. Does Puckett have any advice for the brand-new microcomputer owner?

"Read the book. Of course, it depends upon what you want to do with your computer, but I say think up a simple program, even something as simple as balancing your checkbook, with it. But owners should definitely learn programming. That's the real fun." ■

THE BEST 250 DOLLAR WORD PROCESSOR ON THE MARKET . . .

SuperScribe II is the most powerful and easiest to use Word Processor available for your Apple II or II Plus computer. Besides leaping tall buildings in a single bound, it:

- Gives true upper/lower case text on your screen with no additional hardware whatsoever.
- Works with documents larger than the amount of memory in your Apple—transparently to you!
- Edit not only letters but also any text or binary file, or even basic programs!
- Automatically generates up to 4 separate indices for your document!
- Save typing time through a unique ability to designate specified keys as commonly used words, phrases or even commands!
- Globally search for or replace character strings.
- SuperScribe II has a built-in instruction capability such that if you forget how to use a command and the manual is not close by - you may simply ask SuperScribe][.
- Supports multiple disk drives!
- Will support alternate character sets.
- Supports the shift key modification if made to your Apple.
- Lets you work with your text on a screen at a time basis—reducing typos and allowing you to see your document as you edit it.
- Works with any printer!
- Proportional Spacing!
- Hyphenations made
- Supports the language card or any 16K expansion Ram card to keep more of your document readily available in memory.
- 70 column review—see your text formatted in 70 character lines before you print. Again, without additional hardware!

and it's only 129.95 . . .

SuperScribe II

by

ON-LINE systems

36575 MUOGE RANCH ROAD
COARSEGOLD, CA 93614
209-683-6858

VISA, MASTERCHARGE, CHECK OR CO

ABOUT TIME

from page 36

home. When you're in the house, you may have the Apple turn on the coffee pot at 6:30 a.m. after turning on the lights in your bedroom at 6:00 a.m. You can have your electric fans go on at 5:30 p.m. so you'll come home to a cool house. Also, you can have the lights dimmed or brightened at different times of the day and evening, at your command.

The X-10 Interface Accessory comes with *Scheduler* software and instructions for making your own appliance control schedules. The *Scheduler*, an interrupt-driven program, lets you do other things with your Apple, provided that what you want to do can be done without rebooting your disk, pressing the reset key, turning the computer off, or using interrupts.

Apple Computer is distributing Thunderclock products for use in Apple IIs and IIIs. Expect to see wider acceptance of the Thunderclock as time progresses.

Right now, the Thunderclock Plus sells for $139. The X-10 Interface Accessory, $49. A diskette that allows Pascal 1.1 users to set and read the Thunderclock and send BSR X-10 commands costs an extra $29. (As the Apple Pascal system is based on date-stamped files, the Pascal software disk can automatically set the correct date upon rebooting of a disk modified with the date-updating program.)

You can use BSR-10 Appliance Control System with or without an Apple computer. Approximate pricing is $50 for the console and $15 for each appliance or lamp controller. Many stores offer cheaper pricing on kits that usually consist of a console and more than one controller. Since about three controllers are now on the market, you must buy the one called BSR X-10 Ultrasonic Command Console for your Apple.

Superclock. The Superclock II varies from the Thunderclock in that the Superclock has no X-10 accessory option and it has only two operational modes, which aren't selectable from software. But the Superclock has more interrupt intervals than the Thunderclock.

The Superclock can keep time only to one second precision; interrupt rates may be set for 1 or 1,024 a second, or one every minute or hour. The Superclock can report the time in the Mountain Computer format or, by changing the setting of a switch, add day of the week and year information. Note that the Thunderclock and the CPS card keep day of the week information. Neither keeps track of what year it is.

West Side Electronics has a special diskette that automatically puts a date and time stamp on all programs and files you save or update on your diskettes. (Thunderware has a *DOS Dater* program that performs the same function, plus adds the time of day to the date.)

The Superclock II complete system, which includes a Pascal utility disk allowing Apple Pascal users access to the clock's functions; a Basic (mainly Applesoft) demo diskette; an interrupt demo diskette; and *Super DOS*, the date stamper, sells for $159. In summary, the Superclock system is the lowest priced of the three, counting all options, but it doesn't have any of the fancy additional functions the Thunderclock or the CPS Multifunction cards have.

West Side Electronics has a program for their Superclock called *Time Clock II*, which allows you to keep track of the amount of time your Apple spends running various programs. Example uses suggest that this would best be used in applications where it's essential to know how much time is spent on a particular task (for hourly billing, perhaps) or where keeping track of how much total use various individuals put the computer to is important.

October will bring tidings of more clocks for Apples. ■

ERS CONTEST WINNERS CONTEST WINNERS

from page 4

hopes of someday making one that would be good enough to sell.

My father is a SGT in the AIR FORCE and although he works hard he would never be able to afford to send me to a camp like this, as I have two brothers and one sister who need stuff too.

My two little brothers and a little sister almost never let me alone when I want to use our Apple. It sure would be nice to be somewhere without them. Around our house our computer is like part of the family. Camp is lots of fun but I think it would be great to go to one where I wouldn't have to miss my Apple while I was gone.

My hope for the future is to go to college and study computer science. I have a short term goal to program the BEST arcade game of ALL TIME. My long term goal is to own a computer software and hardware company. My little sister is physically handicapped and I hope that someday I can use my computer skills to design something to help handicapped children like her.

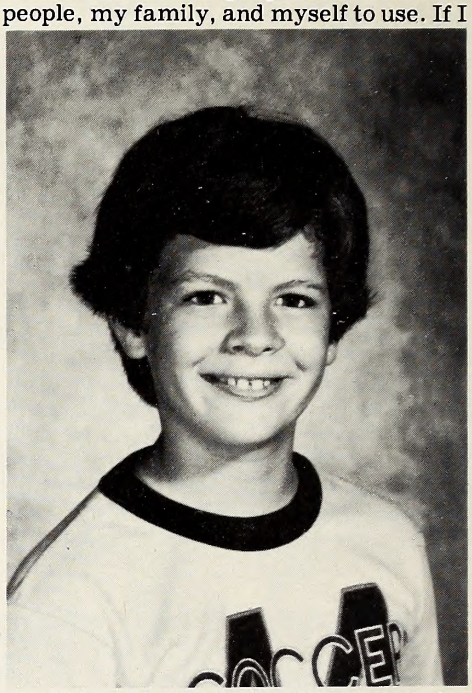
Although I have many reasons why I'd like to go to Computer Camp I think the best one is to get a slice of knowledge from the Apple. I do know I won't have to worry about getting sick, since I will be getting my Apple a Day.

John C. Brandstetter

The reasons that I want to go to computer summer camp are that I am fascinated by the capabilities of my Apple II, and I truly want to learn vastly more about the techniques of programming my Apple. At my house I have a limited learning environment because my only references are Apple II reference manuals and magazines dealing with computer programming. At a computer camp, there will be many more references to help and allow me to expand my knowledge, such as special operating manuals and guidance counselors. Also, there will be other children around my age that know somewhat how to operate and program an Apple. Under these conditions, I can help other children learn, and learn from them myself. We will be able to exchange programs and ideas with each other, and we could help each other solve difficult problems. I will also be able to make new friends with people that own or use Apple IIs.

Computer camp will allow me to merge computer programming with regular outdoor camp activities. It will allow me to have five complete hours of learning about and working with the field that I am most interested in, computers, with no interruptions, and I plan to and want to utilize every useful minute of it. I think that the knowledge that I will gain by working on the computers at summer camp will be very important to me, because right now that is the only place where I am able to gain that.

After I learn and understand how to use a computer more fully, I am going to start editing more sophisticated utility programs, such as text editing systems and financial aid programs, for other people, my family, and myself to use. If I



would be able to use all of the knowledge that I have and let it have a chance to grow, then I believe that these things and more would be possible for me in the near future.

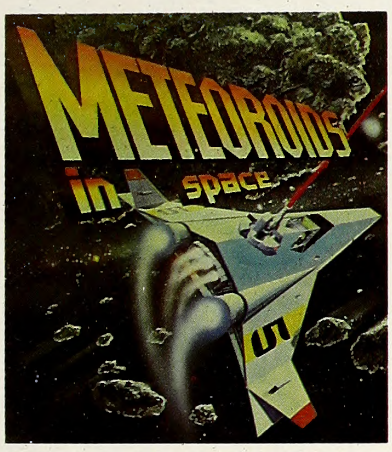
In high school and college I am planning on taking a variety of courses in the field of computers, spanning from robotics to computer engineering, and I would truly like to go on to more advanced courses, and I believe that to do that I would have to have some sort of computer-based background, which I can get best at computer camp.

I am also planning on using computers in my future occupation. I will be able to get a suitable job because of my background in college. I am planning on being a scientist. I want to be able to use computers to analyze, compare, compute, categorize, and store data.

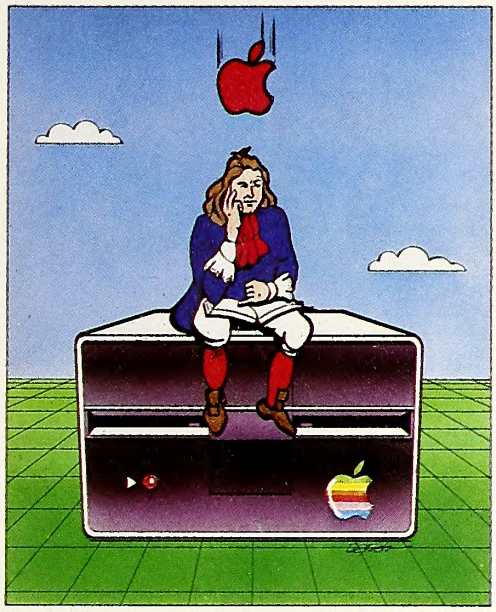
I believe that in the future, computers will have a great impact upon the lives of humans, and want to be able to use my knowledge of computers to create things that will bring benefits to many people. I am preparing myself for that period by using and wanting to learn more about the Apple.

Greg Galperin

our first name is Quality



Our popular space game, *Asteroids in Space*, is now *Meteoroids in Space*, with hyperspace, autobrake, and autofire. This exciting arcade game can be played with or without game paddles. Diskette $19.95.

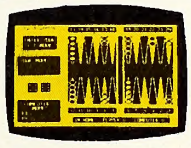


Beneath Apple DOS

This best-selling 170 page technical manual tells you everything you want to know about Apple's Disk Operating System. Book $19.95.

Beginners and experts both enjoy this Othello* playing game that outplays Hayden's *Reversal*. Yet *QS Reversi* has features that *Reversal* doesn't have and costs less. Diskette $29.95.

LINKER
Links and relocates assembled machine language programs. Works with any assembler. Diskette $49.95.

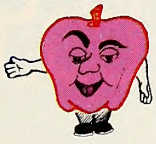


FASTGAMMON
Known for outstanding graphics and speed of play, this popular program is the standard to which other computer backgammon games are compared. Cassette $19.95. Diskette $24.95.

CR

CROSS-REF
Prints out a cross-reference of the variables in your Applesoft program. Options include changing the name of a variable. *Cross-Ref* resides passively in memory with your Applesoft program. Cassette $22.95. Diskette $24.95.

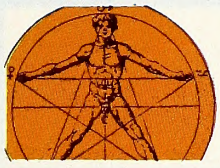
FRACAS
A fantastic adventure game like no other. Up to eight players can journey in the land of Faroph at the same time, searching for hidden treasure and fending off monsters. Cassette $19.95. Diskette $24.95.



BABBLE
This amazing and amusing program will entertain you by twisting your stories around in a most delightful way. *Babble* also works with music and sound. It's a language all its own! Cassette $19.95. Diskette $24.95.

BENEATH APPLE MANOR
The original dungeon game for the Apple computer is still one of the best. Descend into a dangerous maze of underground corridors. See if you can find the Golden Apple. Cassette $14.95. Diskette $19.95.

BATTLESHIP COMMANDER
A superb version of the traditional battleship strategy game. Cartoons and sound effects help make this game fun for everyone. Cassette $14.95. Diskette $19.95.



ASTROAPPLE
Your computer becomes your astrologer, generating horoscopes and forecasts based on the computed positions of the heavenly bodies. Cassette $14.95. Diskette $19.95.

*Othello is a trademark of Gabriel Industries.

6660 Reseda Blvd., Suite 105, Reseda, CA 91335
(213) 344-6599

QS

ASK FOR QUALITY SOFTWARE products at your favorite computer store. If necessary you may order directly from us. MasterCard and Visa cardholders may place orders by calling us at (213) 344-6599. Or mail your check or bankcard number to the address above. California residents add 6% sales tax. *Shipping Charges:* Within North America orders must include $1.50 for shipping and handling. Outside North America the charge for airmail shipping and handling is $5.00. Pay in U.S. currency.

QUALITY SOFTWARE

The Apple In Big Business

BY MELISSA MILICH

Call it a revolution, call it a radical change. Whichever way you choose to look at the introduction of microcomputers into business, a few factors can't be ignored. Cost-effective, highly portable, the amazingly adaptable Apple is doing things for big business that used to be reserved for the not-so-flexible mainframe.

Herein are profiles of five very different multi-million-dollar organizations that use the Apple for tasks customized to their products as well as for more common microcomputer uses. From processing the annual budget to exploding out the ingredient analysis of a single cookie, the Apple is ready for work.

Apple-enthusiastic executives praise the Apples, II and III, for their versatility, efficiency, and simplicity as they see employees freed from many repetitious tasks to enjoy more challenging creative pursuits. In some cases, like that of Eastman Kodak's Denny McElroy, the Apple computer made possible an undreamed-of promotion.

The businesses—Arthur Young & Company, Avery International, Bank of Louisville, Eastman Kodak, and Pepperidge Farm—are using Apples to keep up with the rapidly advancing technology of the eighties. They're representative of an astonishing number of megabusinesses that purchase Apples by the dozen or even the gross—some of whose names may ring a bell, like Ford, Proctor and Gamble, Dow Chemical, CBS, AT&T, Westinghouse, TRW, Xerox, du Pont, Owens Corning, Teledyne.

Whether corporate executives gather round it to make multi-million-dollar decisions or a family uses it solely to play computer games, it's all the same Apple, with enough power for everyone.

Avery International. "I want an Apple on the desk of every person who works for me," says Hugh Kent of Avery International, a company whose product graces many of the items you probably use daily.

Kent is controller of the Fasson Industrial Division in Paynesville, Ohio, one of Avery's thirty divisions and subsidiaries around the world. Avery is best known for its pressure-sensitive labels, which are stuck on everything from mailing packages to shampoo bottles. Financial planning and analysis at the Paynesville division has gone from "totally manual to one hundred percent computer," according to Kent, and for a *very* good reason.

The Apple has increased the output of financial reports considerably. It used to take two weeks to prepare twenty-one financial reports manually, approximately three hundred pages of copy. With the Apple, forty reports are finished in forty-eight hours. "That's a ninety percent increase in the number of reports done monthly," says Kent.

These reports are custom-designed by the users, he explained, and fill specific needs in marketing, manufacturing, and research. "Essentially, the analysts customize programs to fit their needs. If we hired four more people we couldn't turn out this volume of information."

The Avery division in Paynesville presently uses nine Apple IIs and two Apple IIIs. "And we're doing so many things with them, it's hard to know where to begin to tell about them," Kent confesses.

Apples Never Have a Headache. Probably the most important, the most complicated, and the most headache-producing task facing the organization is the annual budget. But the computers take it on without complaint.

"The Apple has allowed us to project unit product cost a full year in advance," continues Kent, "and this allows the marketing department to take responsibility for a product's future and to plan price strategies."

The Apple also gives Avery a full income statement for each strategic business unit, tracking the total capital investments process, including appropriations in building and equipment. For example, all new machines recently purchased by the company are evaluated for efficiency by the Apple. The program figures exactly how much any particular machine is costing the company per month—determining, in effect, whether the machine is earning its keep.

Since many of the Avery products are sold commercially, the company recently designed a system for the Apple to do all customer pricing. Some items are priced according to a standard list, and others have customized pricing.

Eighty percent of the Apple programs are *VisiCalc*, ten percent are DMS, and the last ten percent Kent describes as "home-grown Basic."

Kent says he'd eventually like to network all the computers into a data base manager and communicate with three different time-sharing services. "I'm one of the true enthusiasts for the Apple just because they're so versatile."

Pepperidge Farm. Milanos, Lidos, Pirouettes—who'd ever think an Apple computer would help put some of Pepperidge Farm's finest cookies in the mouths of consumers? Although the Apple still has not learned to cook, it is helping Pepperidge Farm's quality bakers in a variety of jobs from marketing to ingredient analysis.

"Basically what we did was put an Apple in the hands of people who know their jobs in just about every area of the business," says Pete Zezima, manager of manufacturing computer services.

Zezima teaches several different classes in computing for Pepperidge Farm employees, ranging from beginning *VisiCalc* to word processing. It took a very short time for their employees to master the various applications for the micro. Once the exposure began to take, Zezima said, interest in the classes took off "like a landslide."

Zezima emphasized that it's his coworkers who've made the microcomputer fit in so well to life at Pepperidge Farm. About one hundred sixty Pepperidge Farm employees are proficient on the forty Apples there.

Pepperidge Farm is a team-run company, according to Zezima. The processes that get their baked or frozen goods onto grocery shelves are cyclic, and the Apple has its finger in every pie.

Yum. The Apple analyzes sales, helps determine a forecast, and comes up with production schedules with distribution analysis. For example, according to the Apple, chocolate sells best in California, while easterners prefer products made with coconut.

Sound easy? Zezima doesn't think the process would work as efficiently at Pepperidge Farm if the employees hadn't adapted so well to the computers.

"This is a people-oriented business," says Zezima. "Microcomputers save an incredible amount of time. They're not being used to replace people; rather they allow people to stretch themselves and to increase company productivity."

Zezima sees a bright future for the Apple at Pepperidge Farm. "If technology continues the way it's going—the power going up while the price is going down—we'll keep the computers working."

Would it be possible eventually to develop Pepperidge Farm recipes on the Apple? Zezima says no. "The top chefs in Europe are recruited to become Pepperidge Farm bakers. As they've done in the past, they'll continue to come up with the most creative bakery delights."

Arthur Young & Company. Insiders at Arthur Young & Company believe microcomputers have had an almost revolutionary impact on automation and business, and this international accounting and consulting firm is happily supplying the artillery.

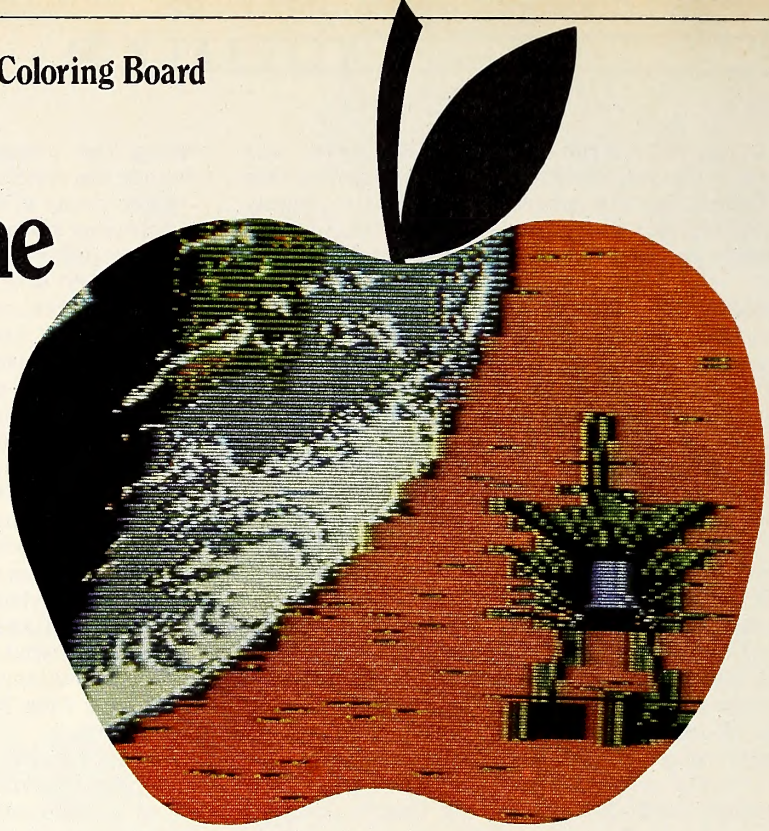
Rick Richardson, national director of computer auditing, who's also known as "the father of Apple use" at Arthur Young, first introduced the company to the Apple three years ago. His goal was to reduce the tab for the time-sharing system used by the firm's seventy-seven domestic and two hundred eighty foreign offices.

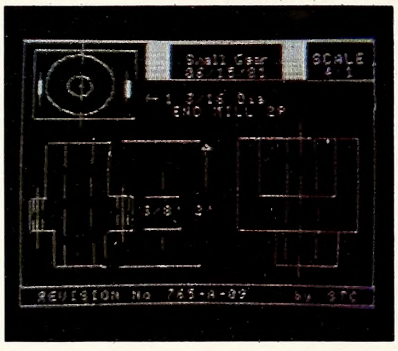
"When I first saw the Apple, not many people had heard about it, but I decided to give it a try," explains Richardson.

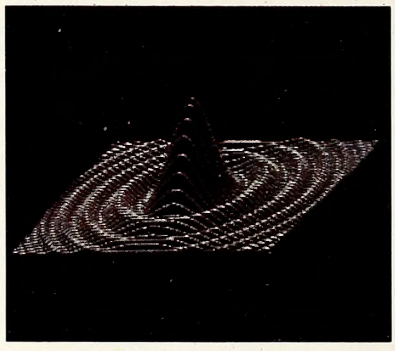
He found it to be more than convenient. It took only one con-

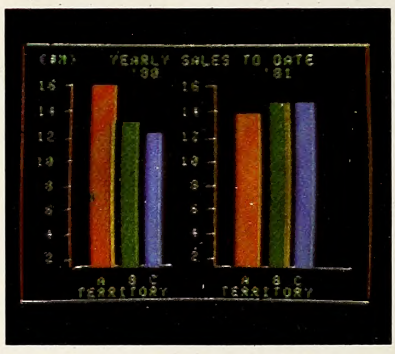
Introducing STC's Low-Cost Coloring Board

Put some color in your Apple for only $60.









$60. vs. 800.
Introducing low-cost color graphics for Apple computers.

To our knowledge, many other color graphic capabilities for the Apple cost at least $800. Now you can create, change, store and retrieve colored drawings for only $60. with STC's new Coloring Board program.

For Education, Business, Industry, Entertainment.
There are virtually unlimited uses for the Coloring Board, any situation that can benefit from graphic displays: classroom instruction, sales meetings, engineering parts drawings, home entertainment, you name it. You get high resolution and six colors: black, blue, white, green, orange and violet. You can combine graphics with alphanumerics of various sizes and colors, upper and lower case.

Easy and Fun To Use.
The Coloring Board is easy to learn and fun to use. No special equipment is needed, just your standard Apple game paddles (or joystick) and keyboard. And we've made it even easier. You can automatically generate arcs, circles, squares, rectangles, ellipses, triangles and other shapes. In addition, the basic package comes with a map of the world, a map of the U.S., a space shot and more. What's more, you can directly output your graphics to a printer.

All for only $60.

Contact Your Local STC Dealer.
We'll be happy to direct you to your local STC Coloring Board dealer. Just write or call Software Technology for Computers, P.O. Box 428, Belmont, Ma. 02178. (617) 923-4334.

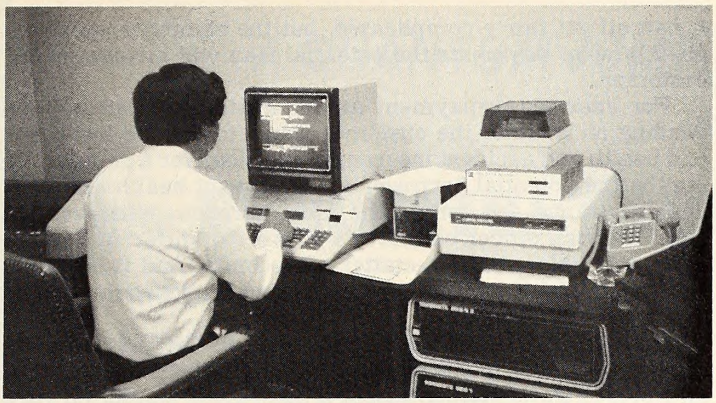
Be sure to look at our other business packages as well.

Dealer inquiries invited.

(Apple is a trademark of Apple Computer Company)

stc

SOFTWARE TECHNOLOGY for COMPUTERS
P.O. Box 428, Belmont, MA 02178 (617) 923-4334



Computer audit manager Dan Oshita (left) at work at Arthur Young & Co., while manager of production control Harald Theida of Pepperidge Farm (right), keeps his products nearby for inspiration.

version to put the company's Random Supply Appreciation Sharing service on a floppy disk, and the diskette, in turn, could be provided to various offices. It's estimated that one hundred Apple IIs are now used in everything from management consulting to tax practice.

But the real breakthrough at the company is the adaptation of microtechnology to the audit process, which gets a lot of help from the Apple III.

III's Brains of AuditComputer. AuditComputer is the new process developed by the company to transmit financial data directly from a client's computer to the Arthur Young computer system for audit testing. An Apple III computer provides the central processing and control unit that runs the audit software used for testing. Two other components make up the rest of the system: a remote data capture unit and a mass data storage unit.

Many of Arthur Young's clients are in the computer business themselves, and AuditComputer links with almost any type of electronic data processing system, while still respecting client data security standards. According to Richardson, the system expands the ability of their auditors to use the computer as an audit tool.

"The AuditComputer permits us to use sophisticated testing and analysis techniques to audit data produced by minicomputers as effectively as we've audited data produced by large mainframes," he says.

The system receives client data over standard telephone lines via direct hookup to the client's computer or by reading floppy diskettes.

One of the benefits of the AuditComputer is illustrated by a client, a multinational company that has a division operating in the European Common Market. This division manufactures word processing equipment and uses a standard cost system to maintain control over inventory. Each manufacturing plant might have different equipment and programming languages, but the AuditComputer system can do the audit testing regardless—using one set of customized audit testing programs.

AuditComputers hit Arthur Young offices only last month, and employees are being trained on them now. The system will be available both to the company's domestic offices and to the international offices from Argentina to Zimbabwe.

Eastman Kodak. When the Eastman Kodak Company of Rochester, New York, was ready to convert their technical education training programs over to the Apple computer, Denny McElroy was a step ahead.

Before May 1979, McElroy was an equipment service representative with Kodak in Oklahoma City, where he was responsible for repairing Kodak products from pocket cameras to computer output microfilmers. But he also had a personal interest in and considerable expertise with the Apple. One of the pioneers in Apple ownership—he bought his Apple, serial number 1,960, in 1977—McElroy used his home computer for business applications including scheduling and project analysis. So, when Kodak needed a project leader to develop a computer training program for Kodak service representatives, McElroy was the natural choice for the post.

Equipment service representative is a complicated job at Kodak. One of the leaders in high technology, Kodak is constantly turning out new product. Repair people scattered in Kodak shops across the country and around the world have to keep up with all the new developments in equipment and procedures.

But Kodak is a progressive company, according to McElroy, so it turned to microcomputers to help train new employees and offer refresher programs for veterans. Several systems were investigated before Kodak finally decided on the Apple.

They Could Go Home Again. Before the computer program was instituted, freshly hired equipment service representatives were trained in the company's Rochester headquarters. This meant a lot of airline fares and hotel expenses for the company. It also meant that new employes in any of Kodak's thirty-eight district offices were dependent upon the training schedule in Rochester. "Now we can sit employees down at the

Apple immediately, in the comfort of their own home offices," says McElroy.

The computers don't replace the human interaction of training; they enhance it, explains McElroy. Employees will usually spend the morning of their training day at the Apple, looking at a program written in Pilot instructing them in equipment repair. The courseware includes simulated problems to give the trainees needed practice in troubleshooting. In the afternoon, the students will go out in the field with instructors to get hands-on reinforcement of what they learned from the computer.

The training and refresher courses on the Apple are highly motivating because they're self-paced. "People are very pleased with the program, probably because humans are geared to special effects—they need attention-getting devices like the graphic displays on the computer."

In the future, Kodak plans to interface videodisk programs to the Apple, according to McElroy. The first two Apples will be installed in district offices in October, eight more will be placed by December, and, by January, all thirty-eight district offices will have their own Apples for training new employees. After that, McElroy says, they'll probably start sending Apples to the Kodak offices overseas.

The Bank of Louisville. Getting a loan from The Bank of Louisville, Kentucky, is a process that always starts with the Apple. There are twenty-seven branches in town that have among them some one hundred fifteen Apples to help determine whether a client will get that new car or new house.

Robert MacElwain, senior vice-president at the bank, explains that the procedure starts with loan shopping, using the Apple to calculate loan payments. Various factors are involved—how much for how long, repayment schedules, and what type of insurance the client might want to add to the loan. It can all get fairly complicated, but the computer spells out exactly what payments the potential loan will involve for the customer.

For example, repayment amounts will differ greatly depending on whether the customer wants to add life insurance and health and accident insurance to the loan for a new car. Or maybe the potential borrower will just want health and accident insurance and forego the life insurance. Whatever the case, the loan officer can have a number of different cost options available within a matter of seconds. And it certainly beats flipping through a loan book to find the schedules one by one, says MacElwain.

Once the bank and customer determine the best loan and the best payment plan, the application for a loan is entered on an Apple in the customer's local branch and transmitted via a Hayes modem to an Apple in the main branch.

The application is then reviewed by a loan committee for approval, their decision is relayed back to the appropriate branch office, and the client either buys the new car or takes up hitchhiking.

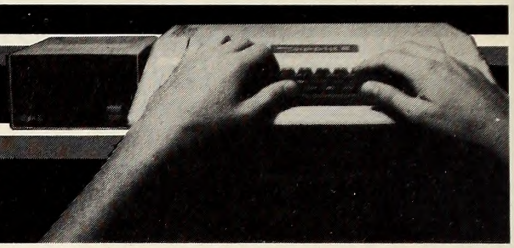
Apple Takes Out after Fraud. Besides loans, the bank uses the Apple as a central communication device. One of its functions involves publishing a loan fraud hotline to keep employees on the alert.

