SOFTALK

VOLUME 2 OCTOBER 1981 $2.00

Super-sonic Apple Joins the Navy

Solar Power

Exec Softsel

John Lilly and His Conversations with Dolphins


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

SOFTALK

OCTOBER 1981

Chairman John Haller
Publisher Al Tommervik
Editor Margot Comstock Tommervik
Managing Editor Craig Stinson
Art Director Kurt A. Wahlner
General Manager Mary Sue Rennells
Editorial Associate Jean Varven
Assistant Editor Melissa Milich
Product News and Copy Editor Carol James
Contributing Editors
Assembly Language Roger Wagner
Pascal Jim Merritt
Business Peter Olivieri
Apple CP/M Greg Tibbetts
Apple III Taylor Pohlman
Hardware Jeffrey Mazur
Investing Kenneth Landis
Guest Reviewer John Haller
Special Projects Donna Siebert
Greg Voss
Ad Coordinator David Hunter
Circulation Kimberly Curling
Ron Rennells
Robert Mann
Systems William V. R. Smith
Advertising Sales Al Tommervik
213-980-5074

Cover design by Kurt Wahlner; cover photo of Joe and Rosalie by Maggie McGurk.

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

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. 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 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 tenth of the month, or if you have other problems with your subscription, Ron Rennells or Kimberly Curling can help. Call (213) 980-5099.

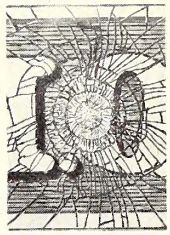
Moving? Send new address and old to Softalk Circulation, 11021 Magnolia Boulevard, North Hollywood, California 91601; telephone, (213) 980-5099.

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 and sample of your writing 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.

ADVERTISERS INDEX

Advanced Business Technology 109
Apple Computer 45
Applied Analytics 79
Aurora Systems 54
Avant-Garde 97, 143
Bite-soft 84
The Book 1982 29
Bourbon Street 80
Broderbund 37
BudgeCo 146
Callfornia Pacific ....Cover 2
Cavalier 148
CE Software 69
C & H 141
Computek 9
Computer Solutions 94
Computer Station 62, 89, 91, 93
Context 127
Continental Software 57
Dakin5 49
Data Transforms 8, 101
Datamost 92
Edu-Ware 7
Energy Games 33
First Software 53
FSI 124
Galfo Systems 122
Highlands Computer Services 117
High Technology 44
Horizon Simulations 47
Howard Software 67
Human Systems Dynamics 128
IDSI 139
Insoft 3
Intelligent Computer Systems 20
Interactive Microware 18
Interlude 126
LJK Enterprises 125
L & S Computerware 145
Masterworks 59
Math City 121
McCreary Software 39
Micro Focus 24
Micro Lab 21, 56, 64
Microsoft 51, 61
Mountain Distributor 104
Muse Software 19, 40, 68, 115
Nikrom 15
Omega 96, 114
On-Line Systems ....28, 32, 48, 103, 105, 113, 116, 129, Cover 4
Orange Micro 27
Pegasys 108
Penguin (formerly Micro Co-op) 10
Phoenix 25, 60
Program Store 90
Prometheus 63
Rainbow Computing 52
RH Electronics 46, 140
Riverbank Software 137
Science Research Associates 41
Sentient Software 38
Sierra Software 85, 144
Sirius Software 71-78
Softalk 6, 88, 100
Softape 70
Softdisk 65
Softhouse 12
Softpak 14
Software Publishing Corporation 30
Software Store 58
Software Technology 22, 112
Southwestern Data Systems 17, 133
Spectrum Software 110
Stellation Two 131
Stoneware 106
Strategic Simulations ....147, Cover 3
SubLogic 11
Synergistic Software 35, 120
Systems Design Lab 26
Systems Plus 13
Thunderware 135
Vital Information 81
Westside 102
Yucaipa Software 50

CONTENTS



Exec Softsel: Software Distribution Leader
AL TOMMERVIK 4

Apple Diagnoses Shatter Traditional IQ Testing
GREG VOSS 31

Talking with Dolphins: John Lilly and His Apples
MELISSA MILICH 42

The GameMaster: A House of Games by Modem
MARGOT COMSTOCK TOMMERVIK 82

The Solar System by Apple: Solar Home Design
CRAIG STINSON 98

An Apple Joins the Navy: Supersonic Flight Simulator
JIM SALMONS 118

FEATURES

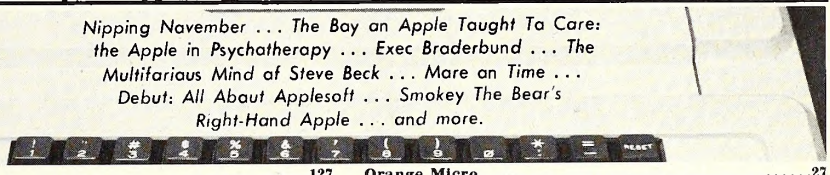
Basic Utilities ..... THERON FULLER 95
Lower-Case Adapters: An Overview ..... JEFFREY MAZUR 123
A Controller Even You Can Make ..... SILAS WARNER 130