There are numerous other uses for the Apple at the Bank of Louisville, and MacElwain said it would take an entire day to describe them all. But he did mention the use of Apples as standard word processors, portfolio evaluators, and determiners of pay-off penalties. "The employees enjoy the microcomputers. The reception was good," says MacElwain.

Many customers also see the benefits of the microcomputers in finance, especially those who get the Apple seal of approval for their loans. ◨

MARKETALK

Reviews



Unless noted, 48K and one disk drive are required.

Tawala's Last Redoubt. By Douglas Carlston. Tawala evidently put up a lot of resistance, but after nearly six months' wait, he must finally face his last battle.

It's a pleasure to say the wait was worthwhile (for us, of course, not for Tawala).

Tawala's Last Redoubt is the fourth in the Galactic Saga series of strategy space adventures linked by characters and plot. In this episode, Tawala has already been deposed and has taken up residence in a well-guarded fort on the planet Farside.

You are Benthi, leader of the rebel forces and, incidentally, a beautiful woman. Your job is to plan and carry out a strategy that will overthrow Tawala once and for all.

As in the rest of Galactic Saga, you play in real time. Because action is unpredictable, you must constantly update your strategy, all the while keeping track of your forces and spies and attempting to keep track of Tawala's forces through espionage and intercepted messages.

One great innovation is that the messages are in code. Still in real time, you must break the code and read the message.

Possible help may come from the four regional rulers of Farside; yet they'll as willingly send your messenger's head back on a platter as fulfill your requests. An outlaw roaming the hills with a band of fighters is a better source of help—if you can find him.

Logistics and timing play an important role in Tawala, as in all the saga games. Frustrating at first, this activity is an excellent educator besides being thoroughly enjoyable once you get the hang of it.

With all this, it's something else that makes *Tawala's Last Redoubt* a quietly significant milestone in microcomputer entertainment: this game goes beyond being a game; it is a first step, albeit a faltering one, toward creating in microcomputing a form of literary and visual art. Doug Carlston is not afraid to let his feelings show and to present them to you in a sensitive, artistic way.

Therefore, in some situations when you lose, you're treated to the vision of a surviving rebel running up a hill against a backdrop of stars. At the top he stops and, saluting, pays tribute to Benthi and offers hope that there'll be another day on which they'll prevail. The words scroll away, but the one saluting soldier against the sky remains. As long as you like.

The graphics aren't abundant in *Tawala*; but they are exquisite. Camp, towns, the redoubt, each is illustrated with colorful, attractively detailed still pictures. The people, scaled to the backdrops, move with a fluidity and reality that make even *Olympic Decathlon* seem obsolete. Well, almost. MCT

Tawala's Last Redoubt by Douglas Carlston, Broderbund (San Rafael, CA). $29.95.

Computer Baseball. By Charles Merrow and Jack T. Avery. The baseball strike has been struck, so you don't need your Apple to get a daily fix of the grand old pastime. But if you're a fan of sports simulation games, you won't be able to pass this entry up.

The authors have done a remarkable job of simulating the real-life abilities of real players on screen. In fact, the thoughtful algorithms that define the outcomes show the power of the computer to its best advantage. But nobody will buy this game to show off the programming, because that's all behind the scenes.

What the original game contains is twenty-six teams of the past, representing the adversaries of thirteen famous World Series. You get the chance to see if your managing could have changed the outcome. Given this computerized reprieve, can Mordecai "Three-Finger" Brown pitch the Cubs past the 1906 White Sox? Can Hank Greenberg slug the Tigers past the Gas House Gang of the 1934 Cardinals? Will the Indians ever win a game against the 1954 Giants?

Only a few games will be enough to convince you that you make a difference as manager. The Waner brothers, "Big Poison" and "Little Poison," will hit, but when should the 1927 Pirates steal against the Bronx Bombers? When should they hit-and-run? When should you change pitchers? Is a defensive change called for in the late innings?

Particularly critical is the manager's handling of the bullpen and the infield positioning. An otherwise relatively even match can be turned into a rout by an inept manager fouling up his own players.

Computer Baseball gives you the option of playing against a friend on the computer. The computer manager, Casey by name, is an opponent whose moves—especially in regard to his pitching staff—are guaranteed to astound.

A nice touch is the box score that is displayed on screen or printed out at the end of the game for players who want to replay entire seasons.

SSI has provided a utility on disk that allows you to enter and play with teams of your own creation or with teams of another season. You can even enter your softball league or your child's Little League team. Or, in the case of the big leagues, you can buy a disk for $15 that holds the 1980 teams.

Graphics is still not an SSI strength. *Computer Baseball* has just enough to prevent ennui while you watch the screen. But as a baseball simulation, it's presently king of the hill. ART

Computer Baseball, by Charles Merrow and Jack T. Avery, Strategic Simulations Inc., Mountain View, CA. $39.95.

Sneakers. By Mark Turmell. You've only got five space ships. Not only do you have to get past the stomping sneakers, you've got to survive the cyclops, saucers, fangs, and H-wings.

Better get them before they get you! Move your ship back and forth across the bottom of the screen with paddles and fire lasers at your foes. But don't get trigger happy with the cyclops. With every missed shot, the formation of cyclops moving across the screen drops lower and closer to your ship. So make every shot count or you'll be creamed.

This new game from Sirius Software has kept half our staff on their Apples after hours for the last week. You'll be glued to your screen too when you try to take on the H-wings. It's speed that counts as H-wings crisscross the field and try to descend on you. Move fast and keep firing.

While you're shooting at the hovering fangs, don't be surprised when your target turns into a bomb and tries to snuff you. Speaking of bombs, there's a whole field of bombs to maneuver through while you try to knock off the flying saucer. Surviving any one of the first five hazards without having your ship destroyed earns you bonus points.

You'll need all the bonus points you can get because, if you do survive, you'll find yourself dodging through a field of small and large meteors. Shoot 'em fast. You get twelve points for the small one and eighty-five points for the large. But if you get hit first—watch out!

When you do get hit, all is not lost; a space pod comes down to give you one of your remaining four ships. Don't worry if you have trouble keeping track of your reserve ships. A shrill coach's whistle sounds with your last craft, reminding you that the end is sneaking up.

CONTINENTAL SOFTWARE

Module #4

PAYROLL

is now available

- Personnel records for 100 employees • Prints checks • Prints W-2
- Security passwords • Quarter & year-to-date earnings • State tax
- Fed. tax • Local/city tax • Standard pay periods • Salary/Hourly Pay
- Optional pay periods • Commissions • Disability ins. deductions
- Handwritten check entry • Payroll summary • Employee master list
- Check register • Payroll register • Employee's earnings record
- Payroll general-ledger listing

*Computer Programmed Accountant

AT YOUR DEALER **NOW!** **$250** per module

#1 GENERAL LEDGER #2 ACCOUNTS RECEIVABLE
#3 ACCOUNTS PAYABLE #4 PAYROLL
#5 PROPERTY MANAGEMENT

All modules are stand alone or may be used as a part of an integrated system.

12101 Jefferson Blvd., Culver City, CA 90230 • (213) 371-5612

**APPLE][is a registered trade mark of Apple Computer Inc.
Hardware requirements: 2 disk drives, DOS 3.3, APPLE][48K, 132-col. printer and Applesoft in ROM.

If you're not quick to dodge the triangular scrambles after the field of meteors, you'll use up those remaining ships fast. By the time you aim and fire, the scrambles change position three times.

Finally, you meet the scrubs. Resembling candy-striped barber poles, they sweep diagonally across the screen. On the first pass, let them go over you. But be quick to move back to the other side of the screen before they pass again or they'll trap you in a corner. As if dodging the scrubs weren't enough, these little devils turn into vertically dropping bombs once they've been hit; they fall from wherever you've fired, so keep moving.

Mark Turmell has designed a real winner here. The game's no piece of cake. If you can get beyond level two, you're doing better than all but two players in our office.

Turmell's graphics and animation are unique. Not only are the characters created in a new style, but their fluid, responsive motion breaks new ground.

Only one problem: how do we get the staff away from their screens and back to work? CV

Sneakers, by Mark Turmell, Sirius Software, Sacramento, CA. $29.95.

Pro-Paddles. By Rainbow Computing. It takes a few games to get used to the size of Pro-Paddles; at approximately three inches by two inches by one, they feel very small in any adult hand. But, once you adapt, any larger paddle seems cumbersome.

Size, however, is not among the crucial factors in paddles; speed, accuracy, and reliability are. That the buttons don't cause what circa 2500 A.D. archeologists are apt to call Apple Arcade Finger may be of secondary import to some, but it's certainly desirable.

On three of these four counts, Pro-Paddles come in magna cum laude. Their speed and accuracy are outstanding, and a plastic square finger-pad-sized button means total comfort. A minor drawback is the slight play in the button if you don't hit it evenly every time. You might yearn for the absoluteness of the Apple paddle button, but, if the tradeoff is a firm button that doesn't always fire for a wiggly one that never fails, wiggly wins.

The fourth count, reliability, is the one most paddles fall down on. Apple paddles, which are super paddles (despite the pain . . .) are renowned for their short life, at least for arcade game addicts. Pro-Paddles' maker, Rainbow Computing, begged us to do our best to crash these—and we tried. Albeit the trial lasted only a few weeks, the Pro-Paddles showed no sign of weakening.

Thus, Pro-Paddles earn a conditional cum laude on reliability as well.

As usual, there's no way to distinguish paddle 0 from paddle 1 until you start playing.

Try Pro-Paddles at your dealer; if your hand is extremely large, they may be too small for you. Otherwise, you're apt to take the little buggers home. MCT

Pro-Paddles from Rainbow Computing, Northridge, CA. $39.95.

Apple Panic. By Ben Serki. Who could ever be panicked by a bunch of apples—the eating kind? *Apple Panic* takes place in what appears to be a building under construction. There are five levels connected by various lengths of ladder. You are a little fellow whose job is to rid the area of a plague of apples and, later, butterflies, by digging holes on all levels for the critters to fall in. Then the challenge is to cover them over when fall in they do. The critters' job, however, is to keep the area free of holes and to get you.

If this sounds a little buggy to you, you're getting the picture—and here's where the panic sets in. The apples resemble nothing so much as horrible insects, especially in their movement and, more especially, when they attack you. Almost inevitably, you see that you're a goner an instant before the thing jumps, and that unavoidable moment's anticipation followed by the crackly noise as you're eaten is plenty to raise your adrenalin to a darn good semblance of panic—followed by "Ugh!"

When the butterflies join the ranks, you find they must fall through two—later three—levels to be done in, meaning that you and your little shovel must dig holes directly above one another. When one does fall through, the holes fill up and you must dig them all over again to catch another creature.

To top it off, there's a limit to the air supply in this strange place. Besides losing bonus points for taking a long time to get all the critters in each group, you could run out of air and lose.

Despite the unpleasantness of these feelings, at least in real life, in the game, they seem only to make you more determined to kill all those darn bugs—uh, apples. *Apple Panic* is extremely addictive.

Graphics and play are up to Broderbund's usual high standard of excellence. MCT

Apple Panic by Ben Serki. Broderbund Software, Eugene, OR. $29.95.

Hi-Res Adventure #3: Cranston Manor. By Harold DeWitz and Ken Williams. It's been a long wait since number two: *The Wizard and the Princess,* but On-Line is finally back with number three, the fourth (because they count like computers) in their supersuccessful line of hi-res adventures.

Let's amend that. On-Line is back, hi-res adventure is back, Ken Williams's programming is clearly in evidence—but Roberta Williams is not. *Cranston Manor* is the creation of Harold DeWitz. It's an able job of adventure devising; but it lacks the plot and purpose we've come to expect and enjoy in On-Line adventures. Those elements are Roberta's bailiwick.

Nevertheless, *Cranston Manor* is much fun to play. While it lacks a plot—you never will know the story behind the robot-like living suits of armor or the deadly tin soldiers—it is riddled with good stumping riddles. It does, at least, have a purpose: to collect the sixteen treasures old Cranston hid in his manor and deliver them from the manor gates to the tiny town outside them—called, coincidentally, Coarsegold.

The entire program comes alive in Williams's immediately recognizable full-color hi-res style.

Rest assured that Roberta Williams will be back with new adventures. She's spending all her time now on number four—a giant multidisk adventure that will take you all over history in living color. MCT

Hi-Res Adventure #3: Cranston Manor, by Harold DeWitz and Ken Williams, On-Line Systems, Coarsegold, CA. $34.95.

Soundchaser. There's a new polyphonic synthesizer in the Apple marketplace. Soundchaser is a modular six-voice system, rather different in design from and less expensive than the AlphaSyntauri. Fully configured, Soundchaser consists of two three-voice cards and a forty-nine-note keyboard. Each of these items is available separately—the cards for $350 apiece and the keyboard for $650—so the Soundchaser components can be interfaced with home brews.

This system takes a subtractive approach to waveform synthesis. On each card there's a DIP switch that selects either sawtooth or square as the basic timbre for the voices controlled by that card. This primary wave is then modified by means of a low pass resonant filter and a low-frequency oscillator. This approach more closely mimics the style of most so-called conventional synthesizers than does Syntauri's additive method—a fact that some musicians accustomed to working with filters and LFOs may appreciate.

Software included with the system allows the player to draw envelopes, using the game paddles, that control a variety of sound parameters, including loudness, pitch, resonance, vibrato, and the cutoff frequency of the filter. A bank of sliders and soft switches, also wired through the game port, controls such other things as the pitch of the overall keyboard and the time period over which the various envelopes act. The keyboard covers four octaves at any one time; one of the soft switches lets the player transpose it up or down in octave steps, so that the entire range of the instrument is eight octaves.

Any particular arrangement of envelopes and other parameter settings can be saved to disk as an instrument and recalled for later use. Soundchaser will hold four instruments in memory at any one time, allowing for relatively quick interchangeability.

There's also a sequencer that allows the player to record songs or parts thereof. If all six voices are not used for the sequence, then the player can perform live over prerecorded material when the sequence is played back. A tempo feature controls the speed at which the sequence is played back; it's possible to achieve as much as a sixteenfold increase or decrease in tempo on playback.

Up to four sequences can be stored in memory at one time, and these four can be linked in any way the player chooses with the four instruments in memory. Sequence two can be played back using instrument four, or whatever. A group of four sequences and four instruments constitutes what the Soundchaser software calls an orchestra and can be saved and reloaded as a block.

One of the more interesting features about this system is the fact that the user actually draws envelopes on the screen, rather than simply specifying numeric values for attack, decay, sustain, release, and so on. This has advantages and disadvantages. On the plus side, the envelope can have any number of inflection points. That is, rather than just drawing some variation on the ADSR curve, the player can also create an envelope that goes up and down eight or ten times.

Since Soundchaser envelopes may, at the player's option, modulate pitch and other qualities, as well as loudness, the provision for drawing envelopes, rather than just dialing in numbers, adds much to the versatility of the instrument. On the other hand, it makes it a little more cumbersome to use. If the player wants to change an attack rate, for example, he has to go into the edit mode and redraw an envelope; he can't just hit a few numbers on the keyboard.

Furthermore, the way the paddle is used to draw envelopes on this system is a touch clumsy. The dial controls vertical position, and the button enters points on the curve. Movement along the curve's x axis is automatic; while the paddle button is held down, the curve moves. For exact work, points need to be entered one at a time, which is awkward, because if the button is held down a moment too long, an erroneous blip is entered, and there's no simple way to erase it. It would be nice if the software provided an option for creating envelopes with the Apple keyboard as well.

It would be nice, too, if there were a way to toggle the basic waveform without having to reach into the computer and fool with a DIP switch.

In short, Soundchaser is a very powerful system that's not real easy to use. Any musician with curiosity and patience can learn to do a great deal with it, but it is not ideally designed for experimentation or subtle changes during the heat of performance. CS

Soundchaser, Passport Designs (La Honda, CA). Keyboard $650; 3-voice cards and software, $350 per card.

Dragon's Eye. By Robert Leyland. Your quest, according to a medieval game manual, is to find the gem that is enabling evil to reign in this fantasy country. In fact, finding the gem brings you few kudos; the real quest is high score—which requires much more than fulfilling the duties prescribed.

Once given a character when the game begins, you choose your name and one of sixteen titles, which the Tailor of Dreams quickly sews on your cloak. You select a sword and learn a number of spells.

The hi-res, one-screen map that you travel on during this game consists of seven cities or provinces, each of which contains three to nine locations. Somewhere in one of these areas hides a vicious red dragon guarding the sacred Dragon's Eye gem. You must find this jewel and return it to your original starting place within a twenty-one day limit.

The monsters and map graphics in this program are wonderful. When you come upon a skeleton it looks like a skeleton, the dragons look like dragons, the vampire bats look like vampire bats. . . . They're all carefully done in fine-lined, full-color, animated hi-res—which makes all the difference in the world.

Throughout the game, several types of monsters, besides those mentioned, pop up and try to kill you. Killing them first gains you points and sometimes treasure. Scenarios like this—with appropriate graphic animation—will often occur: Smash! A hit. Chop—too far. Thrust—missed. Magic bolt, a hit! Monster condition: berserk. Fire (an arrow); a hit! Monster condition: dead. Some monsters' conditions can change from "100%" to "tickled" to "dead" with only a smash or a chop; others take several smashes, chops, and thrusts just to become "afraid." This is what makes the game interesting and fun to play.

If you're the loser in battle, the good wizard appears to attempt your resurrection. In funky animation, the wizard figure turns, slightly crouches, and raises his arms, fingers spread to let the magic flow, as he shouts (via a cartoon-type display), "Revive!" Whether you do or not, this sequence is worth the battle loss in chuckles.

So, if you're into the days of chivalry, fighting dragons for beautiful treasures, and all that good stuff, then this is a game for you. RT

Dragon's Eye by Robert Leyland, Epyx Division of Automated Simulations, Mountain View, CA. $24.95.

Cool Stack. FMJ's entry into the efficient-system market is Apple-colored and close to Apple-textured, which means it blends in nicely with your computer equipment. Black piping around the edges adds a touch of style.

But the main feature of the Cool Stack is its fan. Built-in and perpetual, the fan should be placed on the right side of your Apple, where it can blow away the heat of intense electronic thinking and electrical power. This is the quietest of the outside-the-Apple fans.

Cool Stack's only problem is also one of its features: its custom fit. This stand is made to fit the Apple II and its disk drives precisely, and the sides fold under at the bottom to hold it all securely. Unfortunately, if you frequently have occasion to get inside your Apple, this can be a slight hassle.

Useful companions to the Cool Stack are an L-shaped piece that hooks under your Apple to form an adjustable-size bookend for disks and computer books and a printer stand (Printer Pal) that raises your printer off the desk and accommodates a paper supply directly underneath it. Both pieces are color-coordinated with the Cool Stack and Apple.

Cool Stack is reasonably priced at $69.95, especially considering its efficient, quiet cooling system. But the bookend at $6.95 and the Printer Pal at $29.95 seem slightly overpriced. Still, the whole works makes a tidy system. MCT

Cool Stack, RMJ, Torrance, CA. $69.95.

President Elect. By Nelson G. Martinez, Sr. It's 10:22 p.m. November 5, 1984, and ABC-TV has just projected the landslide victory of Jack Kemp over Ted Kennedy for president of the United States. Jack who? Oh, yes—the young congressman from New York who was considered a vice-presidential possibility in 1980. Considered approximately equal to Kennedy in speaking ability, charisma, and poise according to the ratings in Strategic Simulations's game, Kemp's views are evidently those expected to be more popular in 1984, and it would seem to be so. *President Elect*, playing both candidates, left no room for doubt in this simulation.

To test the theory, we had the computer pit Ronald Reagan with John Kennedy—a task none but the computer is up to—in the same year. Again, the more conservative candidate won easily, even though we denied him his incumbency.

The imagination begins to run wild. What if it were John Anderson versus Teddy? Phil Crane versus Jerry Brown? Or the historical losers: Gerald Ford versus Hubert Humphrey, Barry Goldwater versus George McGovern. And real-election replays: Nixon versus John Kennedy in 1960, Ford versus Carter in 1976, Reagan versus Carter in 1980. All the imaginary elections take place in 1984 and are influenced, one assumes, by the kind of thinking assumed to be prevalent that year. With that in mind, the computer-run election results, respectively, were: Anderson, easily; Crane, giving up only Washington, D.C.; Ford; McGovern; Nixon, very slightly; Ford, easily; Reagan, by a greater landslide than the real one.

All this is great fun, but allowing your Apple to turn into

Cassandra is only a sidelight feature of *President Elect*. It's also a good game.

President Elect comes with numerous candidates already input and rated, like the ones in the predictions, with which you can play; or you can enter real or imaginary candidates of your own. You can even call for the game's candidates, reject their ratings, and rebuild the candidates the way you see them. Rating a candidate involves evaluating his position on numerous issues on a nine-place scale from strongly against to adamantly for and evaluating his oratorical ability from impeded to "orator for the ages," his charisma from easily ignored to greatly influential, and his ability to deal with difficult situations from "falls apart" to "nerves of steel."

Third party candidates are allowed, but the only chance of their winning occurs if the major party candidates are both totally ineffectual people with speech impediments who fall apart at the slightest trial.

When you've got two candidates ready to go, you and another player decide which party is in power and whether the candidate from the party is the incumbent. You must also answer questions setting up the state of the nation politically, economically, and diplomatically as the campaign begins.

During the campaign, you can have your candidate or the running mate travel to a foreign country and make campaign stops throughout America; each week, you plan national, area, and city advertising, and decide whether to debate.

Debates are great fun. After deciding upon a number of questions, each candidate must input the percentage of time he'll spend giving his position, attacking his opponent, telling anecdotes, and wasting time. In rebuttal, similar choices include criticizing with witticisms or with moral indignation.

Each campaign week, pool results are given and shown on a map. Then the relevant world news is broadcast, and you're told if either candidate emitted any questionable statements or committed any downright gaffes that week.

The climax is election night: the anxiety's high and you choose the way you learn the results. If you have plenty of time or if you're playing *President Elect* as a party game—it lends itself well to group campaigning—you might want to watch the results trickle in minute by minute, with the computer projecting winners as it sees fit and announcing the new president when some lucky state's projection takes one candidate's electoral votes over the top; this can take as long as four hours. If you're less patient, you can immediately jump to the results; percentages per state are listed, and the map is available to display your victory or defeat graphically.

President Elect comes with game card and pad of campaign planning sheets, programmed and packaged with the care and high quality we've come to expect from Strategic Simulations. MCT

President Elect by Nelson G. Hernandez, Sr. Strategic Simulations Inc., Mountain View, CA. $39.95.

APPL Payroll System. By Tony Morazan, translated into Applesoft by Bonnie Minor. And: **Payroll.** By Hal Faulkner. There's a philosophy extant among some Apple folk that anything less than a full-bodied, interactive accounting system has little value. This body of thought holds that treating payroll, inventory, or accounts receivable as isolated items underutilizes the computer.

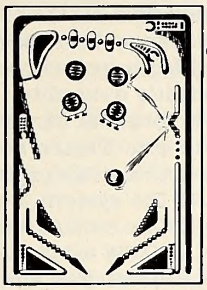
Such sincere convictions should not be discounted without thought, but these Apple users fail to consider that for many businesses, a full-bodied accounting system is too much.

These businesses normally employ an Apple as a multifaceted laborer, churning out projections, correspondence, and graphs as well as providing accounting functions. In this environment, it may be more efficient to have the Apple address only those accounting areas where application of computer power will produce significant savings.

Often, one such area is payroll, which can be a time consuming, laborious task, requiring meticulous attention to detail and exacting calculations of gross earnings, deductions, and net pay.

THE PROGRAM STORE

CALL TOLL FREE 800 424-2738



Raster Blaster

APPLE® For Apple II & Apple II Plus Applesoft, 48K Disk unless otherwise noted

By Bill Budge from BudgeCo
If this program can't make the "Pinball Wizard" in you come alive, nothing can! With superb, HiRes color graphics and sound, this game offers all the fun and excitement of the real pinball machines without having to feed it quarters. Bumpers and traps abound. As you increase your skill, you will truly appreciate the action and full control built into the flippers: you can hit the ball, let it slide to just the right spot, even catch and hold it for positioning. Fun for everyone!

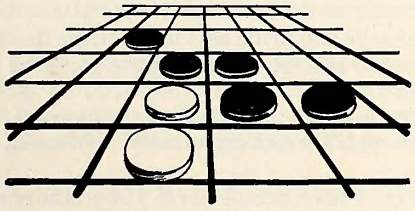
48K Disk...$29.95

ALIEN RAIN

From Broderbund Software
(Formerly sold as "GALAXIAN") Still one of the most popular games for the Apple, and no wonder: Excellent color graphics, high playing speed and real challenge combine to produce a classic. In ALIEN RAIN, the creatures swoop down from all sides. You stand alone against this onslaught, bravely defending earth from the swift and devastating invasion. You wipe out wave after wave of Aliens but they just keep coming! Not for the easily frustrated.

48K Disk...$24.95

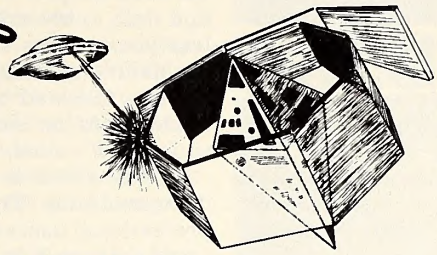
REVERSAL



By Dan & Kathe Spracklen from Hayden
Based on the 200 year-old strategy game Reversi, this brilliant program won the software division of the First International Man-Machine OTHELLO Tournament. You can learn to play in minutes, but mastering the strategy of this classic game may take a lifetime.

REVERSAL allows three playing strategies with nine levels of skill in each strategy. You can tell at a glance who is currently ahead by the expression on the playing pieces' faces, and if "asked" the computer will even kibitz with helpful hints on your play. Excellent use of color graphics and sound.

32K Disk...$34.95
Also by the Spracklens: SARGON II -- $34.95



PULSAR II

By Nasir Gebelli from Sirius
Inspired by the popular "STAR CASTLE" arcade game, PULSAR II combines action, color, sound and challenge to provide a thoroughly enjoyable and absorbing game.

Choose from three modes of play. In the PULSAR option, your objective is to destroy the Pulsar by blasting holes in the fence-like shields. The WORMWALL option challenges you to reach the center through a combination of permanent and momentary openings in the shield. In the COMBO mode, you alternate between the two options. The lower levels are fun for the beginner -- progressing to the higher levels requires skill and nerves of steel!

48K Disk...$29.95

AUTOBAHN

By Nasir Gebelli from Sirius
Take on the world's fastest highway in your Formula 7 racing car. Don't let the lush color graphics and realistic sound lull you into thinking you are out for a Sunday drive, because this road is out to getcha! Winding turns are the least of your troubles; you encounter other cars, trucks, dark tunnels and narrow bridges. Even the little puddles that sweep into view are a hazard at these speeds! The main object is to have fun and this program certainly provides it!

48K Disk...$29.95

the Prisoner™

By David Mullich from Edu-Ware
Venture to the nightmare of 1984 in this fantasy game inspired by the critically acclaimed TV series. Having tried to resign from "The Company," you have been drugged and removed to a prison island. The island's defenses have been designed to break down your reasoning powers and induce delusional thinking as it thwarts every attempt at escape. "They" want to know why you resigned, you want to escape; and it will take all your intelligence and sense of individuality to do so! "The Prisoner" provides a unique challenge that sets it apart from other adventure/fantasy games.

48K Disk...$29.95

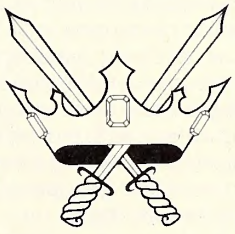
DOGFIGHT

By Bill Basham from Micro Lab
Aerial action for the Apple. DOGFIGHT can be played against the computer, two players teamed against the computer, or two players on different teams. You can even set up your own game with the "Custom" mode.

Fly your plane around the screen in battle against the enemy. Your gun holds four bullets at a time and if you run out you will be a sitting duck while waiting to reload. If you shoot down the enemy, he may try to parachute to safety. Follow him down -- guns blazing -- because if he makes it, he will be back in another plane! Lots of action, graphics and sound.

48K Disk...$29.95

Crown of Arthain



By Dan & Marilyn Meller from Micro Lab
In the Mountain Kingdom of Arthain, the King lies dying. Though he has banished his two sons to the outer reaches, he now calls for their return, and the one who successfully completes the journey may take the crown.

This unusual adventure may be played against the computer or by two players simultaneously. Colorful HiRes graphics and 20 skill levels add to the fun as you magic and monsters along your way.

48K Disk...$34.95

3D GRAPHICS & GAME TOOL

By Bill Budge from Top of the Orchard
Frustrated by the tedium of programming 2-D or 3-D graphics and animation? Awed by the great graphics of Bill Budge's games? With 3-D GRAPHICS/GAME TOOL you can design your own graphics and animations, limited only by your imagination!

Though useful to the expert, this program allows even novice programmers to create impressive animated graphics that move smoothly at frame speeds of 10-40 frames per second. A 52-page booklet shows you how to use it. Designs you produce with the program can be interfaced to either Integer or Applesoft BASIC, so you don't even need to know assembly language programming to benefit from this powerful tool. Order now -- this may well be your most used program!

48K Disk....$39.95

Visit Our Baltimore, MD Store: W. Bell Plaza – 6600 Security Blvd

TO ORDER CALL TOLL FREE 800 424-2738 For information Call (202) 363-9797

MAIL ORDERS: Send check or M.O. for total purchase price, plus $1.00 postage & handling. D.C. residents, add 6% tax. Charge card customers: include all embossed information on card.

THE PROGRAM STORE
4200 Wisconsin Avenue NW, Dept. OA09 Box 9609
Washington, D.C. 20016

In small businesses where the proprietor is also the paymaster, supervisor, middle manager, check signer, and so forth, calculating payroll by hand is not particularly productive.

Here is where a stand-alone payroll package may be of benefit.

The two packages considered here, *Payroll* from Broderbund Software and *APPL Payroll System* from Computer Products International, are more similar than different. Both require two disk drives, both print checks, both leave an adequate trail for later auditing, and both are sufficiently documented and user-friendly; a buyer may be reasonably sure of relatively rapid integration of the system into the business.

The most obvious difference is that Broderbund uses five disks to accomplish their task while Computer Products International requires only three.

Both systems have two program disks. Computer Products writes all its data to a single data disk. Broderbund writes employee data to an address disk, payroll data to a data disk, and uses yet another disk for computations.

A five-disk system may seem unnecessarily cumbersome, but in certain situations the use of specialized disks can help to define the system more easily for the user. And Broderbund sends the data flying between disks so rapidly that the lost time in disk switching is gained in operating.

Entering employee data, the first task in both systems, is equally as efficient; making alterations to the data can be done quickly. Broderbund includes a provision for tracking the ethnic backgrounds of employees, a feature that would seem more suitable to a fuller-bodied system in a larger company.

Both systems allow segmenting the employee population into departments or groups to allow for different taxation schedules at different sites.

When it comes to calculating deductions—tax and otherwise—Broderbund's extra disk system permits them to be more ambitious.

Both systems will calculate federal and state payroll tax and social security deductions. Broderbund's table will also automatically calculate some city taxes and state disability taxes as well. A table in the documentation permits proper keying for those deductions.

In addition, Broderbund allows for as many as thirty special deductions, such as insurance, credit union savings or loan repayment, and so forth, while Computer Products makes provisions for ten. The slot for the tenth deduction is a trackable slot where companies can place their city tax or state disability tax.

Both systems provide standard reports by individual, payroll period, quarter, or year. Computer Products provides a workmen's compensation module not found in Broderbund's system; Broderbund offers a flexible employee listing module that permits formats not yet designed by man or beast.

Payroll checks, payroll registers, and yearly W-2 forms can be handled with approximately equal ease by each system. Broderbund's system, being the newer of the two, has a built-in configuration for the Epson MX-80 printer, but it is a minor effort to configure either package for any suitable printer.

There seems little to choose between these two systems. Broderbund's documentation is more attractive and easier to understand, but Computer Products International's system was implemented in exactly the same total time, which indicates that their documentation is no hindrance. Either package may be depended upon to cut the drudgery of payroll calculations to a minimum. ART

Payroll, by Hal Faulkner, Broderbund Software, San Rafael, CA. 48K, 2 disk drives, 96-column printer. $395.

APPL Payroll System, original design and CP/M programming by Tony Morazan, translated into Applesoft by Bonnie Minor, Computer Products International, Metairie, LA. 48K, 2 disk drives, 132-column printer. $395.

HSD ANOVA. By Stephen Madigan and Virginia Lawrence. Here's a second-generation piece of Apple software that exemplifies how specialty software should be done.

What shows on the screen of this statistical analysis package is modest to the point of severity. But the underlying program is a powerful one that gives an Apple-owning researcher a cost-effective analytical tool. It's software of this nature that ensures a long and varied life for the Apple II.

ANOVA is not a general interest package. Its raison d'etre is variance analysis of data obtained from experimental observation. Within that limited context, it's dynamite.

It'll allow analysis of up to eight factors that can be either between-subject or within-subject factors. Factor definition and data entry are easy, and the calculations are prompt, unless you test the extreme capacities of the system.

Mean and standard deviation results are calculated immediately, followed by a display of the ANOVA table, which includes data on sums of squares, degrees of freedom, mean squares, *F*-ratios, and *F*-probabilities.

If the above is Greek to you, don't look for help from the documentation. The authors wrote a professional program for professional users and documented it accordingly. If you don't know that the *F* is in honor of R. A. Fisher, you don't need this one. ART

HSD ANOVA, by Stephen Madigan and Virginia Lawrence, Human Systems Dynamics, Northridge, CA. $74.95.

The Portfolio Evaluation and Reporting System. By Gregg S. Wilson. *The Portfolio Evaluation and Reporting System* is tailor-made for stockbrokers who want to go the extra miles to service their accounts.

This detailed package will report on almost every kind of transaction a broker might have for his clients, from standard stocks and bonds to options and commercial paper.

Designed to run with the Hayes Micromodem II so holdings can be updated via the Dow Jones Quotation Service, PEAR can give a broker the immediate picture on any of his clients or print out a summation periodically for the client. The printed reports are more detailed and useful than those you normally get from your brokerage firm's mainframe. On the other hand, the individual broker has to generate PEAR's statements.

Because this data is complicated, getting the system on-line is not easy. But after the initial labor of setting a client up, the available utilities and reporting formats make PEAR a breeze to use.

For a broker who sells full price service in these days of discount brokerage houses, PEAR and an Apple could be a nifty and profitable investment. ART

The Portfolio Evaluation and Reporting System, by Gregg S. Wilson, PEAR Systems, Stamford, CT. 2 disk drives, Micromodem II, 132-column printer. $500.

Adventure in Time. By Phoenix Software. The authors are admirers of the adventure work of Scott Adams. When they chose to write an adventure themselves, they attempted to pattern their work after that of Adventure International. The result is not a carbon copy, but a unique escapade in its own right.

As its name implies, this game allows you to move around in time. What's more unique is that you don't move in space—at least, not very far. Your time machine takes you to several eras in the history of the world, but each one takes place in the same spot in the world. Foliage, pathways, buildings, and type of animal life change, but the land is the same. Each time period you conquer lets you to get to another period, although once visited, all are accessible whenever you like. A good touch is that the time periods are not pure imagination; each is based (if loosely) on a real era; real people and places crop up. Watch out for the lions in the Colosseum. . . .

Your task is to destroy a certain famous criminal who wanders freely in time and whose inevitable goal—unless you succeed in reversing history—is to destroy the world.

The adventure is filled with puzzles, most fairly straightforward and not all terribly difficult. Be sure to save the game often, however. Adams's influence shows occasionally in hopeless and unchangeable traps. Most of them appear quickly in their eras and are pretty clearly hopeless. Their saving grace is that, once you've been their victim, you never need

visit them again. More of the puzzles are good posers that will keep you guessing for some time.

Adventure in Time heralds a new entrant into the catch-on text adventure genre. The company shows promise of being unable to avoid a well-based logic. We'll watch their progress with interest. MCT

Adventure in Time, Phoenix Software (Lake Zurich, IL). $29.95.

HSD Stats. By Stephen Madigan and Virginia Lawrence. The authors are two college educators who observed the void in serviceable Apple software for the social sciences.

Computers are particularly adept at number crunching, and social science experimenters are particularly skilled at generating large sample populations that need to be crunched for analysis purposes. So it's in the natural order of things that a marriage of such service and need be made.

What *Stats* provides is a reasonably comprehensive body of analytical tools for the researcher and a hi-res graphics program for displaying frequency distributions.

Stats has some statistical functions in common with Personal Software's new *VisiTrend* program, and the different approaches to the user are interesting.

The *VisiTrend* manual seems to assume that the user just emerged from thirty years in the jungle and goes into copious detail on every facet of the Apple and the program. About the only things missing are a decent index and an explanation of electricity.

Stats, on the other hand, presumes the user is so knowledgeable that any except the barest descriptions of the functions are superfluous. This treatment of the manual limits use of the program to Apple users who are knowledgeable about their equipment and knowing in the folkways of statistics.

But for the user who has the requisite bodies of knowledge, *Stats* returns value received several times over. For ordinary sizes of samples and data, the program is prompt in sending completed analysis to the screen and makes it easy to get hard copy versions or to make files for use with the graphics package.

Statistical operations provided by *Stats* are mean and standard deviation, frequency distribution, two-sample correlation and regression, T-tests, chi squares, and data transformations. Correlation matrices can also be generated for data subsets.

User notes appended to the explanations of each program option help amplify the manual further by briefly explaining program limitations or explaining the formats of results.

Stats worthily fills a void that's existed in the Apple market. ART

HSD Stats, by Stephen Madigan and Virginia Lawrence, Human Systems Dynamics (Northridge, CA). $99.95.

Epson MX-100. From Epson America. No printers have so captured the Apple market as the two handy Epsons, MX-70 and MX-80. Left out in the cold were those who need 132-column output on wide paper—the Epsons are small, accommodating only eight and one-half inch paper or less.

Now the door's open and there's a warm fire in the fireplace with the introduction of the MX-100—the same printer extended to take fourteen and one-half inch paper, with several pluses: tractor feed plus platen for stationery; extended or emphasized text plus graphics. And all the standard good things about Epson: easy use, descending descenders, super low price.

During two months of printing a very large mailing list (yes, ours), printing constantly for nearly forty hours each time, a sample MX-100 proved outstandingly reliable. Occasional extra line feeds, a bane that has plagued the project, were nil with the 100. Normal tasks were a piece of cake.

All the Epsons are dot matrix printers that print at a speed of sixty lines (no matter what length) a minute. Dot matrix, regardless of the character set, is not considered "letter" quality. Less than forty years ago, letter quality meant handwriting—the typewriter was too impersonal. Considering the present rate of progress, it should be a very short time until good-looking printer output is considered just as acceptable as typewriter type.

The MX-100 has software or DIP switchable regular and condensed type in normal mode and expanded type in expanded mode. Leading can be set to six or eight lines per inch and is programmable to other settings. Requires Apple interface card. MCT

Epson MX-100 by Epson America, Torrance, CA. $995.

MARKETALK

Impressions

☐ **Solitaire.** By Art Carpet, Computek (Canyon Country, CA). The name couldn't be more straightforward and neither could the game. Carpet has programmed four games of patience accurately and without hoopla. Klondike (often erroneously called Canfield), casino-style and regular, comprises two games. The other two are Pyramid and a rather obscure but addictively frustrating one called Picture Frame.

The games are done carefully and thoughtfully; they run smoothly and fast. If you remain fascinated with the infinite variations of the magical fifty-two, you might as well let your Apple do the shuffling and dealing. $29.95.

☐ **Handwriting Analysis.** By Eliot Lipps, Micro Lipps (Santa Monica, CA). No, your Apple isn't going to look at a sample of your handwriting and spew forth a right-on and revealing

thumbnail sketch of your character. But it will do the next best thing.

It's best to have specially prepared handwriting samples from your subjects when you try your first analyses. Then you can compare them by word and letter against the many diagrams that make this program as accurate as possible. Your answers to these illustrated questions about the handwriting samples are what the program uses to compile character sketches, which you can choose to print out or see on the screen.

Then the fun is seeing how close the Apple got—and secretly wondering if that off-the-wall statement that you're athletically talented means that you might have been Tracy Austin or John McEnroe if you'd only kept it up.

Handwriting Analysis is all in fun—or is it?—and great for a party or office lunch activity. $19.95.

☐ **The Shattered Alliance: The Chronicles of Osgorth.** By John Lyon, Strategic Simulations (Mountain View, CA). When you purchase *The Shattered Alliance,* you're actually getting eight games in one. The basic game ties together four fantasy wars and four historical wars (loosely speaking; the fourth is from the history of Middle Earth). The fantasy games, however, are the main thrust of this package. For the first time in SSI wargaming, magic plays a role, at least in several scenarios. Characters include beasts, dwarfs, elves, and their ilk.

The thrust of the games is strategy; all scenarios are battles waged on a map. You can choose either a strategic map, which shows a large area and is best for overall planning, or a tactical map, which is a closeup of the area of battle. Both maps are in colorful hi-res graphics; on the tactical map, the characters are clearly individualized and fun.

Strategic hopes that this will be a game for a more general audience than war gamers alone. With no complex ships, planes, or space vehicles to design or at least understand, and with individualized groups of warriors, their wish is apt to be fulfilled. $59.95.

☐ **Santa Paravia and Fiumaccio,** by Murray R. Summers, Instant Software (Peterborough, NH). Veteran Apple owners who remember the Apple program *Hammurabi* will find similarities in *Santa Paravia.* But *Hammurabi* was a solitaire game; *Santa Paravia* accommodates as many as eight players, and while *Hammurabi* could create or destroy you on a whim with a plague, *Santa Paravia* almost entirely depends on you to seal your own fate.

That's not to say *Santa Paravia* is predictable. On the contrary, how the underlying lord of the game will judge your actions can be surprising; but it will be logical.

This is primarily a text game and not at all new. But it arrived on our doorstep direct from the author with a plaint that no one had ever seen fit to give his game ink, and would we just try it. We did and, while it is not a game to light fires with, we found it enjoyable playing—especially with a group of people; a family would do nicely.

Each player selects a name and is given the title of either sir or lady. Each is then assigned a principality, presumably European, to rule. If you rule well, by the game's standards, higher and higher titles will be bestowed upon you. The goal of each player is to be the first to become king or queen. Santa Paravia and Fiumaccio are merely the names of the principalities always assigned first.

Each principality is agricultural at the start and has a number of serfs, clergy members, soldiers, merchants, and aristocrats equal to each other province. You must manage your area, determining how much grain and land to buy or sell, how much grain to allow your serfs, what to charge in various kinds of taxes, how to empty the government coffers.

Spending the revenue is fun. You may choose to build markets or woolen mills, erect palaces and cathedrals, or develop an army. You may also determine the moral (and morale) tenor of your principality through the severity of its justice system. Once during each turn, a hi-res picture map shows your holdings and purchases.

If you are doing what your subjects like, you will be awarded a new title occasionally. But don't be carried away just because your current title is higher than anyone else's; other players may change their ways and pass you by. Or someone may suddenly emphasize arms and armies and wipe you nearly off the board.

This game is not full of bells and whistles, but it is an enjoyable pastime; and, by virtue of its offering play for many people at a reasonable price, it is a fine library piece for a family who likes to enjoy their Apple together. Five skill levels. DOS 3.2. $19.95.

☐ **Stellar Action.** By Frank Heffner, Mytopia Gameware Institute (Sioux City, IA). Mytopia Gameware Institute decided to fight the inflationary spiral in the entertainment marketplace with this $19.95 entry. It was a worthy thought, but it's not yet clear whether there are enough budget-conscious Apple owners around to reward such conscientiousness.

Stellar Action is a variation on the original *Star Trek* game; it uses the same text graphics but changes the scenario to head-to-head combat between two participants. This is a duel to death between two evenly matched spaceships with no artificial time limit to call a halt to the proceedings.

You can choose either the Kolbans or the Rodnocians as enemy for the Environ Protectorate ship. You also have the alternative of playing against the computer or against a friend. That means you're getting four games for the price of one.

Unfortunately for Mytopia, that may not be enough. Apple owners still seem to be buying bells and whistles, no matter what the price, and *Stellar Action* is a simple program with sound gaming values but not much pizzazz.

The question that *Stellar Action* will eventually answer is not whether author Frank Heffner can program a worthy game, but whether the Apple market has matured to the point where gaming value means as much or more than programming oneupmanship among the various software authors. Cassette, 32K; disk, 48K. $19.95. ◨

The *pfs*: software series

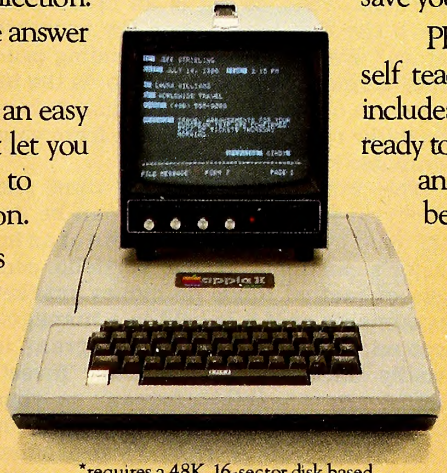
A Personal Information Management System.

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

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

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

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



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

PFS and PFS: REPORT come with simple self teaching manuals plus a support plan that includes program updates and factory experts ready to answer your questions. And all of this at an affordable price. Each program is priced below $100.

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

PFS is a trademark of Software Publishing Corporation.

APPLE* is a registered trademark of Apple Computer, Inc.

Software Publishing Corporation

Buttonwood Apples

BY KEN LANDIS

In the earliest years of our nation's history, when Manhattan north of 14th Street was rich farmland, investment was an individual but fast growing activity. Regular investors met to trade under a shady buttonwood tree on Wall Street. When, in May 1792, under that tree, these people drew up a compact of trading practices by which they agreed to abide, it was called the Buttonwood Agreement. Twenty-five years later, many of the same people created a more formal association based on the same principles; they called it the New York Stock Exchange. The tree stands in spirit, commemorated by a plaque. And, of course, by the Exchange.

Investors have three major concerns in regard to their portfolios. The investor asks: Am I realizing a profit? Do I have timely data to make informed investment decisions? Is my portfolio evaluation system supplying me with all the data I need for investment, tax, and income decisions?

There is a plethora of programs available for the Apple that can supply some or all this information. The majority of these programs may be classified by their primary function: portfolio accounting and analysis; individual stock and commodity technical analysis; and data retrieval.

By reviewing the software available, we'll provide you, the investor, with an accurate profile of these programs and their usefulness in helping you achieve your investment goals.

As investors, we all have one common goal—to make money! In pursuit of this goal, we're faced with making many decisions. First of all, is an investment worth the risk based upon the probable return? We may apply this guiding principle to financial software also. Is the investment of capital and time in software worth the return it will provide from your computer system?

The wisdom of investing time and money in financial software can be determined by the size and breadth of an investor's portfolio. If the portfolio consists of mutual funds or money market instruments or if it is held in a managed account, using a financial program is usually unnecessary. For these investments, any financial program on your Apple will be of little value to you for other than simple profit and loss calculations. For the mutual funds in your portfolio, your best bet is a charting program that compares your portfolio's performance against standard indicators such as the Dow Jones composite averages. For money market funds, choose a similar program using the prevailing average treasury bill rate for comparison.

If your portfolio is varied, you need not only profit and loss figures but current technical and statistical data relating to the components of your portfolio. This is when using your Apple with an appropriate financial program becomes a wise decision.

Ken Landis is president of Interface Associates, a New York City computer consulting firm.

Rather than spending night after night charting highs, lows, closes, and volumes on graph paper, you may now, with very little effort, direct your Apple not only to poll the required information from a remote data base but also to calculate and chart for you. You'll save a great deal of time and effort, and you'll have investment information at your finger tips to help you make more timely, better-informed decisions.

The number of ways you can use your Apple to help manage your portfolio is limited by three factors: the commercially available software for this purpose, your system's configuration, and your own knowledge of programming and investing. The limitation placed on you by the available software is obvious: if what you want to do isn't done in any commercial package, you'll have to write it yourself. You can do only what the programs you load into your computer can do, whether they're store-bought or home-brewed.

Using your Apple for investment decisions requires a certain minimum system configuration. Most analysis and accounting programs are written in Apple-

soft because of Applesoft's floating point capability, but older programs and some charting programs use Integer Basic. So an Apple II with both languages is recommended. Accomplish this by using either an Apple language card, the Language System, or a 16K RAM expansion card.

Two disk drives are better than one. For the Apple to complete its job successfully, it usually must have access to both a program disk and a data disk. The program disk contains software that directs the operations to be performed on the data, which is, as you probably surmised, on the data disk. If only one disk drive is available, these disks will have to be continually swapped, not only slowing computations but increasing the likelihood of disk damage and I/O errors.

A modem allows your Apple access to remote data bases such as the Source, CompuServe, and Dow Jones to retrieve stock prices, volume information, and instant news about specific securities and commodities as well as news about the marketplace as a whole and world events as they affect the marketplace. Without a modem, you'd have to collect and enter this information manually. This is time consuming, and, often, data gathered manually is less accurate than information collected by a data base.

Last but not least, you'll need a printer to provide you with hard copy. For many people, information is much easier to digest and work with when read from a printed page than it is from a CRT. Also, if you have a printer with graphics capability, you can print hard copy of your charts.

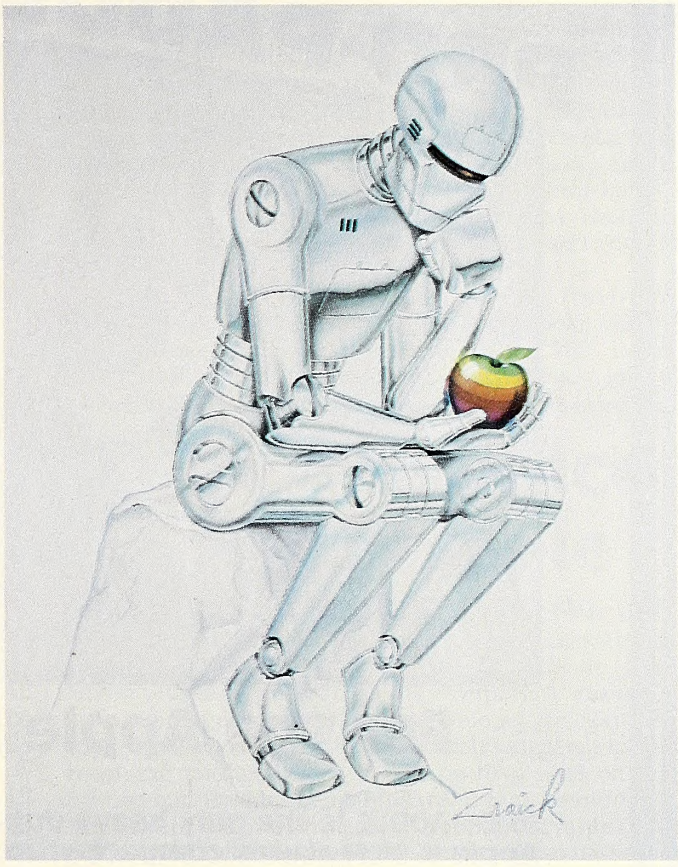
This configuration along with the cost of the software packages you choose may turn out to be a fairly hefty investment. If the costs of this system are prohibitive, don't be discouraged; do as much as you can with the Apple system you have. Our reviews will tell you the minimum configuration you need to use each package correctly.

If your gains—either pecuniary or psychic—outweigh the costs, you can easily and happily justify your expenditures. Buttonwood Apples will be appearing occasionally—if irregularly—in *Softalk*. Software items will be reviewed in their entirety, from documentation to performance to vendor support. Packages with similar purposes will be grouped for review in individual articles.

We'd like to encourage your comments, including requests for reviews of any specific financial software package. There are many packages that are sold only regionally. If you know of such a package and feel that it should be reviewed, please let us know.

The next article in this series will cover the first step in an investment decision: gathering information. We'll review not only the packages that allow us to gather information, but the sources themselves, including Dow Jones, the Source, and CompuServe. ◨

contemplating a byte



For one full year, many of you have been wondering how long we would continue sending you *Softalk* free without trying to put the touch on you for something, whether a subscription, software, peripherals, kidney beans, defective grommets, or spare Edsel parts. Now comes the magic moment.

Softalk commissioned graphics artist Robert Zraick to do August's cover with a poster in mind. The robot contemplating a bite is evocative both of Rodin's *The Thinker* and the Genesis passage on the Garden of Eden . . . not to mention the possible significance to our favorite technological fruit.

The artist and *Softalk* are sharing in the profits from the poster. *Softalk* will distribute its proceeds to individuals developing Apple tools to help the handicapped. *Softalk* guarantees 100 percent distribution of its monies.

In addition to the posters, which are being sold at $6.00, (plus $1.50 to cover shipping and handling), two hundred artist's proofs, signed by Robert Zraick, are available at $75 each.

The size of the poster is 24 inches by 34 inches. The artist's proof will be hand-numbered and hand-signed and be accompanied by a certificate giving its number and guaranteeing that only 200 are being distributed.

Robert Zraick's art will grace any computer room, and your purchase will help others become more self-sufficient.

Orders may be sent to:

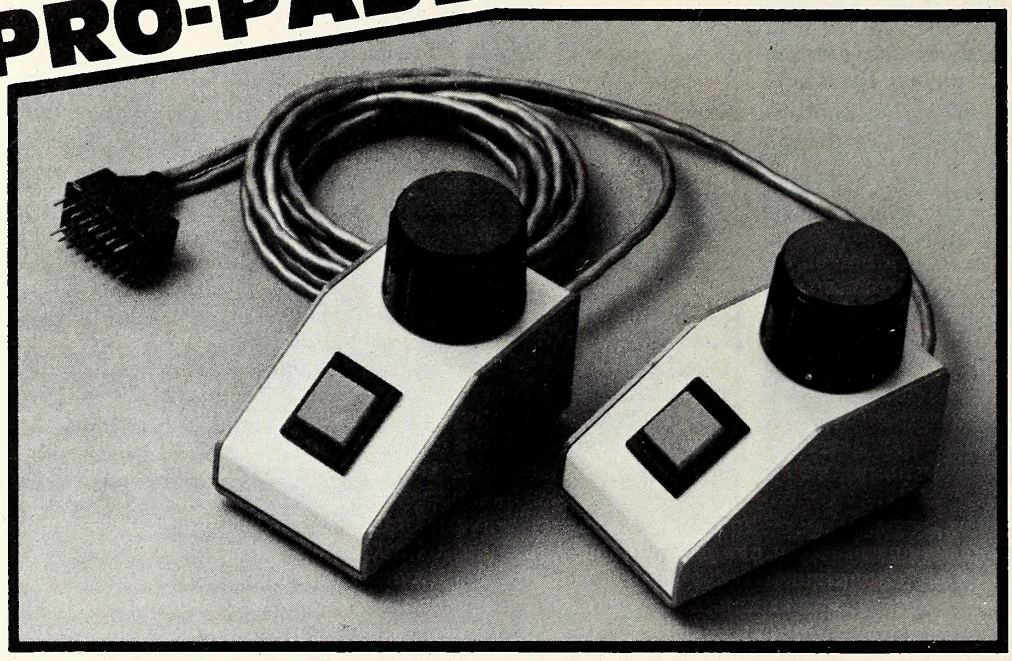
SOFTALK

Softalk Poster
11021 Magnolia Boulevard
North Hollywood, California 91601

Dealer inquiries invited.

Rainbow Computing, Inc. Announces!

PRO-PADDLE



For the Apple II Computer

PRO-PADDLE is the only heavy-duty paddle available for the Apple II. It features compact sturdy metal construction, long-life switches with large buttons and tactile feedback, high accuracy paddle movement, shielded coaxial cables, and a molded plug. These paddles are designed and built by Computerworks and are constructed of the highest quality materials and workmanship available.

End your paddle problems for good! $39.95

Dealer Inquiries Invited

RAINBOW COMPUTING INC.

Mail Order Dept. ST9
19517 Business Center Drive
Northridge, CA 91324
Telephone: (213) 349-0300

Terminal Data

BY JEFFREY MAZUR

When you turn on your Apple and begin typing, you are talking to a built-in operating system. Whether it be DOS, Applesoft, or the machine language monitor, this system will try to decipher what you key in and perform some desired function. This is the operation of a computer.

When you want to talk to another computer, however, it's usually necessary to have a terminal, which is simply a keyboard and some form of display or printer. A modem then connects the two systems via a phone line. Your Apple has a keyboard and a display so it should not be too hard to make a terminal out of it, and, with the proper interface, you can connect the Apple to the phone line. But there's one more step.

As you talk to the other computer system, it would be disastrous to have the Apple's operating system trying to understand what you type. Most of your dialogue with the other computer wouldn't make sense to the Apple, which might keep beeping at you or shouting "SYNTAX ERROR . . . SYNTAX ERROR." Therefore, you need a small program that will allow everything you type to be sent out and all incoming data to be displayed on the screen. This is a basic terminal program and something like it is included in the firmware of most communication cards for the Apple. This program transforms the Apple into what is sometimes called a *dumb terminal.*

Term Programs Use Apple's IQ. If you take offense at having your Apple called dumb, you have every right. Here is a very sophisticated machine being used to do nothing more than input and output information. It certainly has the power to do more, but, as always, it requires software to do it. Along come the *smart terminal programs* for the Apple.

Basically, these programs occupy a relatively small amount of memory space, so whatever RAM is left over can be used as a buffer. A buffer is used to capture data as it comes in, so, even after a message scrolls off the screen, it is still retained in the computer and can be recalled to the screen. A buffer can also be used to prepare a message to be sent while off-line—that's not yet connected to the other system—and then quickly transmitted when on-line.

Since the bulk of the data sent is in the form of readable text, it's also desirable to have upper-case and lower-case capabilities. Many Apples already have a lower-case adapter installed, so a terminal program should know how to handle lower case.

More important, because the Apple keyboard cannot generate lower-case characters, some software scheme must be used to act like a shift key. In most cases, the escape key performs this function: hitting ESC once capitalizes the next letter only, hitting ESC twice in a row sets a shift lock, and hitting it again returns you to lower-case entry. This is not too bad considering the limitation of a shiftless keyboard. At the same time, some of the missing ASCII characters can be added to the Apple keyboard through the use of multiple keystrokes. While you're at it, add some keyboard macros—whole strings of characters that can be output by typing only a few keys.

Some of the more fancy programs will keep track of the phone numbers of the various systems you call. Then, using your modem's autodial feature (if it has one), the program can call and sign you on. With all these functions and more, it's sometimes necessary to have a complete command menu for setting up parameters. One of these commands invokes the actual terminal routine, and you're ready to communicate.

Having identified the nature of terminal programs, let's review some available for the Apple. Special attention is given to the copy protection, if any, used by each program. This is particularly important with this type of software for two reasons. First, these disks will get a lot of use, probably every time you use your modem. Second, there's a lot of new hardware coming out that will work with these programs. Being able to list and modify a program may be the only ways to configure your system the way you want it.

TSC Terminal Program. The *Terminal Program* from the Telephone Software Connection provides desirable features at a low cost—$35. Like all software from TSC, it can be purchased by telephone and downloaded directly into your com-

The Post Office in Your Apple

"Teach your little Apple big-time electronic mail." Therein lies the implied claim that Microcom makes about their new software package, *Micro-Courier.* Electronic mail, if you haven't already guessed, is the computer's answer to writing out a message, putting it in an envelope, and slapping a stamp on it. Instead, information is electronically sent over wires to a receiving station where it can be displayed on a screen or printed out. Besides being quicker, it may be more reliable and less expensive in some cases. Another advantage is that any type of information—not just letters—can be sent from one computer to another. Moreover, sending the same information to a thousand people can be just as simple as sending it to one.