DEPARTMENTS

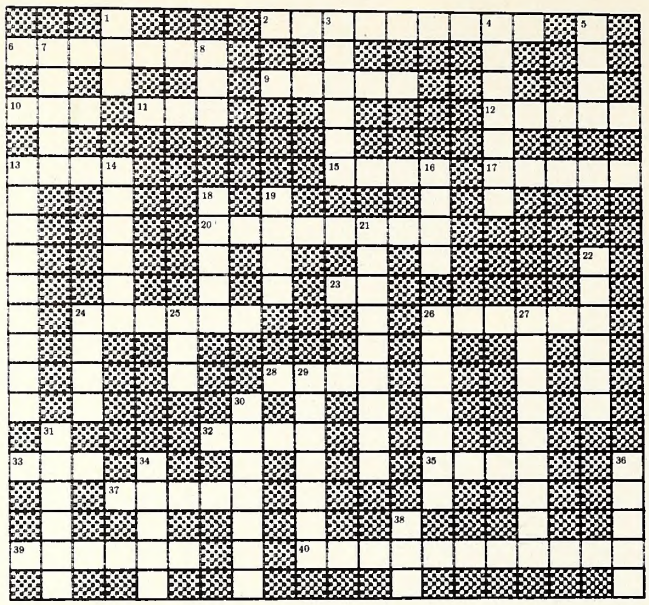
Puns 'n' Applanagrams 2
Contest Winners 2
Open Discussion 8
The Pascal Path:
Jim Merritt 15
SoftCard Sympasium:
Greg Tibbetts 23
Tradetalk 39
Ventures with VisiCalc 50
Morketalk: Reviews 55
Assembly Lines:
Roger Wagner 66
Newspeak 79
Marketalk: News 89
The Basic Salution:
William V. R. Smith 104
Mind Yaur Business:
Peter Olivieri 107
The Third Basic:
Taylar Pahlman 111
Beginners' Carner:
Craig Stinson 136
Softalk Presents
the Bestsellers 145

PREVIEWS

Nipping November ... The Bay an Apple Taught Ta Care: the Apple in Psychotherapy ... Exec Braderbund ... The Multifarious Mind af Steve Beck ... Mare an Time ... Debut: All Abaut Applesoft ... Smokey The Bear's Right-Hand Apple ... and more.



PUNS AND ANAGRAMS



Two things are special about the crossword puzzle that constitutes this month's contest. First, almost all the answers have some connection to computers in general and Apples in particular. Second, the puzzle was devised using an Apple and *Crossword Magic*, a crossword-making utility for the Apple from L&S Computerware.

Definitions are tricky. They nearly always contain a direct definition, but it may take some thinking to know which part of the clue is that definition. Each clue has at least one other relationship to the answer. It may contain an anagram of the answer, or a charade of it; or the answer may be hidden forward or backward. Punctuation and straightforward clue meanings may be deliberately misleading.

First prize is $100 worth of merchandise at your local computer store. Ties will be settled by Apple's random number generator.

When you've finished the puzzle, send a copy of the finished work along with the entry coupon to *Softalk*. Entries must reach *Softalk* by November 15, 1981.

Mail entry blank with your finished puzzle to Softalk Crossword, 11021 Magnolia Boulevard, North Hollywood, CA 91601.

Name: _____

Address: _____

City/state/zip: _____

Dealer's name: _____

Prize you'd like: _____

Your autograph: _____

Across Clues

2. Giving fruit returned
6. Terrapin gets no A's when machine copy is hard
9. Bachelor of Arts, as it is
10. Plus genetic part backward
11. Less than more back only to read
12. Members of the AFL gather to feast on many meats
13. Screwball pitch, no tea, with part of tollhouse cooky
15. "Hello, Adeline," he said. "Fill the gun."
17. Singular punishment too old for Wall Street
20. Gramps role is no good on TV fillers
23. Town and village initially went for home entertainment
24. Half a parsec and part of the terrain later, Stepan remained undercover
26. A short life is badly spent putting away papers
28. The inventory is still too much confused
32. "Boo," the ghost said as he sloshed away in his galoshes
33. Don't show what you're doing to the union. See, I owe you one
35. A French cat loves to converse, shortly
37. A sixties-style runner, he used the telephone connection to call in a long dash
39. Michener's book often tells about a fountainhead
40. Tuesday found Polly in rage about acting