Unlike other electronic mail systems, which require dedicated phone lines or logging on to a large time-sharing network, *Micro-Courier* transforms your Apple into a complete communications terminal. You can establish your own network with two or more terminals connected to standard phone lines. And, of course, when the Apple isn't being used for electronic mail, it's free to perform a multitude of other tasks.

The operation of *Micro-Courier* is quite simple. Basically, it deals with three items: mailbox IDs, which are two-digit numbers representing those to whom messages will be addressed; files, which can be any normal DOS 3.3 text, program, or binary files; and messages, which are defined as instructions to send particular files to given mailboxes at specified times. The user of *Micro-Courier* would create messages typically during the daytime and queue them up for subsequent transmission at night. The system can be unattended when the transfer takes place.

Hardware Requirements. *Micro-Courier* is primarily an Apple-to-Apple, *Micro-Courier*-to-*Micro-Courier* communication system. Each location to be linked requires the following minimum configuration: Apple II Plus with 48K of RAM or Apple II with 48K of RAM and Applesoft or language card; disk II with sixteen-sector controller; Hayes Micromodem II; standard telephone line; *Micro-Courier* software. Also highly rec-

puter. If you prefer, they will mail you the program on disk for an extra $5 change. There is no copy protection, so you're free to make backups as necessary.

This program supports a buffer for received information only. All input from the remote system is stored in a buffer while you are on-line. Therefore, the only editing that can be done is the deletion of unwanted dialogue. The buffer contents can be viewed (in variable speed) or saved to the disk at any time. Ten keyboard macros can be defined by changing the appropriate lines in this Applesoft program.

The *Terminal Program* has a unique auto-log-on feature that memorizes the steps necessary for connecting to any remote computer system. Whenever you call a new system, just log on manually the first time. The *Terminal Program* will record your responses in a disk file so it can sign you on automatically in the future. Of course, you can always take over manual control or change the log-on procedure if necessary.

ASCII Express II. For $60 you can add automatic redial, individual macro files for each system, and improved file transfer capabilities with *ASCII Express II* from Southwestern Data Systems. This program stores the name and phone number for up to eighteen systems in a menu-driven format. Each system can also have a separate set of keyboard macro definitions to help in logging on and general use. With just a few keystrokes you can call up and log on any system. If the line is busy, or for some other reason you're unable to get through, the program can keep redialing the number as many times as you wish. You can also ask for an alarm (five control-Gs from the Apple) to signal when it gets through.

The *ASCII Express* buffer can be used for sending information as well as for capturing incoming data. Simple editing commands are included for preparing messages to be sent. Any DOS file can be sent, including Integer Basic, Applesoft, text, and binary files, with the help of some utility programs supplied.

Two forms of uploading are provided: character-at-a-time and line-at-a-time. In the character mode, the program outputs each character and then waits for the receiving computer to echo it (assuming full duplex) before sending the next one. The line mode allows an entire line to be sent, then it waits to receive a *continue* signal from the other computer. This signal will be in the form of a single character (usually a prompt character) that can be defined ahead of time.

For example, consider the problem of sending an Applesoft program from one Apple to another. The receiving Apple would be returned to the Applesoft interpreter and the NEW command performed to remove any existing program. Then the sending Apple, also running under Applesoft, could simply type LIST to have the program print out. At the same time, each character would be transmitted to the other Apple where it would appear as if someone were typing in a program.

After each carriage return, the Applesoft interpreter looks at the line entered, tokenizes it, and stores it in memory. This takes a finite amount of time, which gets longer as the program grows (you've no doubt noticed this while entering a long program). During this time, if characters are still being transmitted, they may not be accepted into the system.

This is like a pie-making assembly line where the pies are coming in just as fast as you can decorate them. If you're asked to perform an additional task, like placing each one in a box, some pies are bound to end up on the floor.

Teaching Apple Manners. One solution to this problem is to have the sending computer stop after transmitting each line. It then waits until the receiving computer displays (and send back) the Applesoft "]" prompt signifying that it's ready for more.

This is how you would transfer a program with *ASCII Express:* first the program is converted into a text file (more on this later). Then the text file is loaded into the terminal program buffer. You then type S for Send a File, select line-at-a-time mode, and specify the] character as the ready signal. By the way, this dynamic form of two-way transmission is called *handshaking* and the particular method used is sometimes referred to as a *protocol.*

Along with the terminal program come several utilities that enhance the system's capabilities. In fact, these programs are useful in and of themselves even when you're not using *ASCII Express.* Two of them convert Integer Basic and Applesoft programs into text files and another does the same for binary files. This is necessary because the terminal program deals only with text printable characters), and both Basic interpreters store programs in a tokenized, shorthand form. Another program, *Spacecrunch,* removes extra spaces and punctuation from text files such as those created by the other two programs. This can shorten a file by as much as 35 percent with similar savings in disk space and transfer time.

The last program is a special copy routine that makes backup copies of the *ASCII Express II* as well as of any other normal DOS diskettes. However, this program keeps track of how many copies of *AE II* you have made and will only let you make four. Like the original, each copy cannot be duplicated by standard copy programs, and, of course, the copies themselves don't have the copy utility on them. This is certainly one of the most reasonable approaches to the software piracy problem we've seen.

Data Capture 4.0. From Southeastern Software comes *Data Capture 4.0,* which sells for $65 (eighty-column versions are $90). The disk is not protected, so backups aren't a problem. Although it's a fine terminal program, many extras are lacking despite its higher price. For example, there is no storage of frequently dialed numbers—you must type in the number each time you want to call. On the plus side, this program has a unique feature that allows the buffer to be saved on the disk automatically whenever it fills up.

While the program is running, a status display is constantly maintained on the top three lines of the screen. Thus you can tell at a glance how many lines are in the buffer, whether the buffer capture is on, what the carrier status is, and so on. While this is sometimes handy, it takes up several lines and can be distracting. The escape key is used to call up the

main command menu so another key must be used for case shifting: the right arrow. If you have a hardware shift-key modification, it will also work. Since thirty-column versions are available, this program may be a better choice for someone with that hardware.

VisiTerm. Part of the new *Visi* series from Personal Software, this one lives up to its name. Like *VisiCalc,* this is a well-planned, comprehensive package with many innovative features. One is the hi-res sixty-character display it provides on a regular Apple. Suffice it to say that *VisiTerm* is rather expensive ($129) and uncopyable; no information was given on backups, but presumably they are available for a fee.

Before getting into all wonderful things that *VisiTerm* is, here are a few things that is isn't. It isn't a memo dialer—that is, it does not keep a list of systems and phone numbers. Worse yet, when you enter the number you wish to dial, you have to do it *very* slowly, waiting for each digit to be dialed before entering the next. Although this nuisance stems from the Micromodem II, it's something a good terminal program should alleviate. Also, take warning that you'll need a high-quality monitor to use the hi-res display. Color sets, even with direct video input, will probably not work well.

Many of *VisiTerm*'s features relate to its unusual display, so it deserves close examination. Whenever you are in the terminal mode, the text display you see is actually being generated by the Apple's hi-res graphics. This immediately accounts for *VisiTerm*'s ability to display upper and lower case and multiple fonts without any extra hardware. This has been done before on the Apple, for example, in the Mountain Computer ROMPlus+/*Keyboard Filter* combination. But Personal Software extended this idea by allowing variable width characters so small letters, like a lower-case *i,* would take up less room than a capital *E.* This makes it possible to fit more than forty characters on a line, answering one of the biggest complaints about the Apple.

Controlling this display is a vast set of variables that can be changed through the options mode. Two complete fonts are available at any given time; you can choose these two from several on the disk or create your own. The options menu allows you to view any character, in either font, and make changes if desired. It does this by showing a large image of the character with simple controls to turn on or off any dot within the character cell. Each character can also be programmed to sound the bell at one of three volumes whenever it is displayed. Also defined in the options mode are the cursor width, vertical spacing between lines, tabs, and key tick volume (for those of you who like audible verification that a key was pressed). Even the point at which the text will scroll and how much are user controlled.

VisiTerm allows a wide range of protocols for sending text. It even has a keyboard type-ahead buffer for times when the receiving computer has put your transmission on hold. Programs can be sent after the usual conversion to text files, and files generated by the rest of the *Visi* series are compatible. A small status line at the bottom of the display keeps you informed of various conditions; with the default fonts, it's displayed using tiny capital letters that make it readily distinguishable from the text.

Summing Up. Of the four terminal programs examined here, there's a clear conclusion. The price of the Telephone Software Connection program makes it attractive for most people who need the basic functions. The *ASCII Express II* and *Data Capture* fulfill the need for a somewhat more elaborate system. For those who can afford it, *VisiTerm* has the most to offer, as long as you don't mind manual dialing.

All the programs assume you are working in DOS. For Pascal and CP/M users, similar packages are available. You might also be interested in the *Micro-Courier* and *Micro-Telegram* software from Microcom. These allow you to set up an automatic electronic mail system or access Western Union, TWX, Telex, and so on.

Where are terminal programs taking us? One significant direction leads to electronic mail. Along this line, consider the system described in the companion article on page 77. ◻■

Post Office

from page 77

ommended are a clock/calendar board, a printer with interface, and a second disk drive (up to four are compatible).

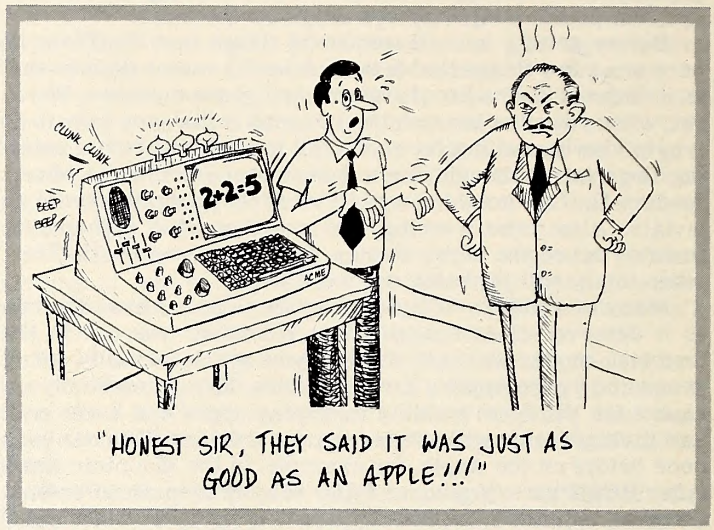
Although the system presently works with only the Micromodem II, the Novation Apple CAT and other autoanswer, autodial modems will be compatible soon. Most printers should work with *Micro-Courier*, since you can specify the type (parallel or serial), slot, and maximum line length in the configuring portion of the program.

The total system actually consists of several programs that are often called from the disk and swapped into memory. Thus you must have the *Micro-Courier* disk in the drive almost all the time. However, any DOS 3.3 file can be sent as a message and any 3.3 diskette can be used for incoming mail storage; thus the recommendation for two drives. Another important factor with this type of operation is that of backup ability. *Micro-Courier* offers backups at $50 apiece.

The Courier Speaks English. *Micro-Courier* is menu driven with simple English prompts. Many commands invoke further submenus and the nested levels are constantly displayed at the top of the screen. You always know where you are and how you got there. The escape key is used to return to the previous menu.

Sending a message might go something like this. From the main menu, you would select "Mailbox Directory Maintenance," assuming you've never sent this person any mail before. After a few seconds, you get another menu allowing you to add, delete, modify, or list mailbox IDs. At this point, you can also define lists of individual mailbox numbers that can be lumped together for future bulk mailings. To add a new mailbox ID, you must select an unused number (one hundred are available), and then supply the name and phone number of the person who is to receive this mail. Provisions have been made for long phone numbers, including access pauses that may be required on some phone systems. Following this, the escape key is used to return to the main menu. To write simple letters, the *Micro-Courier* text editor can be used by selecting "Create/Edit Mail Files." The editor is simple and easy to use, and when you're through, it saves your letter to a file.

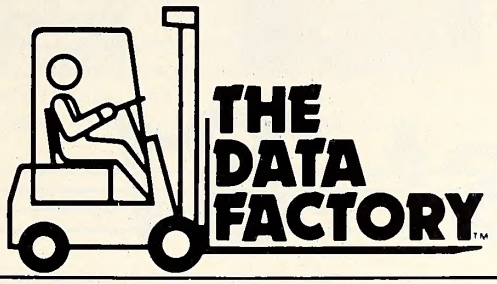
Returning again to the main menu, you are now ready to use the heart of *Micro-Courier*. "Review/Address Outgoing Mail" is selected, followed by "Send Message to Mailbox ID." Here you supply the number of the destination mailbox. The computer verifies your choice by printing out the receiver's name and then asks for the name of the file you wish to send. This could be a letter created by the editor or any DOS 3.3 file including programs, data, *VisiCalc* files and so on. The next query asks when you wish to have the message sent, and, with a clock card, you may enter some time late that night for the lowest cost. Now that a message has been created and entered into a list with possible others, all that remains is the actual sending. Back to the main menu one more time, now to "Send/Receive Mail." This is the mode in which you can leave the computer ready to receive incoming messages and to transmit your messages according to specified times. If the system is unable to get your messages through (if, for example, the phone is busy), it will automatically try again every fifteen minutes—up to ten times.



When you come back the next day, the display will show how many messages were sent and received. You can then review incoming mail or check the status of mail sent (the program will notify you of any mail that it couldn't send). *Micro-Courier* also has a terminal mode in which you can interactively communicate with other computer systems.

Things to Come. *Micro-Courier* is a very comprehensive electronic mail system, but it is only the beginning. Microcom also has another software package that allows the Apple to tie in to Western Union, Telex, and TWX services. These two packages could meet most of the communication requirements of many businesses.

Just as important are the inroads they make toward the acceptance of personal computers. This will have an even greater impact in the home, where existing services such as computerized banking and information retrieval will soon be joined by shopping, advertising, teaching, and innumerable other capabilities. In fact, the future of electronic mail is so bright that political battles have already begun over jurisdiction for national service. Obviously, the United States Postal Service has a lot at stake here. Software such as *Micro-Courier* is indicative of the microcomputer's coming of age. We are nearing the point where the lure of a novelty is giving way to the necessity of tools or appliances. Although the quality of software has lagged behind hardware sophistication, it's catching up fast. And this is what it will take to make predictions of a computer in every home and office a reality. ■



EASY-TO-USE

Get ahead and stay there. The Invoice Factory and Data Factory are a winning combination for any office. The Data Factory keeps information organized and accessible. Find the information you need in seconds. According to a recent study, after only half an hour with the manual, a new operator (who has never used a computer!), can begin to input data. No other data base manager is this easy to understand and use.

LARGE ENTRY CAPABILITY

You can use the total capability of the Apple computer. Up to 88 categories (fields) for information storage can be selected and each one may contain as many as 239 characters. These limits are larger than any comparable product on the market.

FLEXIBILITY

You can change your mind. Append additional fields for information storage at any time. Transfer any part of your records to form a new file. Add or delete characters in a field when more or less space is needed. No need to re-enter any data again. Even if you have a damaged record from power surges or other circumstances, you can reconstruct the usable data. The program is the most flexible one currently available for the Apple micro computer — This is where its power lies.

NEW RECEIVABLE SYSTEM

Use the account data from The Data Factory and transfer it to The Invoice Factory to print invoices, statements, receivable reports, or sales analyses of products quickly, accurately, and effortlessly. With only an account number entry, all pertinent information on an account is automatically before you — shipping address, terms, contact, etc. Choose the product code and enter the amount ordered. Automatically the numbers are extended, tax and handling charges computed and UPS freight rates totaled. An audit trail and product sales analysis is available from the original order entry. You always know how much outstanding money you don't have, how current sales compare to last months, and the total of everything a customer has purchased within any given time period with this complete accounts receivable system.

CONSUMER CONCERN

You can bet on Micro Lab. It is a name that you can rely on. Our products have been sold to thousands of corporations. Our Extended Warranty, back-up program copies, and customer service policies are the standard in the small computer industry. When you decide to buy a Micro Lab product, you can be assured that our system will work.

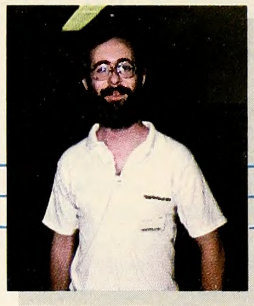
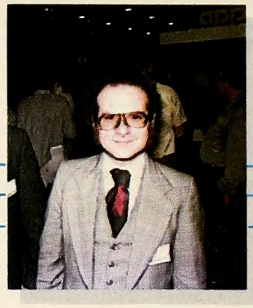
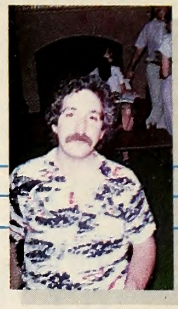
EFFICIENCY PAYS

For a minimum investment in this inexpensive computer system, your entire office will function smoothly and efficiently. By using The Data Factory and The Invoice Factory with an Apple computer, the time saved in data retrieval and sales analyses will soon pay for the entire cost of the system. Large corporations need not tie up their main computer with specialized office procedures and record keeping — you can place one computer in each department, or tie a branch office into your system with a modem. The Data Factory retails for $150 and The Invoice Factory is $200. For further information about Micro Lab products, contact us directly or go to your local Apple computer dealer for a demonstration. You can't afford to be without our products.

micro lab
systems that work

2310 Skokie Valley Road • Highland Park, IL 60035 • 312—433-7550



applefest '81











Industryites from all over the country flooded Boston for Applefest 81 in the first part of June. Some of them, from left to right, were: top row organizer Jonathan Rotenberg of the Boston Computer Society; Dan Fylstra, Sunnyvale, California, chairman of the board of Personal Software, who was returning to the birthplace of his company; Dan Bricklin and Robert Frankston, Boston, authors of VisiCalc, Personal Software's most popular program; and Mitch Kapor, Boston, author of VisiPlot and VisiTrend, two of Personal's newest offerings. Second row: Mark Pelczarski, Chicago, director of Co-Op Software and author of the Complete Graphics System; authors of Stoneware's DB Master, Barney Stone, Oakland, and Stanley Crane, Alpine, Colorado; Mary Carol Smith, Eugene, Oregon, Avant Garde Software. Third row: Lynn Busby was in from Computer Station in St. Louis; Greg LeClair, Los Angeles, represented Axiom's printers; Herschel Pilloff, Oxon Hill, Maryland, demoed H&H Scientific's software; Mike Weinstock from Nibble magazine; Tim Hartley of Hartley Software in Michigan. Bottom row: Quality Software's Bob Christianson from Los Angeles; Art Malin from silicon gulch's Santa Clara Systems; Gary Reinhardt of Washington's Highlands Computer; and Evan Scharf of Berkeley, California, vice-president of Information Unlimited Software.

First Time's Charm

BY PETER OLIVIERI

Probably every Apple owner has fantasized about being locked in a room with an Apple, lots of software, new attachments and equipment, books, and plenty of applications—graphics, music, games, business programs. . . .

Except for the locked door, those who attended Applefest '81 in Boston on June 6 or 7 had their dream come true.

The lines began to form early in the morning of the first day with current and potential Apple owners eager to see nearly a hundred exhibitors at the show.

One of the real pleasures of Applefest was that it was Apple specific. We had an opportunity to look at, for example, several word processing programs—all of them for the Apple! Just about every time we stopped at a booth for information on a product or service, we were talking with people who were Apple users themselves.

Indeed, there was something for everyone. At the show, we spoke with some elementary school kids interested primarily in games; a priest looking for a data base to help keep parish records; two nuns interested in computer-aided instruction; a retired couple looking for ways to supplement their income; and a host of others who were interested in the Apple for personal and professional use.

There were a few new products on display: there was a nicely written electronic mail package called *Micro-Courier* (Microcom, Boston, MA), an impressive word processor called *Executive Secretary* (Personal Business Systems, Minneapolis, MN), the Terrapin Turtle (Terrapin, Cambridge, MA), to name just a few.

Applefest was particularly successful in the presentation of two special features. The first was a series of excellent seminars, all of which were quite well attended. Lecture topics included *VisiCalc*, electronic mail, turtle robots and Logo, the adventure game cult, software for small businesses, and the future for personal computers.

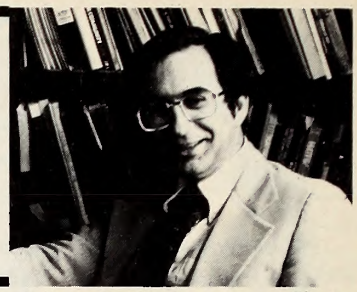
Of particular interest to most attendees was the second special feature of Applefest, the Hands-On Rooms. These were sponsored by Apple Computer and were quite popular. One room had more than twenty Apples, complete with programs, for visitors to experiment with. A second room had nearly twenty Apple III systems along with an interesting forty-five minute demonstration of the III's capabilities. There were always lines stretched down the corridors waiting for a chance to use the machines. Apple also presented a continuous seven-minute video tape about Apple computers.

Kudos to Jonathan Rotenberg and the Boston Computer Society's Apple User Group, sponsors of the show, for the professionalism and success of Applefest.

Watch out, Octoberfest!

Mind Your Business

BY PETER OLIVIERI



Last month, we described some steps that must be followed in developing a data base. Now let's look at how those steps were followed in a particular business, which we'll call E.Z. Day Inc. In this case, the data base was to be used for controlling inventory levels. Following the steps in developing a data base should be worthwhile even if this application isn't relevant to your business.

Establishing the Objectives. The first step is to determine the objectives of the new computer application. E.Z. Day identified seven objectives: (1) to reduce the cost of maintaining data, (2) to increase the speed with which data is processed, (3) to develop control systems to maintain product inventory levels, (4) to maintain an up-to-date inventory vendor file, (5) to provide planning reports to management as needed, (6) to reduce duplication, and (7) to cut down on data errors.

What Reports? With the objectives settled and clear, it's time to anticipate and design the reports that will be required to fulfill the objectives and disseminate data. As soon as detailed report formats have been specified, the records in the data base can be designed. Several meetings among the senior and work center managers at E.Z. Day yielded designs for required inventory control reports:

Report A. Daily sales by item number.
Report B. Items whose quantity on hand has fallen below the item's reorder level.
Report C. Actual order forms for items whose quantity on hand has dropped below the reorder level.
Report D. Current information about particular products.
Report E. An alphabetical list of all vendors.
Report F. Mailing labels for all vendors.
Report G. Listing by vendor of the amount of each vendor's items sold to date.
Report H. Value of current inventory in detail.
Report I. All items for which there is no stock on hand.
Report J. All items ordered more than one week ago and not yet received.

What Data? Knowing the reports that would be generated, E.Z. Day could determine the data needed in the file for each report. The requirements were:

Report A. product number, product description, current day's total sales
Report B. product number, product description, reorder level, amount on hand, and amount on order
Report C. product number, product description, reorder level, amount on hand, amount on order, vendor name, vendor address, vendor city and state, vendor personal contact, amount to order, order date, item price
Report D. depends on the final record contents, but would no doubt include vendor information, product number, product description, amount on hand, amount on order, price, and so on
Report E. vendor name
Report F. vendor name, address, city, state, and zip code
Report G. vendor name and amount of that vendor's items sold to date; perhaps item price
Report H. amount on hand and price for each item
Report I. product number, product description, amount on hand, amount on order
Report J. product number, product description, date of order, vendor name and address, vendor personal contact, vendor telephone number

Look closely at the unique data required to produce reports. Note that a record for each item in the inventory file includes a number of fields, and each field's length follows the field identification.

Field	Name	Length in characters
1	Item number	4
2	Vendor name	25
3	Vendor address	25
4	Vendor city & state	25
5	Vendor zip code	5*
6	Vendor phone number	10
7	Personal contact	15
8	Item description	15
9	Reorder level	5
10	Reorder quantity	5
11	Quantity on hand	5
12	Quantity on order	5
13	Date of last order	6
14	Latest sales entry	5
15	Total sales for today	7
16	Total sold year to date	7
17	Item price	6
18	Special comments	25

*Perhaps 9 for new zip codes.

Determine the length of each record in the data base by multiplying this length by the number of different items to determine how much storage space will be needed to maintain the inventory file in the data base.

Hardware and Software Needs. In this case, procurement of computer hardware preceded these steps, but it's not unusual to use this kind of information to choose a computer system. The E.Z. Day company obviously needs a sprocket-fed printer to accommodate a variety of special forms such as mailing labels and order forms. In addition, the list has clarified the company's disk storage requirements.

The right software for this company must allow ease in entering initial information about each inventory, easy updating of any field in a record as needed; and ease in adding new records. In addition, several reports must be produced, so some reporting capability is required. User-written programs, a purchased package, or a combination can help.

What Has To Be Done? To produce the required reports, it's necessary to manipulate data. After looking over the list of inventory reports, the managers determined these operations were required:

1. Sorting before printing (by vendor name, zip code, or year-to-date amount purchased, for example).
2. Subtracting the latest sales entry from the quantity on hand to get the new quantity on hand.
3. Multiplying price times quantity on hand for each item and totalling to get the value of the current inventory.
4. Adding quantity received to quantity on hand to get new quantity on hand.
5. Adding latest day's sales to total sold year to date to get new total sold year to date.
6. Comparing quantity on hand with reorder level to determine reorders.
7. Comparing date of one week ago to date of last order to determine overdue items.
8. Comparing each quantity on hand with zero to determine what's out of stock.
9. Adding latest sales entry to total sales for each day to get daily updated total sales.

Some operations (numbers 2, 4, 5, and 9) must be performed at the time the data is entered. Others (1, 3, 6, 7, and 8) are done while printing a particular report. Obviously, the software must be able to perform these operations.

Building the file. The company then began using the data base management system selected most appropriate for the task. Using the *create* option of a particular package generated a screen format to represent information needed for a single inventory record. All of the existing information about each item was then entered. Note that this presumes the availability of reorder levels, reorder quantities, and year-to-date figures by item number (among other things). Initial file creation can be quite time consuming.

Once the file was built, it was not long before the users became quite familiar with all of the various file maintenance operations such as adding a record, changing the value of a particular field, and sorting the entire file. At this point, the development of the reports began.

Before using the DBMS features to design printed output, each report had to be laid out in detail. Report titles, dates, headings, footings, and column arrangements all had to be carefully specified. In some instances, the command language of the DBMS could be used to produce a listing; producing other reports may require more detailed steps.

Managing. At the E.Z. Day company, a good deal of work had to be done after the design of the inventory file, the entry of data, and the report preparation. It is here that the probable success of a particular application can be greatly increased. Among the remaining tasks were determination of the frequency of all reports, their number of copies, their distribution, and their retention period; the concurrent operation of two systems (the current one and the newly computerized one) until the new system has been thoroughly tested; the training of personnel responsible for daily data entry; preparation of a user's guide for the computer application; and development of backup procedures and procedures for ensuring data integrity and data security.

While indeed, E.Z. Day faced much work, the rewards that resulted from a smoothly functioning, well-managed system were well worth the efforts.

DBMS Packages. This month we'll look at another data base management package: *Datadex.* Next month, this column will include a summary chart of DBMS packages, rating them on a variety of factors.

Datadex. The *Datadex Interactive Business Management System* was developed by Information Unlimited Software, Berkeley, CA. We reviewed the first edition, May 1981.

Like most data base management systems, *Datadex* allows the user to create files, enter data, create reports, and search the file for specific records. In addition, there are a few special features allowing you to write your own programs using the data created by *Datadex.*

To use *Datadex,* you need an Apple II Plus (or the Apple Language System) with at least 48K memory, DOS 3.3, and a disk drive. Of course, a printer is most helpful, and additional disk drives make the operation of the system even simpler.

The first section of the *Datadex User Manual* begins by listing definitions of some of the key terms that will be used throughout the text. The definitions and the glossary at the end of the manual help ease the reader into using the package.

After an overview, the first chapter orients the user to the control keys (primarily ESC and CTRL) that are used to interact with the menus. The manual includes actual screen images as illustrations—a feature that enhances the text and makes it easier for the user to understand what's taking place.

The next chapter guides the reader through the sign-on procedure, which automatically lists all the peripheral devices attached to your system—another helpful feature. The user can add peripherals to this list and make changes as necessary.

The chapter continues with a discussion of the features of the program's main menu and utility menu. The main menu allows you to enter data; find a particular record; print reports; print forms; and exit. Reports and forms differ in that

reports are in columnar format and forms, such as mailing labels or checks, are not.

The utility menu lets you create files; create reports; sort records; initialize diskettes; and reconfigure your system.

Furthermore, you may add to the utility menu any programs you've written that interact with the files in the data base. This feature is not often found in DBMS packages.

The next section discusses in detail how to create files. As you begin, your diskette's catalog is displayed. If you are creating a file for the first time, a cursor appears on the screen, allowing you to design your input form. After naming the fields and specifying their characteristics (field type, field length, etc.), enter data using the on-screen form you designed. Another feature not often found in DBMS is the capability of changing fields, changing their order, and inserting new fields. Furthermore, one of the options allows you to take one of your existing data base files and easily create a new one with some of the same data but with fewer fields or perhaps with the fields in a different order.

This section of the manual also covers designing a zip code field. The developers of the program allowed for the new *Zip + 4* format proposed by the United States Postal Service. You can save money if you use mailing lists if *Zip + 4* becomes widely used. This speaks well for future-mindedness of *Datadex*'s creators.

The next chapter describes how data is entered. The typical features of adding, deleting, sorting, and retrieving records are explained in detail. The search feature is interesting in that it permits searching forward from the current record, searching backward from the current record, doing a reasonable phonetic search, and searching for possible transpositions of numbers in certain fields. Also, the *browse* feature permits a scan through the file's various speeds. An accumulator is available to create subtotals and totals as desired.

The next section discusses report design. The user can create columnar format to be displayed in the final report as well as headings and footings. The package allows user-developed columns where information can be printed that is calculated at the time of printing but not normally part of the data base. There are a variety of arithmetic operations that can be performed on any two fields to create a third field for printing.

The manual also describes commands for printing reports, printing forms, sorting the entire file, finding a particular record, initializing a diskette, and reconfiguring the system.

The manual then guides the user through specific *Datadex* features. This chapter also includes a fairly detailed section on how to use data created by *Datadex* as input to a user-written program. Look for this feature when you choose a DBMS. The final section of this chapter lists all possible error messages with an explanation of their cause.

The Datadex User's Guide includes a reference card listing all commands named in the text. As you become familiar with the package, the card becomes a very valuable resource.

The Readers Speak. "I recall that when I was in college, we had a computer course that used a service that we dialed up over the telephone. You connected the telephone hand set to a computer terminal and then had access to a wide variety of programs and computer services. While I know that my Apple is a stand-alone microcomputer, I was wondering (1) if there are any such services available to the microcomputer user, and (2) whether I can use my Apple to connect to such a service? Wouldn't it be nice to get stock reports over your Apple?" P.C., Houston, Texas

The answer to both of your questions is yes. These services do not require a microcomputer; they merely require a terminal. However, microcomputer owners are certainly not excluded. In fact, these services expand the capability of your Apple with a great deal of additional computer power.

You'll have to add some hardware to your Apple, consisting of a board to insert in the Apple that connects to a device called a *modem*. The modem translates computer signals to telephone signals and vice versa. Cost is about $300.

Subscription services that supply modem users with whatever information is of particular interest to them include The Source, 1616 Anderson Road, McLean, Virginia 22102, which provides a daily compilation of world news from the *New York Times* as well as the ability to do a keyword search through UPI stories. The Source also gives quotations for stocks, bonds, commodities, treasury bills, mutual funds, options, and precious metals; and financial service, provided by professional economists. Library programs that include cash flow analysis, lease-versus-purchase evaluation of amortization loans, portfolio management, financial modeling, statistical analyses, games, and data base management systems.

The Source also provides access to airline schedules, hotel and airline reservations, electronic mail, and *Datapost*, which allows users to send messages electronically to any address in the country to be delivered as hard copy the next day.

There's also a "barter" network to allow you to offer or purchase goods and services with other businesses and a personnel search network for finding new employees.

You can order a printed document, request a research report on a particular topic, or gain access to more than a thousand information services. There are many other services like dining, real estate, sports, and showtime information.

To become a subscriber, you must first pay a $100 registration fee. Then you're charged monthly according to the amount of computer time you've used. The minimum is $10 monthly. The rates for computer time are $15 per hour on weekdays from 7:00 a.m. to 6:00 p.m. and $4.25 per hour after business hours and on weekends. If you store your own programs or data on the Source, you are charged fifty cents per 2,000 characters per month. You also pay any telephone charges that are due you. One warning, you'll be surprised at how quickly those hours of use add up. There is so much available that you can get carried away! ❑■

SDS GIVES YOU ALL YOU NEED!

APPLE-DOC

We are proud to announce an updated version of our best seller, **Apple-Doc** in DOS 3.3. This versatile package consists of three powerful reference utilities; **Variable Cross-Reference** which creates a table of very variable and the line number on which it occurs, **Line Number Cross-Reference** creates a table of every program lined called by a GOTO, GOSUB, etc., and all lines called from, and finally, **Constant Cross-Reference** creates a table of all numbers used in the program not listed by the above tables. Also included are **Replace**, a global replacement editor and **Lister** which produces a completely professional program listing. Additionally, the new 30 page tutorial manual makes **Apple-Doc** quick and easy to use. Whether novice or professional **Apple-Doc** more than pays for itself in time savings alone!

Price: $49.95 on disk

a.c.e.

Applesoft* Command Editor, or **A.C.E.**, combines a large number of the most useful programming utilities into one co-resident package. Co-resident means these routines become a working part of the Apple II*, thereby making **A.C.E.'s** various commands and utilities immediately available WHILE you're entering or editing existing program lines. It's functions include cross-referencing individual variable names, displaying current values of all variables, and information on memory and diskette status. For machine language programming there is a hex-decimal converter and all monitor commands can be executed directly from Applesoft. User defined macros allow strings to be input with a single keystroke and a powerful line editor makes editing easy and greatly reduces program development time. **A.C.E.** is a winner!

Price: $39.95 on disk

List Master

List Master, our latest addition, provides a number of utilities for performing large scale changes to your program listings in a high-speed and automatic way. **List Master** includes **Applespeed** which can remove REMarks, shorten variable names, combine lines, and renumbers by 1's in any Applesoft program. **Smart Renumber** not only renumbers but also gives you the option of preserving logical blocks of line numbers you have established for various routines thereby retaining the original logic and workability in the renumbered program. Also included are merge routines that allow you to quote blocks of lines from one part of the program to another and automatically renumber appropriate line references. **Comp-List** compares any two Applesoft or Integer programs and lists any lines added, deleted or changed.

Price: $39.95 on disk

All programs require 48K and Applesoft in ROM or language card. Specify DOS 3.2 or 3.3. Calif. residents add 6% sales tax.

*Apple II and Applesoft are registered trademarks of Apple Computer Co.

SDS

southwestern data systems

P.O. Box 582-S • Santee, California 92071 • (714) 562-3670

Hard Times Are Coming

BY DAVID HUNTER

The business world is like a battlefield. Woe to the business that hasn't kept up with the latest arms development.

The microcomputer is one of the best means for zapping the problems of growth that plague new or small businesses.

Many smaller businesses are aptly served by a single microcomputer running on a standard five-and-a-quarter-inch or eight-inch floppy disk system. But there comes a time when you have many microcomputers; then, the cost of software and peripheral devices, like printers, modems, and data bases, becomes very high—unless you can share them among computers. Often, sharing entails its own inconveniences; you could buy several printers, for instance, and still have to lug them between stations.

Who Knows the Mailing List Woes? Some tasks cry out for a hard disk. For instance, if you have a growing mailing list, you'll probably find it gets a bit tiresome when you need 153 disks to store all the names and addresses. The amount of time—labor and computer time—required to service the files may be costly. Also, floppy disks are less than perfect, their imperfections occasionally causing frantic hair pulling and desk kicking.

Three things—mass storage, speed, and reliability—as well as the ability to hook up many computers and peripherals in a network, are what make hard disks and hard disk systems a boon to a business. You can have fifteen times as much storage space and get to it ten times as fast for three to five times the cost of a standard floppy disk system. "My opinion is that five grand is a lot," says Randy Hyde, software author and hardware designer, "but, when you see how fast those suckers run, it's worth it."

Hard disks come in three sizes: fourteen-inch, eight-inch, and five-and-a-quarter-inch. They all utilize something called Winchester technology. Several years ago, engineers at IBM came up with the idea of mass information storage using a new kind of disk drive. The idea was to get thirty megabytes of data on thirty tracks. They noticed that the numbers 30-30 were the same as the popular Winchester rifle. IBM never got around to making the 30-30 model, but the name Winchester stuck.

Essentially, Winchester technology hard disk drives differ from floppies in three ways: the read/write head never touches the disk; the disk is fixed in the sealed atmosphere of the drive; and the disk is not removable (though there are ways to get around this).

The drive consists of an aluminum platter with a fine coat of oxide (440 microinches thick) acting as the storage medium. Most disks are double-sided and some have room for several disks, each with its own read/write head. The drive spins the disk at about 3,600 r.p.m. with little or no side-to-side or up-and-down movement. By comparison, a standard floppy drive spins at 360 r.p.m.

One of the keys to Winchester technology is getting the read/write head as close as possible to the disk surface without touching it. At present, on most drives, they're separated by a distance about the same width as a smoke particle. Since almost anything bigger than a hydrogen atom can cause a head crash, the drive must be sealed, and most current models have a fan that ensures clean air inside the drive.

Habits are hard to break, and some people may be concerned about not being able to remove the disk. It's true that if the disk fails, the whole drive has to be taken to the shop—leaving you out of luck for a while. The obvious solution is to have a backup system. There are several possibilities for providing backup. Floppy disks, streaming tape, cartridges, and a second hard disk drive are some options. Still another option is a fixed/removable configuration in which half the data is stored permanently in the drive and the other half is on a removable cartridge.

As an Apple owner in business, you have a variety of hard disk systems to choose from. The first and foremost criteria for choosing a hard disk system are your storage requirements. The difference between five-and-a-quarter-inch, eight-inch, and fourteen-inch is the difference between five megabytes, ten megabytes, and twenty megabytes. Of course, you can have a lot more than twenty megabytes with multiple drives.

Santa Clara Systems. A typical eight-inch rigid or hard disk subsystem interfaceable with the Apple is the SCS-10 from Santa Clara Systems (San Jose, CA). Formatted at ten megabytes, the SCS-10 delivers data ten to twenty times faster than floppies with low power consumption. No belts and pulleys and longer mean time before failure (about ten thousand hours) make reliability not much of a concern. The system also comes with error correction coding, which detects errors in the transmission of data and corrects them automatically.

President of Santa Clara Systems Thomas Quinn believes the eight-inch has some advantages over the five-and-a-quarter-inch. In cost, proportionally, the eight-inch drive is like the family economy-size hard disk system. You get twice as much for less than twice the cost. Even so, Santa Clara Systems hopes to market a five-and-a-quarter-inch hard disk by the end of the year. Clearly, the ability to draw the dividing line between five-and-a-quarter-inch drives and eight-inch drives comes with experience.

Cameo. Cameo Electronics (Anaheim, CA) uses removable disk packs and cartridge drives (found for years in minicomputers) in their DC-500 subsystem. Utilizing either the Control Data Hawk drive or the Western Dynex Series 6000 drive, the DC-500 stores five megabytes in the drive and five megabytes on the removable cartridge.

"The difference between drives is the difference between a Lincoln Continental and a Cadillac," says sales manager Kristin Dunton.

Morrow. Morrow Designs (Richmond, CA) is a company that sells disk drives of both kinds, floppy and hard. Bob Ance, director of marketing, believes that "the hard disk's time has come. A lot of people out there are running out of capacity." Although they are working on a controller of their own for the Apple, it is possible to use a Morrow drive with the Konan controller, newly arrived on the market. Ance wants the Morrow controller to get away from Apple's DOS, which he believes slows down interaction time, but he admits, "It's going to take a while to get to know the Apple."

XCOMP. Another company considering interfacing Apples with their hard disk systems is XCOMP (San Diego, CA), whose president, John Costello, thinks the Apple market is peculiar. "How many Apple owners are really going to spring for a hard disk system?" he wonders. There is also the problem of software traps. He's looking for a software ace to author programs that would be compatible with Apples.

Costello also believes some hard disk advertising is misleading. He says users get the wrong impression about the simplicity of the system, thinking that a floppy interface is all you need to hook up a hard disk system.

"This is totally wrong. The main issue in hard disks is speed, and floppy controllers don't come close."

It's a tough market and a tough technology, with much of the electronics once found in drives now in the controllers. The ratio of companies offering disk drives to companies offering controllers is about four to one.

Konan Corporation. The David is a series of hard disk con-

trollers available from Konan Corporation (Phoenix, AZ). With error correction, flaw mapping, format command, and a single high-speed, bidirectional port, the David allows you to control most currently available five-and-a-quarter and eight-inch hard disk drives. National sales director David Pippenger believes now that costs are coming down, many first-time customers are going for hard disks instead of floppies. He attributes this to Winchester technology and its record for data reliability, which most businesses can't do without.

Lobo. Hard disk memory systems from Lobo Drives (Goleta, CA) have their own built-in floppy backup system. Technical director Arlene Hazen says the units are now at test sites where they "have been getting a good reponse." Randy Hyde of Lazer Systems used a Lobo to write Pascal drivers for the Lobo Apple controllers and was "very impressed with the speed" as well as the general performance level of the system.

A medium-sized business that uses several microcomputers and assorted peripherals might do well to consider the networking system. Early network systems were in starlike configurations with a hard disk drive connected to a multiplexer that in turn connected to a fixed number of computers and peripherals. The problem with these early networks was that, if a single component like the multiplexer, went down, it would affect the whole network. This problem has been solved with the emergence of Nestar's Cluster/One and Corvus's Constellation and Omninet.

Clusters of Nestars. As many as sixty-five microcomputers can be connected to the Nestar (Palo Alto, CA) Cluster/One Model A. Using sixteen ribbon cables to connect the components, the Cluster/One allows any topology for distances up to a thousand feet. In other words, you can have a station on the fifth floor, one on the third floor, and several on the fourth floor, all sharing the same hard disk drive and the same printer. There can be direct station-to-station communication without a central multiplexer or controller and error correction from station to station.

Mike Chase, sales manager of Nestar, predicts that developing interfaces to mainframes is the next thing to look for in hard disk systems. "People need access to greater computer power at their own desks," Chase explains. "The automated office concept is here. We want to make it so that the user doesn't have to think much but has access to mainframe computer power."

Chase also points out that the Nestar systems have true security. If users don't have access to a file, they can't break in. This is a feature that many businesses and schools look for with the same fervor used in seeking data reliability.

Constellations of Corvuses. Corvus Systems (San Jose, CA) offers two different network systems, Constellation and Omninet, as well as Mirror, a video cassette backup system. Constellation is for smaller network systems confined to a single room, while Omninet can be spread throughout a building. Corvus technical writer Dan Torres believes hard disks are perfect for creative or special applications, though some software applications keep them from becoming as sophisticated as some mainframes. In the future, Torres believes, we'll see gateways from network to network and a lot more software making the leap from microcomputer to hard disk system.

Saratoga Springs for Apples and Corvuses. One satisfied Corvus user is the Saratoga, California, School District, which has two Constellation systems hooked up to seventeen Apple computers and a printer. Administrator Marion Kenworthy has only good things to say about the system. Currently, one of the Constellations is hooked up to fifteen Apples in a classroom. Although the system is primarily used to teach an introductory course in computer science, Kenworthy foresees many applications in different subjects in the near future.

"Eventually, it will get to the point where we'll have two people on social sciences, two people working on foreign language exercises, and five people working on computer languages—all at the same time, all using the same disk," explains Kenworthy.

A problem that only time may overcome is the dearth of software for the Corvus. Kenworthy says there isn't enough because companies are worried about copying. One thing she hopes will alleviate this problem in Saratoga is the school's strict adherence to copyright laws. Although it may be different in other schools, Kenworth says, "there is no illegal copying here. We are trying to teach our kids honesty."

For Now, They're Strictly Business. Hard disks and hard disk systems are obviously of little use to the average home microcomputer owner. Unless you had a book collection the size of a library that you want to keep (not just catalog) on disk, you'd rarely need all that storage space. But, for many businesses, a hard disk system is much better than numerous floppy disks.

Deciding just what system is best for your business is difficult, but certain elements remain important for all hard disk systems. A backup system is a necessity, although it may get little use. Error correction seems standard these days, and it is a handy thing to have. Avoid networks that rely on a central controller or multiplexer because, if one fails, it affects the whole system and could lead to the great-grandfather of headaches. Finally, it's advantageous to purchase your system through a dealer knowledgeable about hard disk systems.

Although expensive at the outset, hard disks and networks ultimately save a lot of money. Even if there are several people trying to use a printer at once, for instance, you can get spooling devices that, essentially, back order the demands as they're received then produce printouts in turn, without waiting for fresh commands. Instead of needing several printers or having your one printer hauled all over the building, you need only one stationary printer.

It isn't getting any easier to survive on the big battlefield out there. If you're plagued by the small size and various quirks of the floppy disk system and need a lot more storage space, a hard disk system is for you. Talk about your requirements to companies that produce the systems and take the time to see the systems demonstrated. Remember, "Five grand is a lot of money, but when you see how fast those suckers run. . . ." ■

Turn your Apple into the world's most versatile personal computer.

The SoftCard™ Solution. SoftCard turns your Apple into two computers. A Z-80 and a 6502. By adding a Z-80 microprocessor and CP/M to your Apple, SoftCard turns your Apple into a CP/M based machine. That means you can access the single largest body of microcomputer software in existence. Two computers in one. And, the advantages of both.

Plug and go. The SoftCard system starts with a Z-80 based circuit card. Just plug it into any slot (except 0) of your Apple. No modifications required. SoftCard supports most of your Apple peripherals, and, in 6502-mode, your Apple is still your Apple.

CP/M for your Apple. You get CP/M on disk with the SoftCard package. It's a powerful and simple-to-use operating system. It supports more software than any other microcomputer operating system. And that's the key to the versatility of the SoftCard/Apple.

BASIC included. A powerful tool, BASIC-80 is included in the SoftCard package. Running under CP/M, ANSI Standard BASIC-80 is the most powerful microcomputer BASIC available. It includes extensive disk I/O statements, error trapping, integer variables, 16-digit precision, extensive EDIT commands and string functions, high and low-res Apple graphics, PRINT USING, CHAIN and COMMON, plus many additional commands. And, it's a BASIC you can compile with Microsoft's BASIC Compiler.

More languages. With SoftCard and CP/M, you can add Microsoft's ANSI Standard COBOL, and FORTRAN, or Basic Compiler and Assembly Language Development System. All, more powerful tools for your Apple.

Seeing is believing. See the SoftCard in operation at your Microsoft or Apple dealer. We think you'll agree that the SoftCard turns your Apple into the world's most versatile personal computer.

Complete information? It's at your dealer's now. Or, we'll send it to you and include a dealer list. Write us. Call us.

MICROSOFT
CONSUMER PRODUCTS

Microsoft Consumer Products, 400 108th Ave. N.E.,
Bellevue, WA 98004 (206) 454-1315

VENTURES WITH VISICALC

BY CRAIG STINSON

This month we continue our survey of *VisiCalc* utility programs with reviews of the *Context Connector* and *Versacalc! 16*.

The *Context Connector*, from Context Management Systems (Palos Verdes, CA) is a combination terminal program and file converter. With the proper hardware, it will allow you to download data from a remote computer and convert it into *VisiCalc* models. You can also use it to send files from your Apple to another computer.

The novelty in this is the file converter. There's no shortage of programs to facilitate the capture of information from remote data bases. Personal Software's own *VisiTerm* will do the job nicely. But the *Connector* goes an important step further by reorganizing standard DOS text files into something that can be loaded into *VisiCalc* directly with an /SL command.

Here's how it works. When you run the conversion module, you're first asked for the name of an input file, which can be any text file. The program fetches your file, asks for an output name (for safety's sake, something different from the input name), then displays line one. At this point you have a number of options.

You can use the converter as a text editor, if you wish. The program will display your file one line at a time and allow you to scroll through it in either direction or go directly to a specifiefd line number. You can also search for any pattern of characters—either through the entire file or through a specified number of lines. You can make changes to the data if you wish.

Or you can convert the file to a *VisiCalc* /SS file. The main thing the *Connector* needs to know to do this is how you want to parcel up your data into *VisiCalc* cells. One way to do it is to specify a column width of *n* characters and let the *Connector* begin a new cell every *n*th character of your input file. Chances are that's not going to be what you want, since your data will likely have short items and long items and will not be padded out to any uniform field width. In many cases, if you just try to break up the file into *n*-character groups, you'll wind up with half a number in one *VisiCalc* cell and half in another.

A second option is to specify the column width of your *VisiCalc* sheet and let the *Connector* scan the data and break it up into whole fields. Label fields will be divided up according to the specified column width. Numeric fields, if larger than the column width, will remain larger in the *Connector*'s parceling and will be truncated by *VisiCalc* when they're loaded onto the spreadsheet.

You also have the option of specifying your own cell division formats and saving those formats to disk. The manual suggests you do this for recurrent situations that don't fall comfortably within the *Connector*'s standard procedures.

Some kinds of mainframe systems send negative numbers with the minus sign following instead of preceding the number. The *Connector* has a utility to handle this kind of aberration. Unfortunately, however, the *Connector* does not recognize scientific notation; any field with an alphabetic character in it, including something like 4.5E10, will, on conversion to *VisiCalc* format, appear as a label field. That's troublesome, because it leaves you, the user, with the job of rekeyboarding—just what you were trying to avoid in the first place.

Getting the data parceled up correctly for the spreadsheet is, to be truthful, not always a simple matter. But there the *Connector* gives you more help. When you finally send the stuff off to be converted, the *Connector* will let you inspect its work line by line, if you wish; if something's not right you can interrupt the process, fix, and send it back on its way.

Lots of defaults are provided to save you keystrokes in the conversion process. If you regularly tap the same data source, you can store a default name for the input files; likewise, if you're frequently funneling information into the same *VisiCalc* sheet, you can name a default output. You can even specify a default format file for the parceling specs. Should you need to interrupt the conversion process, the *Connector* will remember at which row of the *VisiCalc* file to resume, and so on.

We should point out that the conversion process is row oriented; and you don't have any control over that—except that if you have the sixteen-sector version of *VisiCalc* you can use it to rearrange rows into columns, by way of the Data Interchange Format.

Once the data conversion process is done, just boot *VisiCalc* and call up your output file with a /SL.

Besides allowing you to convert files from remote data bases into *VisiCalc* form, the *Connector* can assist you with the consolidation of separate *VisiCalc* sheets. If you want to transplant three lines from sheet A and four lines from sheet B onto sheet C, you can save the appropriate portions of A and B into /PD or /PF files, and then, with the *Connector*,

consolidate and convert them back to /SS files. The final consolidation onto sheet C takes place by way of a standard *VisiCalc* overlay.

This ability to consolidate separate *VisiCalc* forms is especially handy for those with *VisiCalc* 3.2. For 3.3 users it's not clear that this method offers any advantage over sending data out to DIF files and loading back into the target worksheet.

The other half of the *Connector* is a communications package, with facilities for uploading and downloading text files. You can use it as a transmitter as well as a receiver. The system requires the Hayes Micromodem or a modem that runs with the Apple Communications Card or the SSM/AIO board. For Micromodem users, the *Connector* also provides a modifiable autodial list.

The *Connector*, by George Lilly, Adam Schneider, and Craig Vaughan, retails for $150.

Did you ever do a /SL and give the name of a /PF or /PD file by mistake, instead of a normal *VisiCalc* file? You probably got some plaintive remarks from your Apple speaker and some disintegrated matter on your screen.

The Print to File or Print to Disk option (the two commands are synonymous in *VisiCalc* 3.3) is useful for transferring *VisiCalc* data to other programs, but information so stored cannot be loaded back onto the *VisiCalc* screen.

But try this. Boot *VisiCalc*. Type /GFR and /GC20. Now set the cursor movement to vertical and enter the following, as labels, from A1 to A6. Be sure to precede each entry with quotes, so that it will appear as a label.

```
>B24
/GC20
"VISICALC IS EVEN
"MORE INTERESTING
"THAN YOU THOUGHT
"IT WAS!
```

For A3 to A6, you'll need to hit the quote key twice, once to specify a label and the second time to make the quotes appear on the screen. Now send this off to a /PD file, clear the worksheet, and load the /PD file back onto the screen, just as though it were a normal /SS file.

Aha. You've just created the *VisiCalc* equivalent of an exec file.

If you've got sixteen-sector *VisiCalc* and a printer in slot one, try this: load any smallish /SS file, turn on the printer, and type /SS,S1 (no file name needed). What you'll get is a printout of the commands that *VisiCalc* writes to disk when it saves your file; when you reload the file from disk, these are the commands that *VisiCalc* has to execute to re-create your spreadsheet on the screen.

You can mimic these commands by putting them into labels, aligning them flush right, and saving them in /PD files.

Aurora Systems (Madison, WI) has researched and experimented with this aspect of *VisiCalc*, and the result of their work is an enormously interesting and cleverly assembled package of utilities called *Versacalc! 16*. As the title suggests, the package is intended for use with *VisiCalc* 3.3.

The first novelty about *Versacalc* is that it comes on a two-sided disk. The flip side, to which the manual first directs your attention, holds an elaborate tutorial on three subjects: the various conditional features of sixteen-sector *VisiCalc*, the storage of *VisiCalc* commands in /PD files, and the creative use of screen overlays. Since the first of these subjects is covered poorly by the *VisiCalc* manual and the second and third are ignored altogether, the tutorial alone may prove to be worth a good portion of the purchase price.

Its worth increases if you read between the lines and study the way it was made. The tutorial consists entirely of *VisiCalc* files. To run it, you first boot *VisiCalc*, then load the file *tutorial* from side two of the *Versacalc* disk. At various points in the tutorial you're asked to load in new *VisiCalc* files; the way these files modify the screen actually illustrates the principles the tutorial teaches.

Not all *VisiCalc* commands will execute from a /PD file; or at any rate, the people at Aurora Systems say they haven't figured out yet how to make all of them work. They invite owners to experiment along with them and share their discoveries.

Side one is a set of utilities to help you manipulate your *VisiCalc* worksheets in various ways. The utilities, while written in Applesoft, also employ the principles of altering worksheets by means of /PD files.

For example, there's a program that will sort an /SS file, using a column or portion of a column as the sort key. You first load the file you want to sort. Then you save, as a /PD file, the column or part thereof that you want to be your sort key. This can be either a set of numbers or a set of words, and the sort that results may be either ascending or descending, at your option.

Having saved your /PD file (which must be named SORT), you boot *Versacalc* and call up the sort utility. After you answer a few questions, the program goes into your /PD text file, reorganizes it in the manner you have specified, and also embeds in it the commands necessary to make *VisiCalc* reorganize your entire worksheet according to the sorted column. When this is done, you go back to *VisiCalc*, reload your original worksheet, do an /SL and load in that /PD file named SORT, and presto—the rows of your *VisiCalc* file have been rearranged, with the values in the sort key column in order as specified.

This is just a sample of the ingenuity of *Versacalc*. Other utilities assist you in the creation of /PD files to perform specific kinds of manipulations on your *VisiCalc* data, like fixing values in place (replacing formulas with values), replicating labels from row to column or vice versa, moving values from row to column or vice versa, inserting rows or columns, blanking out areas of the screen, and so on. There's also a set of utilities called accumulators that accumulate year-to-date values without causing circular references.

Finally, there's a set of file manager programs. One of them will look at your *VisiCalc* data disk and create a /PD file that, when loaded, displays your catalog on the *VisiCalc* screen. Other modules lock, unlock, or delete files. There's also a print file option that will run off your commands and data on the printer. This last item does essentially the same thing that the /SS,S1 option on *VisiCalc* does except that it can be applied to /PD as well as /SS files.

Versacalc! 16 appears to be for two kinds of *VisiCalc* user—the experimental and curious who wants to learn much more about the behind-scenes aspects of *VisiCalc* and the person who needs to perform complicated rearrangements of *VisiCalc* data. The program is by Jerrold H. Bents, Thomas A. Farin, and L. Darryl Mataya and retails for $100. ■

LETTER PERFECT T.M. LJK

WORD PROCESSOR Apple II, and Apple II Plus*, DOS 3.3., 40/80 Columns

MAIN - MENU
CURRENT DRIVE
NUMBER #1
SLOT #6

→ Editor ←
Change Drive #, Slot
Load
Save
Merge
Screen Format
Printer
Lock
Unlock
Delete
Format Disk
Reconfigure
Data Base Merge
Quit

Press ← or → to move cursor
Press (Return) for selection

40 Column
80 Super-R-Term
Smarterm
Videx
Bit 3

USE: HAYES MICROMODEM II T.M.
TO SEND FILES

EASY TO USE: LETTER PERFECT is a character orientated word processor with the user in mind. Fast action (machine language), menu driven, single load program. 34K Free. Requires Paymar LCA, one disk drive, and 32K memory. One time configuration for your system, printer type etc., can be reconfigured at any time. Right hand justification. Supports incremental spacing, underlining, boldface with NEC or Qume/Diablo. Will use any of the special print characters of your specific printer. Key codes make mnemonic sense for easy use. All text packed during saving for greater disk storage capacity. Full typewritten page of buffer space for easy manuscript editing. Menu driven printer selection, or use your own special print driver. Works with DOS 3.3. "Screen format allows you to preview printed text".

All this and more, for $149.95.

Features:

FULL CURSOR CONTROL
Home Cursor
Scroll Page Forward
Scroll Page Backward
Pause Scroll
Scroll Line at Time
Scrolling Speed Control
Move Cursor Down
Beginning of Text

MULTIFUNCTION FORMAT LINE
Standard Formats a Default
Formats Easily Changed
Right Justification
Left Margin
Page Width
Line Spacing
Lines Per Page
Font Changing
Set Page#
Top Margin
Bottom Margin

Delete a Character
Insert a Character
Delete a Line
Insert a Line
Headers and Footers
Shift Lock and Release
Global and Local Search and Replacement
Underlining and Boldface
Automatic Centering
Horizontal Tabs
Special Print Characters
Split Catalog
Page Numbering up to 65535
Prints up to 256 Copies of Single Text File
Non Printing Text Commenting

FUNCTIONS
Delete All Text
Delete All After Cursor
Delete All Before Cursor
Delete Next Block
Delete Buffer
Move Next Block to Buffer
Add Next Block to Buffer
Insert Block From Buffer
Merge Text Files

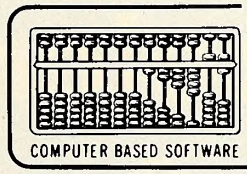
SEND FOR FREE CONTROL PAGE

* Data Base Merge
* Screen Format
* Font types changed in body of text
* With RH justification
* Multiple Print Drivers

NEC
Qume/Diablo
CENT 737
Epson
Paper Tiger
Write Your Own

Printer — Use any type: Epson NEC, Qume, CENT 737. All parallel, serial printer types.

DEALER INQUIRES INVITED



COMPUTER BASED SOFTWARE

LJK T.M. ENTERPRISES INC.

LJK ENTERPRISES INC.
P.O. Box 10827
St. Louis, MO 63129
(314) 846-6124

DEALER INQUIRES INVITED

*Apple T.M. of Apple Computer Inc., — *T.M. M&R Enterprises

Applesoft Accelerated

The New Compilers

BY CRAIG STINSON

Where once there were none, now suddenly there are many. Over the past two months, four Applesoft compilers have descended upon the marketplace.