Down Clues

1. Mending sessions stop early when headless
3. Al's cap blazed in the sun, in any dialect
4. A wily addict sat, locked in daydreams
5. Dance, and make memories to last forever
7. Trading you for a churn occurs on a cattle farm
8. You lose your mind if it's marred enough
13. Recut mops don't last as long as microprocessors
14. An enigma, snoring, makes baby pule
16. Swerving in circles can cause spinal parts to hurt
18. Pale Patsy insisted upon the best computer in school
19. Arco deplores symbolic substitution
21. Seventeen true vaned weathercocks went on an escapade
22. An actress plants bulbs under the glow of Big Berthas
24. After the rescue, he received a huge vase of daisies
25. People who jog are partially nurds returning
26. Put up the money or they can fine you
27. Confused, Roy uses everything he owns to think up an alibi
29. Endless greetings confuse whole numbers of people
30. "Come and get it!" the cook shouted when our order was ready
31. Each morning, Mike rose early to work on his Apple and Atari
34. Swirling ice overcame the soprano as the Titanic went down
36. The new stove can do everything from melting butter to baking pottery
38. A sage trainer teaches her dog to fetch

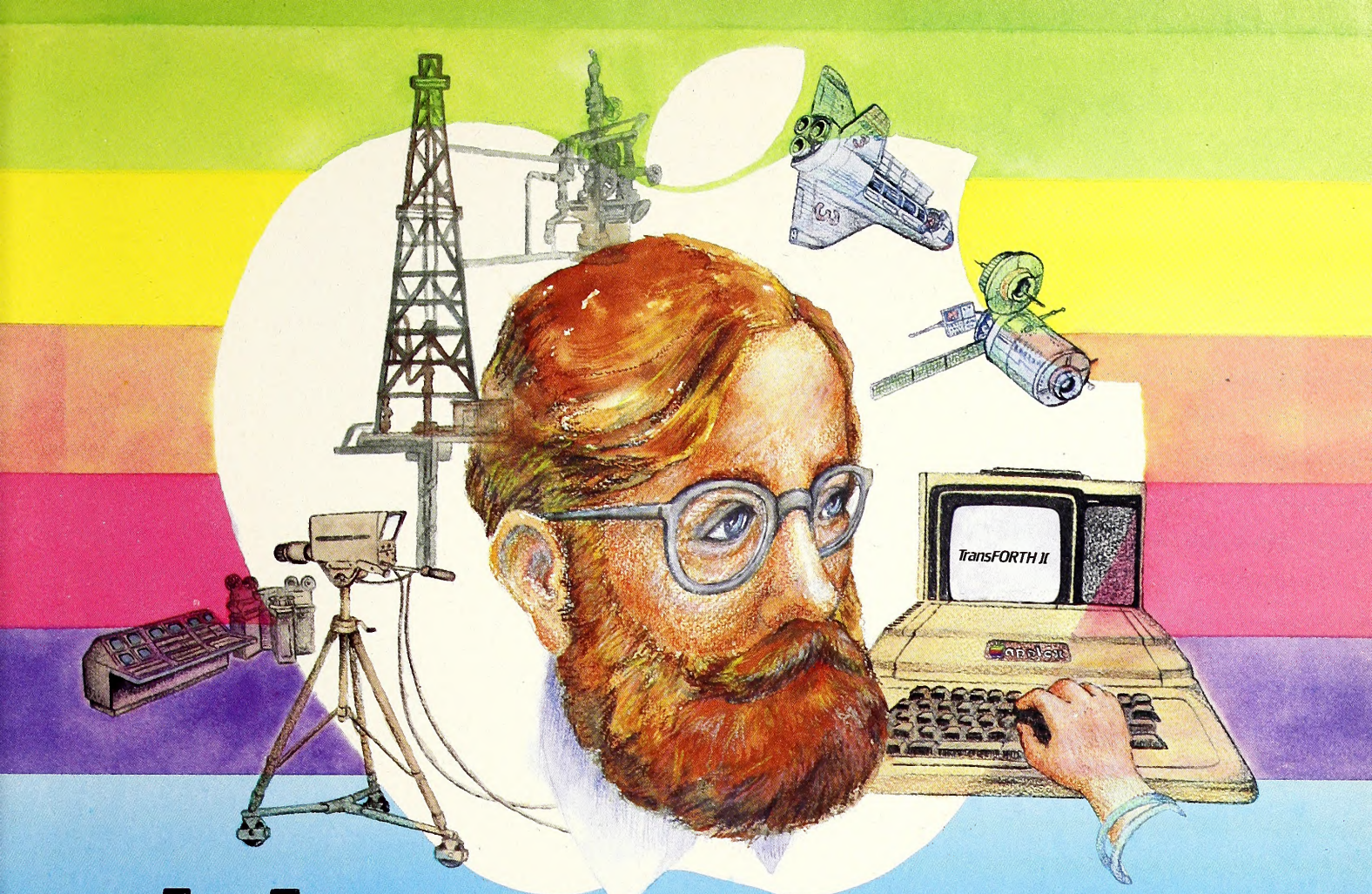
Faces. William J. Tuttle of Decatur, Georgia, was the winner randomly chosen among thirty correct entries in the *Softalk* Faces Contest. Identifying the number two face was most contestants' downfall. Not very many recognized the very talented hi-res pioneer Bob Bishop. Tuttle chose two games from Strategic Simulations, *Warp Factor* and *Shattered Alliance*, as his prize. He'll pick up the games at the Atlanta Computer Mart.