Hayden Book Company's *Applesoft Compiler* arrived first—more or less on the first of July. On-Line Systems's *Expediter II* appeared about a week later. Within days of *Expediter*'s debut, a company in Tucson called Phase Zero surprised the market with another compiler: *Ascomp 2.5*.

Ascomp may become a collector's item, because it disappeared about as suddenly as it arrived. Southwestern Data Systems bought the product and promptly withdrew it from the marketplace—not because they found bugs, but because they wanted to take a good look at it to see if there were features they might want to add. As of early August, *Ascomp* was undergoing minor improvements on its way to becoming *SpeedStar*. The reincarnated compiler was due to appear in the stores September 1.

Meanwhile, Microsoft, the company that brought us Applesoft, was preparing to enter the fray with *TASC—The Applesoft Compiler*. *TASC* was in the final testing stage when this issue of *Softalk* went to press; the product was scheduled for release in mid August.

As if to add one more bit of chaos to this history, On-Line Systems announced during the first week of August that it was revising *Expediter II,* with the update due to arrive circa September 1.

In Competitive Market, the Buyer Wins. If you're an Applesoft programmer, this glut of compilers is, of course, a great boon. For the modest price of a compiler, you can enjoy the best of two worlds. You can use an interpretive high-level language for program development and debugging and then ship the finished product off to the compiler for conversion into swift, machine-level code. Or you can compile only speed-critical subroutines and call them from Applesoft programs.

That there are four compilers to choose from makes for a healthy, competitive market, but it also imposes some homework upon the prospective buyer. This review will attempt to ease that burden.

This is a somewhat unusual review for *Softalk*. Ordinarily, we look at finished products only, and ordinarily we don't report—amid the reviews—secondhand information about forthcoming changes in products. This kind of information properly belongs on the news pages, properly attributed.

However, the market is in such a state of flux right now that it wouldn't be meaningful to review the compilers without mentioning the changes promised by their publishers; and the products are significant enough to demand exposure without waiting until everybody is on terra firma.

Presenting the Contenders. So, *nota bene*: the review you're reading is based on specific versions of these compilers:

1. A beta test version of Microsoft's *TASC* with preliminary documentation. *TASC* was written by James M. Peak and Michael T. Howard.

2. The original version, in DOS 3.2, of Hayden's *Applesoft Compiler* by Jonathan Eiten.

3. *Ascomp*, from Phase Zero, by David Rifkind, Daniel Davidson, and Allan Goodman. To reduce confusion, we'll refer to this product as *SpeedStar*, from Southwestern Data Systems, since the SDS version will be the one available and essentially the same in design and approach.

4. On-Line Systems's *Expediter II,* Version 2.2—the mid-summer version, not the update expected in September. The *Expediter*'s authors are Stewart Einstein and Dennis Goodrow.

The four compilers range in price from $99.95 for *Expediter* to $200 for Hayden's *Applesoft Compiler*. *SpeedStar* will sell for $135 and *TASC* for $150.

General M.O. The four compilers have a few things in common and a lot of subtle differences. To start with common ground, all four compilers increase speed of program execution by translating high-level code into machine language. The tradeoff for this gain in speed is an increase in the size of the code; because of the nature of machine language, a compiled program uses up more memory and occupies more disk space than an equivalent program in Applesoft.

All four compilers control the size increase to some extent by using preexisting routines within the Applesoft ROM. All of them also employ some kind of run-time library of routines that must coreside in memory with the object code when the latter is executed.

For the purpose of this review we'll look at these two matters—execution speed and code size—and then describe the idiosyncrasies of the various compilers.

Turning Elmer Fudd into Roadrunner. How much faster will your program run after compilation than before? *Expediter*'s ads and manual talk about speed increases up to twentyfold, but that much acceleration would be unusual. It's realistic to expect your programs to speed up by a factor between two and ten. How much faster a program executes will depend on the nature of the program; in theory, the more symbols the source program contains, the greater will be its percentage speed increase upon compilation.

To make some comparisons between the compilers, we tested with relatively short and simple programs that isolated specific kinds of processing. The speed increases obtained with these programs may be a bit low compared to what you may achieve with more complex source programs. Use our figures for comparison only; your mileage may differ. With larger programs, you'll probably gain more.

First, we tested a simple two-level nested FOR-NEXT loop. The program took the Applesoft interpreter four minutes and twenty-two seconds to execute. Compiled by *Compiler*, the program ran in 3:14; by *TASC,* in 3:46; and by both *Expediter* and *SpeedStar*, in 3:47. Using an integer arithmetic option on *TASC,* an option that required adding a REM line to the source program, we reduced the execution time dramatically to 2:29.

A slightly more complex test program asked the computer to generate a random number, determine whether the result fell within a given range, divide the frequency of hits within that range by the total number of numbers generated, and print data to the screen after every hundredth iteration. Here we tested to see how many random numbers the 6502 could churn out in a given period of time (forty minutes). The source program yielded 95,200. Of the four compilers, Hayden's performed the best, with 178,500. *TASC*, *Expediter*, and *SpeedStar* followed, with 173,400, 172,200, and 156,000 respectively.

Priming Three Pumps. Next, we tried a simple prime-number finder. Applesoft took 4:37 to display all prime numbers less than a thousand. The *Compiler* accomplished the same task in 2:54. *Expediter* was close behind at 2:56, and *TASC* and *SpeedStar* trailed at 3:01 and 3:17 respectively.

A program involving both arrays and string-handling, as well as random numbers and some simple arithmetic calculation, produced both a greater spread between the various compilers and a larger increase in speed of all four over the Applesoft interpreter. The program dealt, evaluated, and tabulated one hundred random poker hands, all in text mode. The interpreter took exactly twelve and a half minutes to do the job. Of the four compilers, Hayden's came in first at 3:16. *Expediter* and *TASC* followed at 3:24 and 3:56, respectively, and *SpeedStar* brought up the rear, with 4:59.

Brian's Theme, from the Apple System Master disk, was used as a simple test of compiled hi-res graphics. The source program took ten minutes and thirty-three seconds to spin out thirty patterns. Of the compilers, Hayden and *Expediter* were again one and two with times of 5:34 and 5:38, respectively. *TASC* and *SpeedStar* were neck and neck at 5:57 and 6:00.

For a lo-res test, we compiled *Little Brick Out*. You haven't played pong until you've tried *Little Brick Out* at machine-language speed. In this case, it wasn't possible to come up with an objective speed comparison, but we can report the following subjective evaluation: Hayden's compilation was practically unplayable. *Expediter*'s and *TASC*'s were at levels slightly slower than Hayden's, and the *SpeedStar* version, while still a

formidable challenge, was a touch slower than the other three.

In terms of execution speed alone, the following picture emerges: for most programs, Hayden's *Compiler* appears to produce the fastest code. Microsoft's *TASC* may catch Hayden in programs that use a lot of integer variables but otherwise sits in third place, behind On-Line's *Expediter* and ahead of SDS's *SpeedStar*.

Code Is Bursting Out All Over. Execution speed, however, is only one criterion for evaluating a compiler. The speediest program in Appledom is worthless if it won't fit in your machine. In many applications, the size of the object code may matter more than the rate at which it executes.

On this point, Microsoft's *TASC* is a clear winner. *Little Brick Out*, for example, grew from 28 sectors on disk to 39 when compiled by *TASC*. The *Compiler* and *SpeedStar* expanded the same source to 60 sectors each, and the *Expediter* made it 62.

The 39-sector figure looks a little less impressive when you consider that a 17-sector (4K) run-time library has to be Bloaded before *Little Brick Out*—or any other *TASC*-compiled program—will run (the 60 and 62 sector figures given for the other compilers include their libraries). Nevertheless, in all programs tested, Microsoft produced the most compact object code, and with relatively large source programs, the differences became sizable.

A 56-sector game, for example, grew to 74 sectors (plus the obligatory 17) when compiled by *TASC*. Under *Expediter* and Hayden's *Compiler*, the same source grew to 112 and 127 sectors, respectively. The *Ascomp* we tested was unable to compile the program because of its size. That doesn't necessarily mean that *SpeedStar* won't be able to do it. One of the added features in SDS's revision of *Ascomp* will allow the user to compile larger programs by overlaying sections of the compiler instead of keeping it all in memory during compilation.

On-Line Systems has announced that the forthcoming update to the *Expediter* will give the user the option of producing a more compact but slightly slower running code.

Where, Oh, Where Has the Memory Gone? With all four compilers, programs that are too large to be compiled directly can be broken into separately compiled modules that share common variable space. The way the various compilers provide for the sharing of variables between modules, together with the way they allocate memory spaces in general, constitutes one of the more important sources of difference between them.

TASC's Task Handling. In *TASC,* the user controls the locations of three memory compartments: the run-time library, the program itself, and variables. In the default order of things, the library sits at decimal 2051, the program follows directly at 6020, and variables occupy the space between the top of the program and himem. Numeric variables build upward from the top of the program, and strings build down from himem. Lomem statements in the source program are not compiled because the bottom of the variable space is defined at compile time.

If you want to avoid clobbering areas of memory with your program, you can designate a starting address other than 6020. Either enter a specific number or type HGR1 or HGR2 to begin your program at the first byte following either of the hi-res pages. The program itself must be a contiguous block in memory; *TASC* will not JMP it around areas that you want to protect.

If you specify an alternate location for the library, you'll need to take note of its new address, because the library must be Bloaded separately from the rest of your object code at run-time.

To allow modularization with shared variables, *TASC* employs commands embedded in REM statements in the source program. The statement REM ! DEFCOMMON A,B,C would be ignored if the program were run through the interpreter, but it would instruct the compiler to allocate memory for three numeric variables in a protected global variable area. Other modules could gain access to these variables by means of a REM ! USECOMMON statement in their source listings. DEFCOMMON initializes global variables, while USECOMMON refers to them without reinitializing them.

All modules that use a particular set of global variables have to be compiled with the same starting address for the program space, since *TASC* allocates its common variable area at the beginning of this space. The global block is subdivided into two parts to separate numeric globals from string globals.

Modules that call upon common variables don't have to use the same variable names. One module could define common variables *A*, *B*, and *C*, and another could access those same variables under the names *D*, *E*, and *F*. What matters is the order in which the variables are declared in the source's REM statements.

Within the common block, *TASC* allocates two bytes per integer variable and five per real and expects the programmer to keep track of what kind of variable lies where. The compiler will check to see that each module is declaring the same overall amount of space in each of the common subunits but won't prevent the programmer from trying to use ten bytes of storage as five integers in one module and as two reals in another.

Hayden's One-Way Road. Hayden's *Compiler* takes a somewhat different approach to memory allocation. In the default configuration, the program is on the bottom of things, at decimal 2051. This block is followed first by variables and then by strings. The strings build upward from the top of the variable compartment toward a default himem of hex 9600.

In this default scheme, the compiler draws upon a library of ready-made subroutines and includes them as needed within the program space. You have the option of bringing the entire 3.3K library into memory as a separate compartment, external to the program space. Presumably this would be an efficient, space-saving thing to do only in cases where you were going to keep several compiled modules in memory at once to be called from an Applesoft program.

Variations from the default memory scheme are effected by means of an "active REM" statement, which must be the lowest numbered line in the source program. Allocating memory areas from within the source program rather than at compile time may seem like a little extra hassle, but it has the advantages of leaving you a convenient record of what you've done.

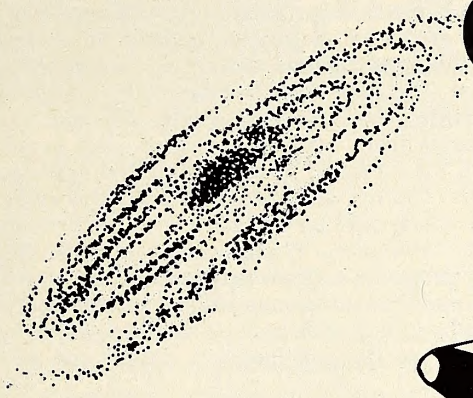
Hayden offers you the following allocation options: *B* specifies the starting address of the program: *X* summons and locates the optional external library; *L*, for lomem, specifies the beginning of the variable space; *S* declares the bottom end of the string pool; and *C* is for ceiling, the upper end of the string pool. These parameters can be arranged in just about any order you like, with the single proviso that *L* be lower than *C*. Both *L* and *C* may underlie the program space if you choose.

Separately compiled modules may share variables by declaring the same lomem. As with the Microsoft compiler, Hayden's global variables do not need to bear the same names. The order of declaration is what matters, and it's up to you, the programmer, to keep things straight. Since Hayden, unlike Microsoft, does not wall off a separate common variable block in memory, all the variables of at least one of the modules must be shared.

To avoid an implicit clear that would reinitialize shared variables, modules may be called at their starting address plus two. Hayden's manual recommends calling at the base address plus two, even if there are no variables in the called module, as a way of saving execution time.

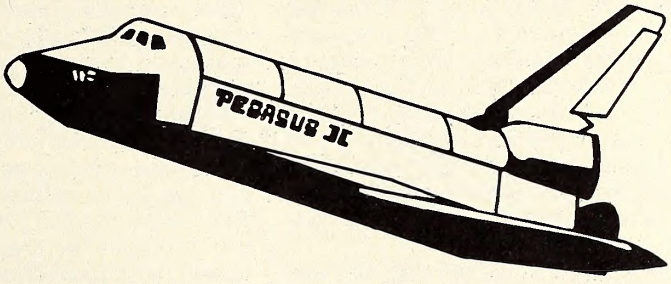
At least a couple of things are worth noting about Hayden's approach to string allocation. First, strings build upward toward himem rather than down and away from it. This wouldn't be possible with an interpreted Applesoft program because the variable space growing up from lomem is dynamically allocated. In a compiled program, all data spaces are static, so the compiler can set the low end of the string pool without fear of encroachment from the variables below. According to Hayden's manual, the upward building approach was chosen for two reasons: it lends speed to concatenations

WATCH OUT NASIR...
OLAF'S HERE!



PEGASUS II

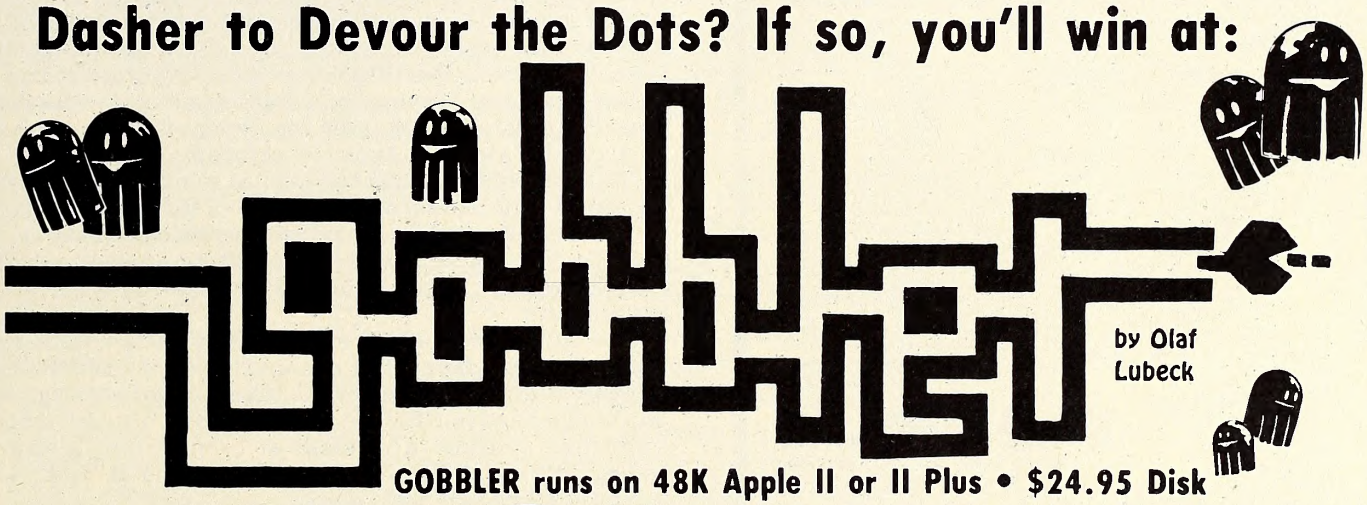
by Olaf Lubeck



$29.95
Disk 48K
Apple II or II Plus
Machine Language
DOS 3.2 or 3.3

ASK YOUR DEALER *for a Demonstration*

Can you Guide your Gobbler through the Gateways without Getting Got by a Ghost while Directing your Dasher to Devour the Dots? If so, you'll win at:



by Olaf Lubeck

GOBBLER runs on 48K Apple II or II Plus • $24.95 Disk

More Fine Products by... **ON-LINE systems**

To order send purchase price plus $1 for shipping to:
ON-LINE SYSTEMS - 36575 Mudge Ranch Road - Coarsegold, CA 93614 - 209-683-6858
Visa, MasterCharge, C.O.D. or Checks Accepted

and arithmetic string operations, and it permits the user to alter himem from within the source program.

The second—perhaps more important—point about Hayden's strings is that no housecleaning takes place. If you want to compile a sizable program with a lot of string handling, you'll need to sprinkle some FRE() statements into your source code or you may find yourself out of memory.

A Moving Expedition. On-Line's *Expediter* arranges memory in the following manner: A 2K run-time library occupies the space from hex 800 to hex 1100. The library is not relocatable; it may, however, be Bsaved and Bloaded separately from the rest of your object code as a means of conserving disk space (the normal procedure saves the entire library with every compiled program). The program itself, in the default arrangement, rises out of hex 1100; any other address above 1100 may be selected instead. Variables—both local and global—work their way downward from himem, which defaults to hex 9600. Either group of variables can be located elsewhere at the user's option.

As with *TASC*, *Expediter* users declare global variables by means of active REM statements in the source program. The procedures are a little different, however. In *TASC*, all DECOMMON or USECOMMON statements have to appear at the front of the program, before any non-REM statements. In *Expediter,* the REM <*G*> and REM <*L*> commands act like switches and can be used at any point in the program. All variables following a REM <*G*> statement and before the next REM <*L*> are global variables; all variables following REM <*L*> are locals, and so on. As with the other compilers, the programmer must keep track of the order in which global variables are referenced.

Strings in the *Expediter* are handled much the way they are in Integer Basic—as static data structures. The user dimensions them at compile time. The default value is 40; this figure can be overridden either globally, via a compile-time option, or for individual strings by means of active REM statements in the source program.

According to Stewart Einstein, coauthor of *Expediter*, the static approach to string handling was chosen for the sake of execution speed. A future revision may offer the user a choice between static and dynamic allocation.

Another option at compile time allows the user to protect up to ten areas of memory. This makes for an easy, space-efficient way to avoid clobbering the hi-res pages, shape tables, or whatever else you may want to protect. With this approach, you don't have to worry about relocating entire blocks of code; essentially, you just tell the compiler how high and it will JMP.

SpeedStar Takes Extra Lap. The ability to jump over protected areas of memory is one of the features Southwestern Data Systems is adding to *Ascomp* as *Ascomp* becomes *SpeedStar* in late August. The JMPs will be effected by means of active REM statements in the source program. This approach—specifying the jumps from within the source—will require a first pass through the compiler to determine the appropriate places from which to jump; but *SpeedStar* compiles extremely quickly, so the extra pass should not prove burdensome.

Other user-selectable memory parameters in *SpeedStar* include the starting program address and lomem. As with Hayden's *Compiler*, *SpeedStar* users can share variables between separate modules by giving each module the same value for lomem and avoiding the implicit clear on calling the various modules.

SpeedStar also allows you to specify an origin different from the starting address of the program. This might be useful if your were trying to compile a program above hi-res and had run out of memory because *SpeedStar* itself occupies the space between hex 7200 and DOS. With the origin option, you could compile the program at some lower base address, Bsave it, and, with *SpeedStar* out of memory, Bload it back in at the true starting address.

SpeedStar's run-time library, like *Expediter*'s, is approximately 2K. Unlike *Expediter,* however, *SpeedStar* keeps the run-time library attached to the program. One revision that *SpeedStar* programmers are considering for a future version (not the version due for release in September) would allow the user to relocate the library away from the program code or to strip it out altogether and allow separate modules to share the same library.

In any event, the fact that the library does not sit immovably at hex 800, as does *Expediter*'s, is important for *SpeedStar*, because this compiler has some features that facilitate the calling of compiled subroutines from within coresident Applesoft programs; having the library attached to relocatable program code gives the user the option of making hex 800 available for a calling Applesoft program.

To facilitate the calling of compiled subroutines, *SpeedStar* includes a parameter feature that allows the transfer of values from the Applesoft program into the compiled subroutine. The compiler also supports the Applesoft USR command, which allows the transfer of data from the subroutine back into the calling program.

The parameter option is selected via an active REM statement. Another active REM command allows a subroutine to be called without any implicit garbage collection taking place. Still another active REM command, when embedded within a subroutine, provides a message at compile time giving the calling address of that REM line. This would be useful for a compiled subroutine with multiple entry points.

One more note on the subject of memory allocation: *SpeedStar* programmers report that they're considering another optional feature—for another revision down the road a piece—that will shove the compiler off into a language card, if you have one, leaving more room in memory for your burgeoning object code. As things stand now—in the September version of *SpeedStar*—the compiler code either may be in memory in toto during compilation, which makes for very speedy compilation, or may be overlaid from disk, allowing for a larger object code. Using the language card would preserve compilation speed without encroaching upon potential program space.

Arithmetic Lesson. So much for matters of memory alloca-

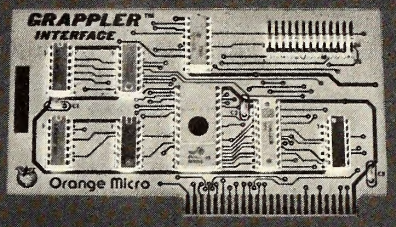
Mix APPLES and ORANGES?
. . . And you get GRAPHICS!

NEW

THE GRAPPLER™ Apple Parallel Interface and Cable

FROM ORANGE MICRO

The Grappler™ Interface is the first universal parallel interface card to provide sophisticated on-board firmware for Apple high resolution dot graphics. No longer does the user need to load clumsy software routines to dump screen graphics — it's all in a chip. Actually, it's our E-PROM, and there are versions to accommodate the Anadex, Epson MX100, MX80 & MX70, IDS Paper Tigers, Centronics 739, and future graphics printers. The Grappler™ accepts 18 simple software commands accessible through the keyboard or user program, making it the most intelligent Apple interface available.



GRAPPLER INTERFACE FEATURES

The High-Resolution Graphics commands include: **select page** 1 or page 2, print **inverse** graphics, **rotate** picture 90°, print **double** size picture, **center** graphics, and **chart recorder** mode. The Text commands allow the user to turn off the printer **bell** (Epson), turn on high bit for **block graphics** (Epson MX80 and Okidata), set left and right **margins,** set auto **skip-over-perf., variable line length** wrap-around with breakpoint on nearest blank, and **text screen dump.**

$165.00 Includes Cable

Dealer/Distributor Inquiries Invited

NEW

MX80/MX70 FRICTION FEED

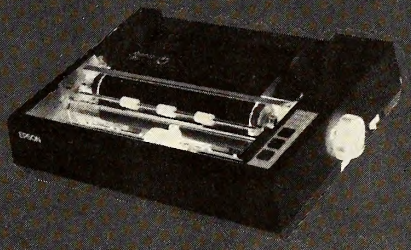
FROM ORANGE MICRO

HERE AT LAST! A high quality friction feed kit for your EPSON MX80/70. The kit allows the user to convert his Epson printer to a friction feed and pin feed mechanism. It uses a solid rubber platen for perfect tracking.

The friction feed will accept single sheets of your letterhead, or multiple copy forms such as invoices with up to 4 part carbon copies.

The pin feed replaces the adjustable sprocket mechanism. It allows the use of 9½" wide continuous fanfold paper which is an industry standard size.

No drilling is required. Initial installation takes about 30 minutes; return to adjustable sprockets in just 10 minutes. All parts are included with easy to follow instructions.



$75.00

Dealer/Distributor Inquiries Invited

CALL TOLL FREE:

(800) 854-8275

CA, AK, HI (714) 630-3322

Phone orders WELCOME; same day shipment. Free use of VISA & MASTERCARD. COD's accepted. Manufacturer's warranty included on all equipment. Prices subject to revision.

RETAIL PRINTER STORES:
3150 E. La Palma, #I, Anaheim, CA.
13604 Ventura Bl., Sherman Oaks, CA.
Store Hours: M-F 10-6 Sat. 10-4

master charge
VISA
AMERICAN EXPRESS

Orange Micro inc.

3150 E. La Palma Suite G, Anaheim, CA 92806

tion and communication between program modules. There are some other important differences among the compilers.

Microsoft's *TASC* and Hayden's *Compiler*, for example, will perform true integer arithmetic upon integer variables. Here's how this works. Applesoft allows you to declare integer variables by adding the percent sign to the variable name. When the interpreter goes to do anything with your integer variable, it first converts the value to a real and then performs its customary floating point arithmetic. So, actually, you usually slow your program down by declaring integer variables, because the values have to be converted before becoming useful as operands. The only place where declaring integer variables appears to be advantageous is in large arrays, where the savings in memory space (two bytes per element instead of five) may offset the drag on computation.

TASC and Hayden's *Compiler* will perform integer arithmetic to the extent they can. Hayden's will treat all numeric variables as integers until it encounters a real quantity or an operation requiring a real operand, at which time it will convert. *TASC* will treat as integers all variables declared as integer variables, and, rather than making you rewrite your source program to put in all those percent signs, it provides an active REM option that converts real variables to integer variables.

One of the limitations of Applesoft is its refusal to use integer variables as counters in FOR-NEXT loops. You can't even write FOR I% = 1 TO 10 without getting a syntax error. The REM ! INTEGER feature in *TASC* allows you to overcome this restriction, and the *TASC* manual strongly urges you to use it, claiming that loops with integer counters may execute up to two times faster than loops with real counters.

It isn't clear from its manual whether Hayden's *Compiler* uses integer arithmetic for loop counters, but, judging by the results of our simple loop test, where *TASC* aced out the usually faster Hayden compiler, it would appear that it does not.

Obedience Is Perfect in None. Not every Applesoft command has been implemented by these compilers. In particular, since the compilation eliminates the need to have source code in memory, the commands DEL and LIST have been ignored by all four compilers. CONT was passed up by everyone but Hayden; all it does in deferred execution is hang up your computer, but, if you want to put it in your source code, *Compiler* will compile it.

NEW, which might seem like a candidate for noninclusion, was implemented by all four compilers. In a compiled program, NEW clears out any Applesoft code that might have been lurking in memory without affecting your object code or variables. Oddly enough, Hayden's *Compiler* requires you to terminate a source program with NEW, rather than with END; your Apple may behave peculiarly if you don't.

Commands relating to cassette use—SAVE, LOAD, STORE, RECALL, and SHLOAD, have been almost universally ignored; the only exception is *Expediter*'s inclusion of SHLOAD.

TRACE and NOTRACE were excluded from implementation by Microsoft and Hayden on the reasonable grounds that debugging should be done with the interpreter before the program is compiled.

Users of *Expediter* and *SpeedStar* have the option of enabling line tracing features—at a considerable cost in code size and execution speed. The overhead for *Expediter*'s line trace option is seven bytes per source line, in return for which you get error messages, stops, and control-C breaks identified by program line number. You also get to TRACE in deferred execution. *SpeedStar* imposes a higher overhead. A numbers option, which tags your stops and error messages, costs eight bytes per source line. If you want to TRACE, you pay an additional three bytes per statement (not per line). On the other hand, *SpeedStar* allows you to select these features for specific sections of a program only—something that's not possible with *Expediter*.

The two compilers that did not implement TRACE and NOTRACE did implement RESUME. With *TASC*, there's an option available at compile time that will enable compilation of RESUME; the price is three bytes for every source statement capable of generating an error. If the source program does not use ONERR GOTO statements and the resume option is selected, run-time errors will be reported, with line numbers—making this feature useful as a debugging tool.

With Hayden, the support of RESUME is automatic; if the compiler encounters a RESUME, it generates an additional three bytes for every statement in the source program.

SpeedStar and *Expediter* did not implement RESUME.

The Dimensions of It All. Dimension statements are handled differently by all four compilers than by the Applesoft interpreter. All arrays are dimensioned statically, which means that DIMs with variable arguments will not compile; and only one dimension statement is permitted for a given array.

With Hayden and Microsoft, the latter constraint also applies to defined functions; these compilers will permit only one DEF FN statement per function name. *Expediter* and *SpeedStar*, on the other hand, treat DEF FNs exactly as they're treated by the interpreter.

One other interesting tidbit: *Expediter*, since it uses static string allocation and doesn't require the collection and disposal of garbage, has redefined the FRE token. FRE (X) in an *Expediter* source program will, on compilation, return the address of variable *X*.

Of the four compilers, Microsoft's *TASC* is the most disk-based. Each of three compiler passes has its own large binary file on disk, and there is constant I/O activity during the compilation. As a result, *TASC* takes more time to compile a program than any of the others; a large program may take as long as twenty minutes.

During pass one, the compiler acknowledges active REM statements and displays each line of source code as it's processed. If any fatal errors are encountered, the compiler halts, displays the source line and points to the offending syntax. At

Just when you thought it was SAFE to go into a computer store...

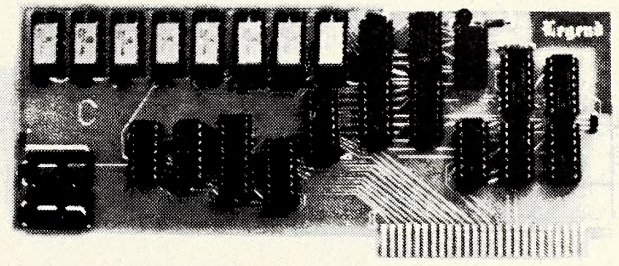
—Another Apple peripheral that you cannot live without!

*** THE 64KC BOARD FROM LEGEND ***

64 KC © 1981

by

Legend Industries Ltd.



An additional 64K of dynamic RAM for your Apple II using only one peripheral slot and less power than a single 16K RAM expansion board. This card plugs easily into any available peripheral slot inside the Apple. When plugged into slot 0 the card will be recognized as a 16K RAM card when used with Apple DOS 3.3. Bank #0 on the 64KC will be loaded with the language not contained in ROM on the Apple mother board when you initially 'Boot-up' the DOS 3.3 System Master diskette making both languages available to the user. Banks 1 through 3 on the 64KC card still provide 48K of additional memory. Pascal and Microsoft CP/M will also recognize the 64KC as a RAM board when installed in slot 0.

The LEGEND 64KC board contains eight state-of-the-art 64K dynamic RAM integrated circuits. The LEGEND 64KC system comes complete with a 64KC card, a 16 sector Demo diskette and an instruction manual written in English—not computerese. The demo programs and sample routines are written by Michael McLaren, whose program Dos Plus was listed in the Home/Hobby 10 in Softalk Magazine.

The most amazing part of this breakthrough product is the price— $349!

Go to your local dealer and see a demonstration today!

Products from Legend Industries—

64KC Board $349	McLaren LCG $49.95
64KC Manual $8	(lower case generator)

For dealer orders or if your local dealer does not carry these products—contact:

GREAT LAKES DIGITAL RESOURCES
P.O. Box 32133
Detroit, MI 48232

Phone (313) 538-7963

VISA/Master Card Accepted on Individual Orders

Apple is a registered trademark of Apple Computer, Inc.
64KC is a trademark of Legend Industries, Ltd

this point, you have the option to continue pass one to find any additional errors. Nonfatal errors—syntax that can't be compiled, like an extra colon, or a statement following on the same line with a GOTO—get a beep and a warning.

After compilation, *TASC* displays the ranges of each memory block—library, program, local and global variables, and the compiled addresses of each source line. A control-D PR#1 command at the start of compilation sends all of *TASC*'s output to the printer.

The Race That SpeedStar Wins. At the other end of a spectrum, *SpeedStar*, in its default configuration, will compile even a relatively large file in a matter of seconds. That's because, unless you specify otherwise, the compiler is entirely memory-bound. If you wish, *SpeedStar* will list your program as it goes, showing the compiled address of each line as it encounters it. Following the compilation, *SpeedStar* displays the address and length of two blocks: the program and library together, and the variables. Entering PR#1 before you load the source program will provide you with a permanent record of the compilation.

Hayden's *Compiler* does its work nearly as quickly as *SpeedStar*, though it has to go to disk occasionally during the process. Oddly, though the compiler sends the screen a running account of its every activity, it doesn't provide a convenient way to record this output on paper. At the end of the compilation, your screen will show the addresses and lengths broken down as follows: literals, constants, main code, run-time modules, end of binary file, scalar space, array space, string pool, and initial himem. Questionable or faulty parameter specifications show up on this map in inverse video.

Expediter, like *SpeedStar*, will display or print a running list of source program lines and compiled addresses. The same list includes the locations of all variables as they are compiled and the amount of source code remaining.

At the end of the compilation, statistics are displayed or printed: a symbol table; the start, end, and length of procedure, global data, and local data; the number of lines compiled; the number of variables; the number of data statements; the amount of temporary storage used by *Expediter*; and the number of forward references resolved during the compiler's final pass. The compiled code may be unreliable if the last figure exceeds 600.

What *Expediter* does not do during compilation is stop on fatal errors. The compiler merely identifies them and moves on. Since the compilation process is rather slow (although quicker than *TASC*'s), some people may find this aspect of *Expediter*'s performance a little irritating.

Of Protection and Licenses. At *Softalk*'s press time, Microsoft was planning to put *TASC* on an unprotected disk. Two of the others—Hayden's *Compiler* and SDS's *SpeedStar*—have dealt with the piracy problem by providing hardware keys that plug into the game port. This solution is handy for making backups, but it carries its own hazards and drawbacks. That little plug is easy to lose, and pins may bend and break, especially if the key is often swapped with game paddles. *Expediter* comes on a protected disk.

When you buy *Expediter*, you get one copy in 3.2 and one in 3.3. If you return either, On-Line Systems will provide you with a backup of the copy you keep. Microsoft's and SDS's compilers come on 3.2 disks that may be Muffined to 3.3. Hayden's is for the nonce available only in thirteen sectors. A 3.3 version is said to be forthcoming; in the meantime, the compiler comes with a copy program to provide for backups and a reverse *Muffin* (called *Niffum*) that will carry your sixteen-sector programs backward across time to the land of DOS 3.2.

Preliminary licensing arrangements have been announced for compiled commercial programs: Microsoft plans to charge nothing; Hayden wants to deal with the matter on a case-by-case basis; and both On-Line and SDS plan to charge a fee of five times the retail price of a compiled program per year, up to a maximum of $750 per year.

All four compilers require a 48K Apple. ■

A NEW STAR ON THE HORIZON

SpeeD Star

The Applesoft Compiler from Southwestern Data Systems

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

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

GET MORE THAN YOU PAY FOR

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

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

REQUIREMENTS

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

$134.95

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

SpeedStar . . . ask for it by name.

SDS
southwestern data systems

P.O. Box 582-S, Santee, CA 92071 • 714-562-3670

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

Everyone's Guide to Assembly Language, Part 12

COMMANDS COVERED SO FAR:

JMP	LDA	LDX	LDY	TAX
JSR	STA	STX	STY	TAY
RTS	INC	INX	INY	TXA
NOP	DEC	DEX	DEY	TYA
—	CMP	CPX	CPY	PHA
BEQ	BNE	BCC	BCS	PLA
SEC	CLC	ADC	SBC	

Figure 1.

This month I'd like to cover two main groups of machine language commands: shift operators and logical operators. Shifts are the easiest to understand, so we'll do them first.

Shift Operators. You'll recall that the accumulator holds a single eight-bit value and that, in previous topics, it has been possible to test individual bits by examining flags in the status register. An example of this was in testing bit 7 after an LDA operation. If the accumulator is loaded with a value from $00 to $FF, bit 7 is clear and only BPL tests will succeed, since the sign flag remains clear. If, however, a value from $80 to $FF is loaded, a BMI would succeed since bit 7 would be set, hence the sign flag will also be a 1.

The shift commands greatly extend our ability to test individual bits by giving us the option of shifting each bit in the accumulator one position to the left or right. There are two direct shift commands, ASL (arithmetic shift left) and LSR (logical shift right).

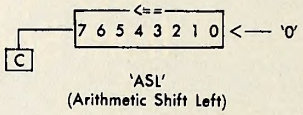


Figure 2.

In the case of ASL, each bit is moved to the left one position, with bit 7 going into the carry and bit 0 being forced to a zero. In addition to the carry, the sign and zero flags are also affected. Some examples appear in figure 3.

ASL

VALUE HEX	VALUE BINARY	RESULT HEX	RESULT BINARY	(C) CARRY	(N) SIGN	(Z) ZERO
$00	0000 0000	$00	0000 0000	0	0	1
$01	0000 0001	$02	0000 0010	0	0	0
$80	1000 0000	$00	0000 0000	1	0	1
$81	1000 0001	$02	0000 0010	1	0	0
$FF	1111 1111	$FE	1111 1110	1	1	0

Figure 3.

In the first case, there's no net change to the accumulator, although the carry and sign flags are cleared and the zero flag is set. The zero at each bit position was replaced by a zero to its right.

However, in the case of $01, the value in the accumulator doubles to become $02 as the one in bit 0 moves to the bit 1 position. In this case, all three flags will be cleared.

When the starting value is $80 or greater, the carry will be set. In the case of $80 itself, the accumulator returns to zero after the shift, since the only one in the pattern, bit 7, is pushed out into the carry.

Notice that, in the case of $FF, the sign flag gets set as bit 6 in the accumulator moves into position 7. Remember that in some schemes, bit 7 is used to indicate a negative number.

The ASL has the effect of doubling the byte being operated on. This can be used as an easy way to multiply by two. In fact, by using multiple ASLs, you can multiply by two, four, eight, sixteen, and so on, depending on how many you use. In last month's discussion of DOS and RWTS, you might remember that the IOB table required the slot number byte in the table to be sixteen times the true value. If you didn't want to do the multiplication ahead of time, you could do it in your access program, as in figure 4.

```
.
.
.
LDA USLOT     A5 09
ASL           0A
ASL           0A
ASL           0A
ASL           0A
STA SLOT      8D E9  B7
.
.
.
```

Figure 4.

USLOT holds the value from one to seven that you pass to the routine and SLOT is the location in the IOB table in which

the value for "USLOT *16" should be placed. Even though the four ASLs look a bit redundant, notice that they only took four bytes. Actually, the LDA/STA steps consumed more bytes (five) than the four ASLs.

In general then, ASL is used for these types of operations:

(1) Multiply by two, four, eight, and so on.

(2) Set or clear the carry "for free" while shifting for some other reason.

(3) Test bits 0 through 6. *Note:* This can be done, but it's usually only done this way for bit 6; there are, in general, better ways of testing specific bits, which we'll describe shortly.

The complement of the ASL command is the LSR. It behaves identically except that the bits all shift to the right (figure 5).

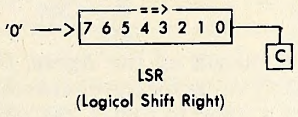


Figure 5.

The LSR can be used to *divide* by multiples of two. It's also a nice way to test whether a number is even or odd. Even numbers always have bit 0 clear. Odd always have it set. By doing an LSR followed by a BCC or BCS, you can test for this. Whether a number is odd or even is sometimes called its *parity*. An even number has a parity of zero, and an odd number, a parity of 1.

LSR also conditions the sign and zero flags.

In both LSR and ASL, one end or the other always gets forced to a zero. Sometimes this is not desirable. The solution to this are the *rotate* commands, ROL and ROR (rotate left, rotate right) (figure 6).

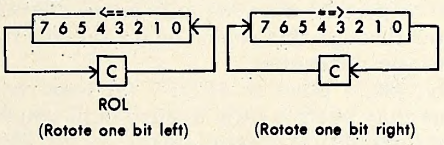


Figure 6.

In these commands, the carry not only receives the *pushed* bit, but its previous contents are used to load the now available end position.

ROL and ROR are used rather infrequently but do turn up occasionally in math functions such as multiply and divide routines.

So far, all the examples have used the accumulator as the byte to be shifted. As it happens, either the accumulator or a memory location may be shifted. Addressing modes include all the direct modes and indexed modes using the X register, with the exception of (MEM,X). The Y register cannot be used as an index in any of the shift operations.

Logical Operators. Logical operators are, to the uninitiated, some of the more esoteric of the machine language commands. As with everything we've done before, though, with a little explanation they'll become quite useful.

Let's start with one of the most commonly used commands, AND. You're already familiar with the basic idea of this one from your daily speech. If this AND that are a certain way, THEN I'll do something. This same way of thinking can be applied to your computer. As we've seen, each byte is made up of eight bits. Let's take just the first bit, bit 7, and see what kind of ideas can be played with. Normal text output on the Apple is always done with the high bit set. That is, all characters going out through COUT ($FDED) should be equal to or greater than $80 (1000 0000 binary). Likewise, when watching the keyboard for a keypress, we wait until $C000 has a value equal to or greater than $80.

Suppose we had a program wherein we would print characters to the screen only when a key was pressed and a standard character was being sent through the system. What we're saying is to print characters on the screen *only* when both the character AND the keyboard buffer show bit 7 set to 1.

We can draw a simple chart (figure 7) that illustrates all the possibilities (and you know how fond computer people are of charts. . .).

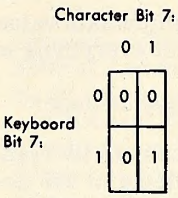


Figure 7.

The chart shows four possibilities. If the character's bit 7 is zero (a nonstandard character) and the keyboard bit is zero (no keypress), then don't print the character (a zero result). Likewise, if only one of the conditions is being met but not the other, then the result is still zero, or don't print. Only when *both* desired conditions exist will we be allowed to print, as shown by the one as the result.

Taken to its extreme, what we end up with is a new mathematical function, AND. In the case of a single binary digit (or perhaps we should call it a bigit), the possibilities are few, and the answers given as a simple zero or one.

What about larger numbers? Does the term *5 AND 3* have meaning? It turns out that it does, although the answer in this case will not be eight, and it is now that we must be cautious not to let our daily use of the word *addition* be confused with our new meaning.

As we look at these numbers on a binary level (figure 8), how to get the result of *5 AND 3* will be more obvious.

```
x = 5      0 1 0 1
y = 3      0 0 1 1
           ---------
x AND y    0 0 0 1 = 1
```

Figure 8.

If we take the chart created earlier and apply it to each set of matching bits in x and y, we can obtain the result shown. Starting on the left, two zeros gives zero as a result. For the next two bits, only a single one is present, in each case, still giving zero as a result. Only in the last position do we get the

necessary ones in bit 0 of *both* numbers to yield a one in the result.

Thus *5 AND 3* does have meaning, and the answer is one. (Try *that* at parties!)

Don't be discouraged if you don't see the immediate value in this operation; you should guess by now that everything is good for something!

AND is used for a variety of purposes. These include:

(1) To force zeros in certain bit positions.

(2) As a mask to let only ones in certain positions through.

When an AND operation is done, the contents of the accumulator are ANDed with another specified value. The result of this operation is then put back in the accumulator. The *other* value may be either given by way of the immediate mode or held in a memory location. These are some of the possible ways of using AND:

```
LDA    #$80
AND    #$7F
AND     $06
AND    $300,X
AND    ($06),Y
```

To understand better how the AND is used, some other ideas should be clarified. One of these is the nature of machine language programs in general. I believe that, at any given point in a program, one of two kinds of work will be going on. One is in the operational mode, where some specific task such as clicking a speaker or reading a paddle is taking place. At these moments, data as such does not exist. In the other case, the processing mode, data has been obtained from an operational mode, and the information is processed and/or passed to some other routine or location in memory.

A given routine is rarely made up of just one mode or the other, but any given step usually falls more into one category than the other.

These ideas are important because, in general, all the logical operators are used during the processing phases of a program. At those times, some kind of data is being carried along in a register or memory location. Part of the processing that occurs is often done with the logical operators.

In the case of the two modes of use mentioned, these are really just two different ways of looking at the same operation. To illustrate this, examine this partial disassembly of the Monitor starting at $FDED:

```
*FDEDL
FDED—  6C  36  00    JMP  ($0036)
FDF0—  C9  A0        CMP  #$A0
FDF2—  90  02        BCC  $FDF6
FDF4—  25  32        AND  $32
FDF6—  84  35        STY  $35
FDF8—  48            PHA
FDF9—  20  78  FB    JSR  $FB78
FDFC—  68            PLA
FDFD—  A4            LDY  $35
FDFF—  60            RTS
```

For normal text output on the Apple, the accumulator is loaded with the ASCII value for the character to be printed, the high bit is set, and a JMP to COUT ($FDED) is done. From looking at the listing, you can see that at $FDED is then an indirect jump based on the contents of $36,37 (called a *vector*).

If this seems a little vague, then consider for a moment what I call the *flow of control* in the computer. This means that the computer is *always* executing a program somewhere. Even when there's nothing but a flashing cursor on the screen, the computer is still in a loop programmed to get a character from the keyboard. When you call your own routines, the computer is just temporarily leaving its own activities to do your tasks until it hits that last RTS. It then goes back to what it was doing before; usually, that's waiting for your next command.

When characters are printed to the screen, disk, printer, or anywhere else, there's a flow of control that carries along the character to be printed. For virtually *every* character printed, the 6502 scans through this region as it executes the necessary code to print the character.

Normally, $36,37 point to $FDF0 (at least before DOS is booted). This may seem a little absurd until you realize that a great deal of flexibility is created by the vector. For instance, a PR#1, such as you do when turning on a printer, redirects $36,37 to point to the card, which, in turn, after printing a character, returns usually to where $36,37 *used* to print.

The card thus borrows the flow of control long enough to print the character, after which it gives control back to the screen print routine. Likewise, when DOS is booted, $36,37 get redirected from $FDF0 to $9EBD, which is where phrases preceeded by a control-D are detected. If no control-D is found, the output is returned to $FDF0.

Now, back to what the AND is used for. Normally, when the routine enters at $FDF0, the accumulator will hold a value between $80 and $DF. The characters from $80 to $9F are all control characters and are passed through by the BCC following the first CMP. Characters passing this test will be the usual alphabetic, numeric, and special characters shown on page 15 of the new *Apple Reference Manual*. You'll notice at this point an AND with the contents of $32 is done. Location $32 is called INVFLG and usually holds either $FF, $7F, or $3F depending on whether the computer is in the NORMAL, FLASHING, or INVERSE text modes. Let's assume that the accumulator is holding the value for a normal A and look at figure 9 to see what happens when an AND is done with each of these values.

EXAMPLE 1:	HEX	BINARY	ASCII
Accumulator:	$C1	1100 0001	A
INVFLG:	$FF	1111 1111	—
Result:	$C1	1100 0001	A
EXAMPLE 2:			
Accumulator:	$C1	1100 0001	A
INVFLG:	$7F	0111 1111	—
Result:	$41	0100 0001	A (flashing)
EXAMPLE3:			
Accumulator:	$C1	1100 0001	A
INVFLG:	$3F	0011 1111	—
Result:	$01	0000 0001	A (inverse)

Figure 9.

In the first example, ANDing with $FF yields a result identical to the original value. This is because, with each bit set to one, the result bit will always come out the same as the corresponding bit in the accumulator. (Can you guess what the result of ANDing with $00 would always yield?) This means that the character comes out in its original form.

In the second case, ANDing with $7F has the effect of forcing a zero in bit 7 of the result. Examining the chart on page 15 in the manual, we can see that $41 corresponds to a flashing A.

The Apple uses the leading two bits to determine how to print the character. If the leading two bits are *off*, then the character will be in inverse. If bit 7 is zero and 6 is one, then the character will be printed in flashing mode. If bit 7 is a set, then the character will be displayed in normal text.

Using the AND operator forces a zero in the desired positions and lets the remaining bit pattern through.

In general, then, the way to use the AND is to set a memory location equal to a value wherein all bits are set to one except for those that you wish to force to zero.

You can also think of AND as acting rather like a screen that lets only certain parts of the image through. When INVFLG is set to $3F, the leading bits will always be zero, regardless of whether they were set at entry or not; hence, the expression *mask*.

Sometimes figuring exactly what value you should use for the desired result is tricky. As a general formula, first decide what bits you want to force to zero and then calculate the number with all other positions set to ones. This will give the proper value to use in the mask. For example, to derive the inverse display mask value:

(1) Determine which bits to force to zero:

0 0 x x x x x x

(2) Calculate with the remaining positions set to ones:

0 0 1 1 1 1 1 1 = $3F (63)

Try this with the desired result of forcing bit 7 only to zero and see if you get the proper value for INVFLG of $7F.

BIT. The command somewhat related to the AND is BIT. This is provided to allow the user to determine easily the status of specific bits. When BIT is executed, quite a number of things happen. First of all, bits 6 and 7 of the memory location are transferred directly to the sign and overflow bits of the status register. Since we've not discussed the overflow flag, let me say just briefly that its related commands, BVC and BVS, may be used just as BPL and BMI are used to test the status of the sign flag. If V (the overflow flag) is clear, BVC will succeed. If V is set, BVS will work.

Most important, though, is the conditioning of the zero flag. If one or more bits in the memory location match bits set in the accumulator, the zero flag will be cleared. If no match is made, Z will be set. This is done by ANDing the accumulator and the memory location and conditioning Z appropriately. If the AND results in a nonzero number (one or more bits match), Z will be set to zero. The confusing part is that this may seem somewhat backward. Alas, it's unavoidable; it's just one of those notes to scribble in your book so as to remember the quirk each time you use it.

Note that one of the main advantages of BIT is that the accumulator is unaffected by the test.

Here is an example of how BIT might be used.

EXAMPLE 1: To test for bits 0 and 3, set:

```
LDA #$05  ; 0000 0101
BIT MEM
BNE OK    ; (1 OR MORE BITS MATCH)
```

EXAMPLE 2: To test for bit 7, set in memory:

```
CHK  BIT $C000  ; (KEYBOARD)
     BPL CHK    ; (BIT 7 CLR, NO KEY PRESSED)
     BIT $C010  ; (ACCESS $C010 TO CLR STROBE)
     ...
```

If you want to test for *all* of a specific set of bits being on, the AND command must be used directly.

EXAMPLE 3: To test for *both* bits 6 and 7 being on:

```
LDA CHAR  ; '1100 0000'
AND #$C0
CMP #$C0
BEQ MATCH ; BOTH BITS "ON"
```

This last example is somewhat subtle in that the result in the accumulator will only equal the value with which it was ANDed if each bit set to one in the test value has an equivalent bit on in the accumulator.

ORA and EOR. These last two commands bring up an interesting error of sorts in the English language, and that is the difference between an *inclusive OR* and the *exclusive OR*. What all this is about is the phenomenon that saying something like "I'll go to the store if it stops raining or if a bus comes by" has two possible interpretations. The first is that if *either* event happens, and even if both events occur, then the result will happen. This is called an *inclusive OR* statement.

The other possibility is that the conditions to be met must be one or the other but *not* both. This might be called the purest form of an *or* statement. It is either night *or* day, but never both. This would be called an *exclusive OR* statement.

In machine language, the inclusive OR function is called ORA for OR accumulator. The other is called EOR for exclusive OR. Figure 10 shows the charts for both functions.

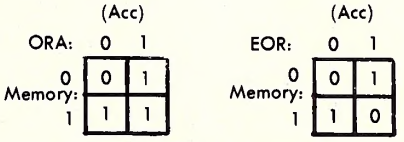
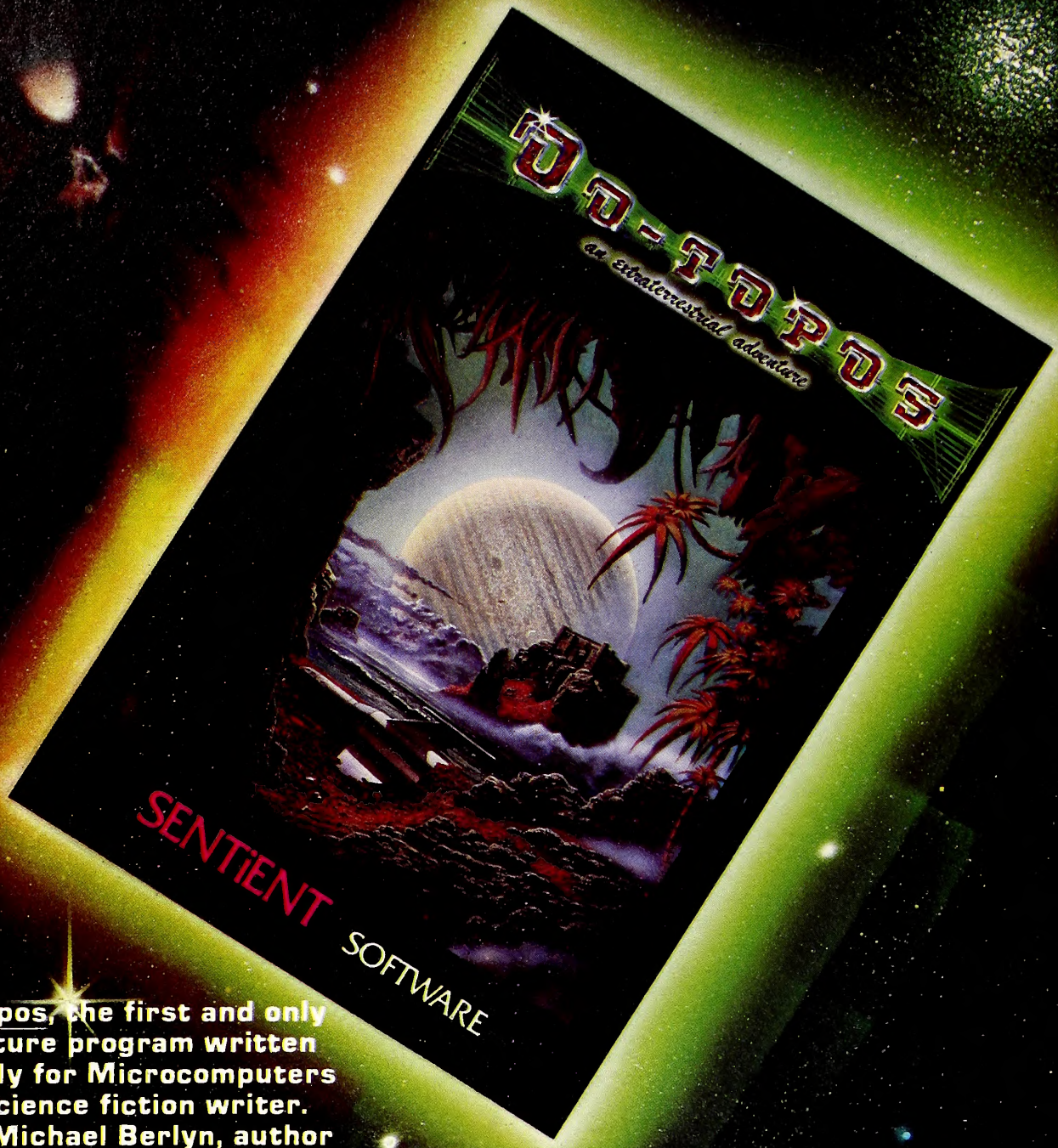


Figure 10.

First, consider the table for ORA. If either or both corresponding bits in the accumulator and the test value match, then the result will be a one. Only when neither bit is one does a zero value for that bit result. The main use for ORA is to force a one at a given bit position. In this manner, it's something of the complement to the use of the AND operator to force zeros.



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

See your local software dealer and discover Oo-Topos.

Requires DOS 3.3 Applesoft* in ROM

From

SENTIENT SOFTWARE

P.O. Box 4929, Aspen, Colorado 81612

(303) 925-9293

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

Exclusively distributed by

SOFTSEL

4079 Glencoe Ave.
Marina del Rey, CA 90291
(800) 421-0980
In CA (213) 822-8933

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

Figure 11 presents some examples of the effect of the ORA command.

	Example #1:			Example #2:		
Accumulator:	$80	1000	0000	$83	1000	0011
Value:	$03	0000	0011	$0A	0000	1010
Result:	$83	1000	0011	$8B	1000	1011

Figure 11.

Use of ORA conditions the sign and zero flags, depending on the result, which is automatically put into the accumulator.

The EOR command is somewhat different in that the bits in the result are set to one only if one or the other corresponding bits in the accumulator and test value are set to one, *but not both*.

EOR has a number of uses. The most common is in encoding data. An interesting effect of the table given is that for any given test value, the accumulator will flip back and forth between the original value and the result each time the EOR is done. See the examples in figure 12.

Accumulator:	$80	1000	0000	$83	1000	0011
Value:	$03	0000	0011	$0A	0000	1010
Result:	$83	1000	0011	$89	1000	1001
Accumulator:	$83	1000	0011	$89	1000	1001
Value:	$03	0000	0011	$0A	0000	1010
Result:	$80	1000	0000	$83	1000	0010

Figure 12.

This phenomenon is used extensively in hi-res graphics to allow images to overlay each other without destroying the image below. EOR can also be used to reverse specific bits. Simply place ones in the positions you wish reversed.

You might find it quite rewarding to write your own experimental routine to be able to EOR certain ranges of memory with given values. Then make the second pass to verify that the data has been restored. This is especially interesting when done either on the hi-res screen or on blocks of ASCII data such as on the text screen.

It would be a shame if you've stayed with me long enough to read through all this and didn't get a program for your efforts, so I offer the demonstration program in figure 13. It provides a way of visually experimenting with the different shifts and logical operators. Assemble the machine language program listed and save it to disk under the name OPERATOR.OBJ.

```
:
:ASM

                        1  ****************************
                        2  *  BINARY FUNCTION DISPLAY  *
                        3  *  UTILITY — 9/1/81         *
                        4  ****************************
                        5  *
                        6  *
                        7          OBJ   $300
                        8          ORG   $300
                        9  *
                       10  NUM     EQU   $06
                       11  MEM     EQU   $07
                       12  RSLT    EQU   $08
                       13  STAT    EQU   $09
                       14  *
                       15  YSAV1   EQU   $35
                       16  COUT1   EQU   $FDF0
                       17  CVID    EQU   $FDF9
                       18  COUT    EQU   $FDED
                       19  PRBYTE  EQU   $FDDA
                       20  *
                       21  *
0300: A9 00            22  OPERATOR LDA  #$00
0302: 48               23          PHA
0303: 28               24          PLP
0304: A5 06            25          LDA   NUM
0306: 25 07            26          AND   MEM     ; <= ALTER THIS
0308: 85 08            27          STA   RSLT
```

```
030A:  08          28           PHP
030B:  68          29           PLA
030C:  85 09       30           STA   STAT
030E:  60          31           RTS
                   32  *
030F:  A9 A4       33  PRHEX    LDA   #$A4    ;  '$'
0311:  20 ED FD    34           JSR   COUT
0314:  A5 06       35           LDA   NUM
0316:  4C DA FD    36           JMP   PRBYTE
                   37  *
0319:  A5 06       38  PRBIT    LDA   NUM
031B:  A2 08       39           LDX   #$08
031D:  0A          40  TEST     ASL
031E:  90 0D       41           BCC   PZ
0320:  48          42  P0       PHA
0321:  A9 B1       43           LDA   #$B1    ;  '1'
0323:  20 ED FD    44           JSR   COUT
0326:  A9 A0       45           LDA   #$A0    ;  'SPC'
0328:  20 ED FD    46           JSR   COUT
032B:  B0 0B       47           BCS   NXT
                   48  *
032D:  48          49  PZ       PHA
032E:  A9 B0       50           LDA   #$B0    ;  '0'
0330:  20 ED FD    51           JSR   COUT
0333:  A9 A0       52           LDA   #$A0    ;  'SPC'
0335:  20 ED FD    53           JSR   COUT
                   54  *
0338:  68          55  NXT      PLA
0339:  CA          56           DEX
033A:  D0 E1       57           BNE   TEST
                   58  *
033C:  60          59  EXIT     RTS
                   60  *
033D:  EA          61           NOP
033E:  EA          62           NOP
033F:  EA          63           NOP
                   64  *
0340:  C9 80       65  CSHOW    CMP   #$80    ;  STAND CHAR?
0342:  90 10       66           BCC   CONT
0344:  C9 8D       67           CMP   #$8D    ;  <=C/R>
0346:  F0 0C       68           BEQ   CONT
0348:  C9 A0       69           CMP   #$A0    ;  'SPC'
034A:  B0 08       70           BCS   CONT
                   71  *
034C:  48          72           PHA
034D:  84 35       73           STY   YSAV1
034F:  29 7F       74           AND   #$7F    ;  FORCE '0' IN BIT 7
0351:  4C F9 FD    75           JMP   CVID
                   76  *
0354:  4C F0 FD    77  CONT     JMP   COUT1
                   78  *
0357:  00          79  EOF      BRK
                   80  *
                   81  *

------ END ASSEMBLY------
TOTAL ERRORS: 0
88 BYTES GENERATED THIS ASSEMBLY
```

Figure 13.

Then enter the accompanying Applesoft program (figure 14) and save it under the name OPERATOR DEMO PROGRAM. I have a lower-case display board that allows me to use the vertical bar in the display. If you do not have this feature, try an exclamation point (!) on lines 1100 to 1885 in place of the bar (¦).

```
LOGICAL OPERATOR DEMO PROGRAM

9/1/81
```

```
0  IF  PEEK (768) < > 169 THEN  PRINT  CHR$ (4);"BLOAD OPERATOR.OBJ,A$300"
5  POKE 54,64: POKE 55,3: CALL 1002: REM HOOK UP CTRL SHOW
10  REM LOGICAL OPERATOR PROG.
15 OP = 774:F = 768:PH = 783:PB = 793
20  TEXT : HOME : GOTO 1000
100 KEY =  PEEK ( - 16384): IF KEY > 127 THEN 1000
110 A =  PDL (0):A =  PDL (0)
120 M =  PDL (1):M =  PDL (1)
125  POKE 6,A: POKE 7,M
130  CALL F: REM EVALUATE FUNCTION
140 R =  PEEK (8):S =  PEEK (9)
200  VTAB 11: HTAB 1: PRINT "OPCODE:";: POKE 6,0: GOSUB 500: VTAB 11: HTAB 32: PRINT "/";O$;"/"
210  VTAB 14: PRINT "ACC:";: POKE 6,A: GOSUB 500: HTAB 30: PRINT " ";: HTAB 30: PRINT  CHR$ (A);: VTAB 14: HTAB 3
3: PRINT "(P0)";: POKE 1742,A: IF A = 13 OR A = 141 THEN  VTAB 14: HTAB 30: INVERSE : PRINT "M": NORMAL
215  IF O1 = 7 THEN  VTAB 16: PRINT "MEMORY:";: POKE 6,M: GOSUB 500: HTAB 30: PRINT " ";: HTAB 30: PRINT  CHR$ (M
);: VTAB 16: HTAB 33: PRINT "(P1)";: POKE 1998,M: IF M = 13 OR M = 141 THEN  VTAB 16: HTAB 30: INVERSE : PRINT "M"
: NORMAL
220  IF O$ < > "BIT" THEN  VTAB 18: PRINT "RESULT:";: POKE 6,R: GOSUB 500: HTAB 30: PRINT " ";: HTAB 30: PRINT  CHR$
(R): POKE 1270,R: IF R = 13 OR R = 141 THEN  VTAB 18: HTAB 30: INVERSE : PRINT "M": NORMAL
230  VTAB 20: PRINT "STATUS:";: POKE 6,S: GOSUB 500: PRINT
240  VTAB 22: HTAB 10: PRINT "N V - B O I Z C"
250  GOTO 100
499  END
500  REM PRINT BITS & HEX
510  HTAB 10: CALL PB: HTAB 26: CALL PH: RETURN
1000  REM SELECT FUNCTION
1010 T =  PEEK ( - 16368): FC = FC + 1 - (KEY = 136) * 2: IF FC > 8 THEN FC = 1
1011  IF KEY = 193 THEN FC = 1
1012  IF KEY = 194 THEN FC = 3
1013  IF KEY = 197 THEN FC = 4
1014  IF KEY = 204 THEN FC = 5
1015  IF KEY = 207 THEN FC = 6
1016  IF KEY = 210 THEN FC = 7
1019  IF FC < 1 THEN FC = 8
1020  FOR I = 1 TO FC: READ O$,O,O1: NEXT I: RESTORE
1025  IF KEY = 155 THEN  PRINT  CHR$ (4);"PR#0": END
1030  POKE OP,O: POKE OP + 1,O1: HOME
1050  ON FC GOSUB 1100,1200,1300,1400,1500,1600,1700,1800
1055  POKE 32,0
1060 A =  - 1: GOTO 100
1100  REM 'AND'
1110  POKE 32,9
1140  VTAB 1: PRINT "-----------------"
1145  PRINT "| AND |  0  |  1  |"
1150  PRINT "-----------------"
1155  PRINT "|  0  |  0  |  0  |"
1160  PRINT "-----------------"
1165  PRINT "|  1  |  0  |  1  |"
1170  PRINT "-----------------"
1175  PRINT : HTAB 7: PRINT "'AND'"
1180  VTAB 23: PRINT "^           ^"
1185  RETURN
1200  REM 'ASL'
1220  VTAB 1: HTAB 9: PRINT "-------<==-------"
1225  HTAB 4: PRINT "-----|7|6|5|4|3|2|1|0|<-- '0'"
1230  HTAB 4: PRINT "|    ---------------"
1235  HTAB 3: PRINT "---"
1240  HTAB 3: PRINT "|C|"
1245  HTAB 3: PRINT "---"
```

DISCOVER THE UNIVERSE

PROUDLY ANNOUNCING

SPACE ADVENTURE

A REAL-TIME SPACE FLIGHT SIMULATOR

Welcome aboard the Callisto.

The Callisto offers you an excellent view of the vast universe before you.

You take the helm and enter the coordinates: quadrant 56, 58, 53; sector 14, 62, 12. Instantly the autopilot takes control of the ship. The viewport reappears as your course automatically changes and you begin your hyperwarp journey. Upon arrival, the message indicator flashes. You take the message and ask the computer for a tactical plot of the situation.

SPACE ADVENTURE is this and much more!

SPACE ADVENTURE offers you real-time instruments, real-time graphics, and an onboard computer that will amaze you.

SPACE ADVENTURE gives you the chance to pilot a space craft through true 3-D space for an adventure you will never forget. **SPACE ADVENTURE** . . . for any Apple II or II+ with 48K and disk. Suggested retail price is $29.95.

Sierra Software

EPISODE TWO COMING SOON
DEALER INQUIRIES INVITED

SIERRA SOFTWARE • 536 E. SAHARA AVENUE • LAS VEGAS, NEVADA 89104

How's your love life?

A little dull around the edges? Routine? Predictable? Boring? Maybe all it needs is a little Interlude. Interlude is the most stimulating computer game ever conceived. It combines a computer interview, an innovative programming concept, and a one-of-a-kind manual to turn your love life into exciting, adventurous, delicious fun!

Interlude is: romantic...playful...outrageous...a fantasy. Interlude is: ■ A Bed of Roses (Interlude #1) ■ Mata Hari (Interlude #49) ■ The Chase (Interlude #7) ■ Rodeo! (Interlude #71) ■ The King and I (Interlude #60) ■ Some Enchanted Evening (Interlude #84) ■ Caveman Caper (Interlude #82) ■ From Here to Ecstasy (Interlude No. 30) ■ Satin Dreams (Interlude #72).

More than 100 Interludes are included in the program. Most are described in detail in the accompanying manual, but several surprise Interludes are buried in the program awaiting that very special time when your interview says you're ready. (When you learn secret Interlude #99, your love life may never again be the same!) Interlude can give you experiences you'll never forget. Are you ready for it?

Interlude™
The Ultimate Experience.

INTERLUDE, Dept. ST, 10635 Richmond Avenue, Houston, Texas 77042. I'm really ready. Send my Interlude today.

Apple II (16K)*
☐ Cassette ($18.95)
☐ Diskette ($21.95)
☐ Diskette — Pascal or DOS 3.3 ($21.95)

TRS-80 (Model 1, 16K)**
☐ Cassette ($18.95)
☐ Diskette ($21.95)

Poster
☐ 20"x 24" reproduction of this ad without ad copy ($4.95—includes shipping charges)

Add $1.75 for shipping and handling.

Available for immediate shipment.

Please enclose your check payable to INTERLUDE or complete the charge information:

☐ MASTERCARD ☐ VISA All charge customers must sign here._____
Account No. _____ Expiration date_____ MasterCard Bank Code _____

CHARGE CUSTOMERS: Order by phone toll-free! **1-800-231-5768 Ext. 306** (Texas: 1-800-392-2348 Ext. 306)

Name _____ Age _____
Address _____
City _____ State _____ Zip _____

*Apple II is a registered trademark of Apple Computers, Inc. **TRS-80 is a registered trademark of Radio Shack, a Tandy Co.

```
1250  VTAB 7: HTAB 16: PRINT "'ASL'": HTAB 8: PRINT "(ARITHMETIC SHIFT LEFT)"
1280  VTAB 23: HTAB 10: PRINT "^           ^ ^"
1285  RETURN
1300  REM 'BIT'
1310  POKE 32,9
1340  VTAB 1: PRINT "-------------------"
1345  PRINT "| AND |  0  |  1  |"
1350  PRINT "-------------------"
1355  PRINT "|  0  |  0  |  0  |"
1360  PRINT "-------------------"
1365  PRINT "|  1  |  0  |  1  |"
1370  PRINT "-------------------"
1375  PRINT : HTAB 7: PRINT "'BIT'"
1380  VTAB 23: PRINT "M M         ^": PRINT "7 6";
1385  RETURN
1400  REM 'EOR'
1410  POKE 32,9
1440  VTAB 1: PRINT "-------------------"
1445  PRINT "| EOR |  0  |  1  |"
1450  PRINT "-------------------"
1455  PRINT "|  0  |  0  |  1  |"
1460  PRINT "-------------------"
1465  PRINT "|  1  |  1  |  0  |"
1470  PRINT "-------------------"
1475  PRINT : HTAB 7: PRINT "'EOR'"
1480  VTAB 23: PRINT "^           ^"
1485  RETURN
1500  REM 'LSR'
1520  VTAB 1: HTAB 9: PRINT "-------=>-------"
1525  HTAB 2: PRINT "'0' -->|7|6|5|4|3|2|1|0|-----"
1530  VTAB 3: HTAB 9: PRINT "---------------   |"
1535  HTAB 29: PRINT "---"
1540  HTAB 29: PRINT "|C|"
1545  HTAB 29: PRINT "---"
1550  VTAB 7: HTAB 15: PRINT "'LSR'": HTAB 8: PRINT "(LOGICAL SHIFT RIGHT)"
1580  VTAB 23: HTAB 10: PRINT "0           ^ ^"
1585  RETURN
1600  REM 'ORA'
1610  POKE 32,9
1640  VTAB 1: PRINT "-------------------"
1645  PRINT "| ORA |  0  |  1  |"
1650  PRINT "-------------------"
1655  PRINT "|  0  |  0  |  1  |"
1660  PRINT "-------------------"
1665  PRINT "|  1  |  1  |  1  |"
1670  PRINT "-------------------"
1675  PRINT : HTAB 7: PRINT "'ORA'"
1680  VTAB 23: PRINT "^           ^"
1685  RETURN
1700  REM 'ROL'
1720  VTAB 1: HTAB 9: PRINT "-------<=-------"
1725  HTAB 4: PRINT "-----|7|6|5|4|3|2|1|0|<----"
1730  HTAB 4: PRINT "|    ---------------   |"
1735  HTAB 4: PRINT "|          ---          |"
1740  HTAB 4: PRINT "----------->|C|-----------"
1745  HTAB 16: PRINT "---"
1750  VTAB 8: HTAB 15: PRINT "'ROL'": HTAB 9: PRINT "(ROTATE ONE BIT LEFT)"
1780  VTAB 23: HTAB 10: PRINT "^           ^ ^"
1785  RETURN
1800  REM 'ROR'
1820  VTAB 1: HTAB 9: PRINT "-------=>-------"
1825  HTAB 4: PRINT "---->|7|6|5|4|3|2|1|0|----"
1830  HTAB 4: PRINT "|    ---------------   |"
1835  HTAB 4: PRINT "|          ---          |"
1840  HTAB 4: PRINT "-----------|C|<----------"
1845  HTAB 16: PRINT "---"
1850  VTAB 8: HTAB 15: PRINT "'ROR'": HTAB 9: PRINT "(ROTATE ONE BIT RIGHT)"
1880  VTAB 23: HTAB 10: PRINT "^           ^ ^"
1885  RETURN
2000  DATA AND,37,7,ASL,10,234,BIT,36,7,EOR,69,7,LSR,74,234,ORA,5,7,ROL,42,234,ROR,106,234
65535 REM COPYRIGHT (C) 1981
65535 REM  ROGER R. WAGNER
```

Figure 14.

The basic theory of operation for the program is to rewrite locations $306 and $307 with the appropriate values to create the different operators. These values are contained in the DATA statement on line 2000 of the Applesoft program. In addition, there are routines to print the value in location $06 in both binary and hex formats. Also, there is a routine to show control characters in inverse. You may wish to examine each of these to determine the logic, if any, behind their operation.

The Applesoft program itself operates by getting a value for the accumulator and the memory location from paddles 0 and 1. The double read in lines 110, 120 minimizes the interaction between the two paddles. Pressing any key advances the display to the next function, the left arrow backs up. Pressing A, B, E, L, O, or R will jump to selected functions.

The screen display shows the hex and binary values for each number and also what character would be printed via a PRINT CHR$(X) statement (control characters are shown in inverse). To the far right is the character obtained when the value is poked into the screen display part of memory.

I suppose if I were a purist, the entire thing would have been written in machine language. Oh well, maybe next issue.

This may be the first Apple II game that is copied for the arcade machines. *RASTER BLASTER* for the Apple II and Apple II Plus is so technically sophisticated and fun to play that it is sure to attract the attention of the big arcade manufacturers. But you can get it right now for your Apple.

RASTER BLASTER

Video Pinball for the Apple II

Dealers Inquiries Invited

BudgeCo, 428 Pala Avenue, Piedmont, California 94611

(415) 658-8141

Softalk Presents The Bestsellers

A spate of new product kept the month of July from being a disaster at the software counters of most retailers. Dealers across the nation were reporting business generally down by 10 percent or more, with the decline limited to that magnitude only by several new software packages that caught the public's eyes.

Three new entries made the top ten and five other programs made the Top Thirty for the first time. Another package regained a previously held position on the list.

The weaker marketplace, combined with greatly expanded choices for the buyer, resulted in the closest spread from top to bottom yet witnessed in the poll. Under normal marketplace conditions, the bestselling program will outperform the bottom program on the Top Thirty by approximately five to one. This month the spread is barely larger than three to one.

The narrowed margin of sales is best seen right at the top where *Raster Blaster* held on to first place, but by only 1.70 index points over *Gorgon* and only 3.98 index points over *VisiCalc*. This is the smallest difference between first place and third place yet noted in the bestsellers list.

Raster Blaster's third straight first-place finish makes Bill Budge the leading Apple programmer, but Nasir continues to threaten. Nasir has *Space Eggs* in twelfth and *Pulsar II* in twenty-first as well as *Gorgon* in second.

Mitch Kapor of Micro Finance Systems is bidding for top business programmer honors. The combined sales of *VisiPlot* and *Visitrend/VisiPlot* would rank tied for sixth. This would place him ahead of *DB Master*'s triumvirate of Stanley Crane, Jerry Macon and Barney Stone and within striking distance of the alltime champs—Dan Bricklin and Robert Frankston, authors of *VisiCalc*.

Hottest new software package was *Expediter II* from On-Line Systems. By arriving on the market as the lowest priced of the Applesoft compilers, it captured by far the majority of early buyers.

On-Line snared another spot in the top ten with *Hi-Res Adventure #3: Cranston Manor*.

The final new entry in the top ten was *Apple Panic* from Broderbund Software. *Apple Panic* was not out for the full month of July, but at the end of the month it was the hottest individual item in the market.

Business 10

This Month	Last Month	
1.	1.	*VisiCalc,* Software Arts/Dan Bricklin and Robert Frankston, Personal Software
2.	2.	*DB Master,* Alpine Software/Stanley Crane and Jerry Macon; and Barney Stone, Stoneware
3.	4.	*VisiTrend/VisiPlot,* Micro Finance Systems/Mitch Kapor, Personal Software
4.	5.	*VisiDex,* Peter Jennings, Personal Software
5.	—	*Personal Filing System,* John Page, Software Publishing Corp.
6.	3.	*Apple Writer,* Apple Computer Inc.
	7.	*Magic Window,* Gary Shannon and Bill Depew, Artsci
8.	8.	*VisiPlot,* Micro Finance Systems/Mitch Kapor, Personal Software
9.	9.	*PFS: Report,* John Page, Software Publishing Corp.
10.	—	*Wordstar,* Micro Pro

Other new programs making the Top Thirty were *Computer Baseball* from Strategic Simulations, *Space Warrior* from Broderbund, *Oo Topos: An Extraterrestrial Adventure* from Sentient Software, *The Complete Graphics System* from Penguin Software, and *Dragon Fire* from Level 10 Software. Penguin Software is the new name for Co-op Software.

Finding its way back to the Top Thirty after a month's hiatus was *Personal Filing System* from Software Publishing Corp. Some pundits had speculated that Personal Software's *VisiDex* would impact *PFS* sales extensively, but both programs seem to be finding a large number of adherents.

VisiCalc again led the Business 10. The biggest news in that list was the appearance of *Wordstar* at the bottom. Micro Pro's CP/M word processor is apparently finding considerable support among owners of Microsoft's SoftCard.

In the Home/Hobby 10, *Expediter* came from nowhere to grab the lead with *The Complete Graphics System* moving into fifth spot. This list was marked by the first appearance of Apple's Special Delivery Software: "*The World's Greatest Blackjack Program*" nudged into tenth spot.

In this regard, it should again be noted that these bestseller charts only track sales at the retail level. Software packages that are sold exclusively or primarily directly to the end user by the distributor or publisher will be underreported in relation to their true sales.

Beginning with the October issues, *Softalk* will expand its bestseller listings to break out more categories similar to the Business 10 and the Home/Hobby 10. The expansion of the report will better reflect the diversity of special interests within the Apple community. ■

Home/Hobby 10

This Month	Last Month	
1.	—	*Expediter II,* Stewart Einstein and Dennis Goodrow, On-Line Systems
2.	1.	*DOS 3.3,* Apple Computer Inc.
3.	3.	*DOS Tool Kit,* Apple Computer Inc.
4.	2.	*Typing Tutor,* Image Producers, Microsoft
5.	—	*The Complete Graphics System,* Marc Pelczarski, Penguin Software
6.	5.	*Home Money Minder,* Bob Schoenburg and Steve Pollack, Continental Software
7.	4.	*Super Disk Copy,* Charles Hartley, Sensible Software
8.	7.	*VisiTerm,* Tom Keith, Personal Software
9.	8.	*ASCII Express,* Bill Blue, Southwestern Data Systems
	—	*"The World's Greatest Blackjack Program,"* Special Delivery Software, Apple Computer Inc.

Apple-franchised retail stores representing approximately 6.8 percent of all sales of Apples and Apple-related products volunteered to participate in the poll.

Respondents were contacted early in August to ascertain their sales leaders for the month of July.

The only criterion for inclusion on the list was number of sales made—such other criteria as quality of product, profitability to the computer retailer, and personal preference of the individual respondents were not considered.

Respondents in August represented every geographical area of the continental United States.

Results of the responses were tabulated using a formula that resulted in the index number to the left of the program name in the Top Thirty listing. The index number is an arbitrary measure of relative strength of the programs listed. Index numbers are correlative only for the month in which they are printed; readers cannot assume that an index rating of 50 in one month represents equivalent sales to an index number of 50 in another month.

Probability of statistical error is plus-or-minus 6.1 percent, which translates roughly into the theoretical possibility of a change of 4.6 points, plus or minus, in any index number.

The Top Thirty

This Month	Last Month	Index	
1.	1.	90.99	*Raster Blaster,* Bill Budge, BudgeCo
2.	3.	89.29	*Gorgon,* Nasir, Sirius Software
3.	4.	87.01	*VisiCalc,* Software Arts/Dan Bricklin and Robert Frankston, Personal Software
4.	2.	81.89	*Pool 1.5,* Don Hoffman, Howard de St. Germaine, and Dave Morock, Innovative Design Software
5.	—	76.77	*Expediter II,* Stewart Einstein and Dennis Goodrow, On-Line Systems
6.	13.	72.79	*Robot War,* Silas Warner, MUSE
7.	—	64.83	*Apple Panic,* Ben Serki, Broderbund Software
8.	22.	49.48	*Ultima,* Lord British, California Pacific
9.	—	47.20	*Hi-Res Adventure #3: Cranston Manor,* Harold DeWitz and Ken Williams, On-Line Systems
	16.	47.20	*Snoggle,* Jun Wada, Broderbund Software
11.	17.	46.06	*DB Master,* Alpine Software/Stanley Crane and Jerry Macon; and Barney Stone, Stoneware
12.	7.	44.93	*Space Eggs,* Nasir, Sirius Software
	6.	44.93	*DOS 3.3,* Apple Computer Inc.
14.	27.	44.36	*VisiTrend/VisiPlot,* Micro Finance Systems/Mitch Kapor, Personal Software
15.	8.	42.65	*Hi-Res Adventure #2: The Wizard and the Princess,* Roberta and Ken Williams, On-Line Systems
16.	5.	40.95	*Flight Simulator,* Bruce Artwick, SubLogic
17.	19.	40.38	*Gobbler,* Olaf Lubeck, On-Line Systems
	29.	40.38	*VisiDex,* Peter Jennings, Personal Software
19.	19.	38.10	*DOS Tool Kit,* Apple Computer Inc.
20.	10.	36.40	*Alien Rain,* Tony Suzuki, Broderbund Software
21.	14.	30.15	*Pulsar II,* Nasir, Sirius Software
22.	—	29.57	*Computer Baseball,* Charles Merrow and Jack T. Avery, Strategic Simulations
	—	29.57	*Space Warrior,* Marc Goodman, Broderbund Software
24.	—	29.00	*Personal Filing System,* John Page, Software Publishing Corporation
25.	27.	28.44	*Warp Factor,* Paul Murray, Strategic Simulations
26.	11.	27.30	*Typing Tutor,* Image Producers, Microsoft
27.	24.	25.59	*Asteroid Field,* Jim Nitchals, Cavalier Software
	—	25.59	*Oo Topos: An Extraterrestrial Adventure,* by Michael Berlin, Sentient Software
	—	25.59	*The Complete Graphics System,* Marc Pelczarski, Penguin Software
	—	25.59	*Dragon Fire,* Level 10 Software

WE'VE CREATED A WORLD JUST FOR YOU.

OSGORTH! It is a wonder world of swords-and-sorcery adventure to serve as your personal playground.

THE SHATTERED ALLIANCE™, first in a line of forthcoming games from SSI that comprise the **Chronicles of Osgorth**, is a sophisticated strategy simulation centered on this rich fantasy planet and its ethereal trappings: magic, rivalry between realms, war among exotic creatures.

With the multitude of armies either preassembled or designed by you, you can conjure up limitless scenarios — even the unlikely confrontation of Osgorthian forces against Alexander the Great's Companions or Caesar's Legionnaires!

Ah, the possibilities!

THE APPLE COMPUTER paints in beautiful, color Hi-Res graphics an Osgorthian battlefield of knolls, mountains, forests, and open plains. Unit types are categorized into five classes of infantry and four of cavalry. The tribes that populate this world are as varied as they are enchanted: dwarves, unicorns, amazons, centaurs — just to name a few.

Your fantasy soldiers have been carefully rated for armor, speed, strength, and morale. Their weapons may be the bow, javelin, rocks, spears, pikes, swords, axes, clubs, or cavalry lance....and of course, spells that can magically enhance a unit's speed, defense, attacking power, or courage.

AS RICHLY DETAILED as this game is, it never becomes unwieldly because the computer keeps track of everything. Its simple but elegant movement system was designed so that all action can proceed at a brisk pace. It also comes with a powerful and intelligent computer opponent for solitaire play.

THE SHATTERED ALLIANCE is fun, fast, and easy to play — geared to incite your imagination. It comes with the game program disc (for a 48K Apple II with ROM card and a mini floppy disc drive), a rulebook, and four player-aid charts. For $59.95,get this fantastic piece of real estate today!

To order with your VISA or MASTERCARD, call toll free **800-227-1617, ext. 335** (800-772-3545, ext. 335 in California). To order by mail, send your check to Strategic Simulations Inc, Dept. S3, 465 Fairchild Drive, Suite 108, Mountain View, CA 94043. All our games carry a 14-day money back guarantee.

As part of our demanding standards of excellence, we use maxell floppy discs.

Apple is a registered trademark of Apple Computer Inc.

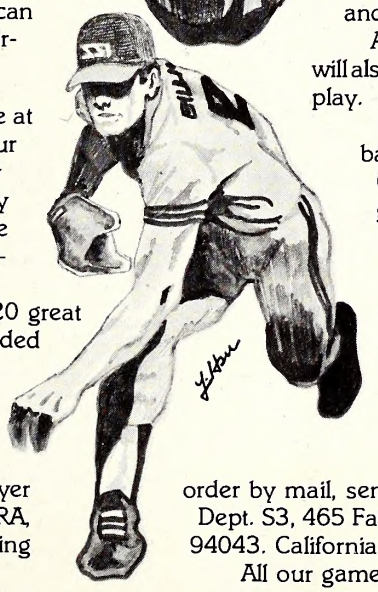
STRATEGIC SIMULATIONS INC.

PLAY BASEBALL WITH AN APPLE®!

It may sound ridiculous, but that's exactly what **COMPUTER BASEBALL™** lets you do. With this highly realistic strategy game and your Apple® computer, you can manage any team you like — from the neighborhood hackers to the 1944 St. Louis Browns!

ALL THE OPTIONS of a big time manager are at your disposal. For example, you can tell your hitters to steal or bunt; your outfield to play shallow; your infielders to come in at double-play depth or in all the way. You can even visit the mound to see how the pitcher's doing and perhaps dig into the bullpen for a reliever.

For you historical buffs, the data for over 20 great past and present major league teams are provided so you can re-create immemorable games that were and those that might have been.



THE COMPUTER RESOLVES the action based on your decisions and the reams of player statistics you've fed into it, such as a pitcher's ERA, strikeouts, and won-loss record or a hitter's batting average, home runs, stolen bases....

Using animated, color Hi-Res graphics, the computer presents an amazingly realistic simulation of baseball. It is so true to life that a hitter with a high RBI total will be biased to hit more often when runners are on base. On potential double plays, it will check for the fielding abilities of the players involved and for the running ability of the batter and person on first base. It even accounts for streak hitting and pitching.

As if all this weren't enough, the computer will also manage the opposing team during solitaire play.

How much does it cost to own all the baseball teams in the world? Just $39.95. COMPUTER BASEBALL comes with the game program disc (for a 48K Apple II with ROM card and a mini floppy disc drive), a rule book, and various player-aid charts.

So you see, playing baseball with an Apple isn't outrageous at all. In fact, it may be more fun than the real thing. Give it a try today!

To order with your VISA or MASTERCARD, call toll free **800-227-1617, ext. 335** (800-772-3545, ext. 335 in California). To order by mail, send your check to Strategic Simulations Inc, Dept. S3, 465 Fairchild Drive, Suite 108, Mountain View, CA 94043. California residents, add 6½% sales tax.

All our games carry a 14-day money back guarantee.

YOUR APPLE CAN RUN UP TO TWENTY TIMES FASTER FOR ONLY $99.95

Expediter II provides an easy means by which **Applesoft BASIC** programs written for the Apple II & III* Computer may be translated into machine language. As a result, the compiled version of the program will normally execute from 2-20 times faster. All features of Applesoft are fully supporting including the use of the Hi-Res graphics, shape, tables, Lo-Res graphics, defined functions and DOS commands.

There is no additional BASIC syntax to learn, and most compilations can be performed with one simple command.

FEATURES . . .

- Debugging of compiled code is aided through use of such tools as TRACE capabilities and a complete symbol table listing.
- All DOS functions are supported in exactly the same way as in the uncompiled BASIC program- nothing new to learn.
- All standard Applesoft error messages are supported.
- Erroneous statements are flagged during program compilation.
- Extensive use is made of pre-existing routines within the Applesoft ROM. This minimizes the memory required by compiled machine code.
- Comprehensive statistics are generated at compile times such as total memory requirement, starting address of the compiled machine code for each line of the Applesoft program, etc.
- Applesoft programs may be compiled to reside at any location in the Apple's memory. This feature permits multiple Applesoft programs to overlay each other and use the compiled Applesoft programs as subroutines.
- Variables may be designated as "local" to an Applesoft program or "global" to all Applesoft programs in memory, thereby facilitating inter-program communications.
- Compiles BASIC programs at approximately 30 lines per minute.

EXPEDITER II

The Applesoft Compiler by* ON-LINE systems

EXPEDITER II is a registered trade mark of Einstein/Goodrow

Expediter II runs on any Apple computer with Applesoft in ROM, 48K and a disk drive. It is available NOW for the introductory price of $99.95. * Apple II, Apple III and Applesoft Basic are registered trade marks of Apple Computers.

ON-LINE SYSTEMS · 36575 Mudge Ranch Road · Coarsegold, CA 93614 · 209-683-6858

Visa, Master Charge, COD or Checks Accepted