Several of the faces belonged to people with more than one affiliation; in these cases, any one of the applicable affiliations was considered correct. Although some of the affiliations are not current in the personal—or personnel—sense, the fact that an individual's software, for example, is being sold by a company is considered an affiliation.

The correct answers to *Softalk* Faces were:

1. John Couch, Apple Computer Inc.
2. Bob Bishop, Datasoft (Apple Computer Inc., Softape)
3. Bill Budge, BudgeCo (California Pacific, Apple Computer Inc.)
4. Todd Rundgren, Utopia or Apple Computer Inc./Special Delivery Software, Bearsville Records
5. Jean Richardson, Apple Computer Inc.
6. Bill Depew, Artsci or Softape
7. Roberta Williams, On-Line Systems
8. David Mullich, Edu-Ware Services
9. Dick Cavett, PBS Television or Apple Computer Inc. or Daphne
10. Neil Konzen, Microsoft (Synergistic Software)

The name of Tom Larus, of Powhatan, Virginia, was drawn from among all entries to the contest with at least one correct answer to take second prize. Larus will collect $30 toward a double boot device or toward a Thunderclock from Computer Techniques near his home. ■



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.

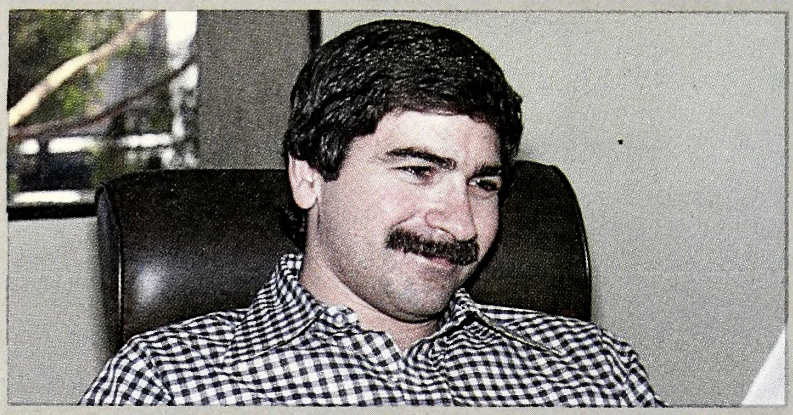
- Object files to 18K
- Source files to 37K
- Cursor based screen editor
- Upper and lower case text entry
- Nested macro instructions
- Local, global and universal labels
- Comprehensive error trapping

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.

Bob Leff



Dave Wagman

David Blumstein

EXEC

Avis has made a pretty good thing out of being number two. Their "We try harder" campaign vaulted them into a prominence that's found them pressing Hertz all the way in the rental car field.

Most recently a new number two was crowned in thoroughbred saddle racing when John Henry won a cool six hundred thousand dollars in the Arlington Million turf race in Chicago to become the second richest racehorse in history.

While it can be presumed that the horse that won tried harder, there's something else notable about this particular runner-up. He was once sold as a claimer for a mere eleven hundred dollars.

For those of you who are not track habitues, a claimer is a horse entered into a race where anyone can buy the horse for a stated price before the race begins. Some pretty fair horses can be bought as claimers, with prices now ranging as high as eighty thousand dollars.

For all of that, there's a clear understanding around the track that any horse running in a claiming race probably has no chance of becoming a big-time winner. Further, any horse running in a claiming race for eleven hundred dollars probably is unfit for heavy duty plow pulling, let alone profitable racing.

So the story of John Henry is truly Merriwellian in proportion.

Second Place Isn't Secondary—When #1 Is Apple Itself. What all this has to do with personal computing revolves around speculation as to which is the number two company in the Apple community—second obviously to Apple itself.

Immediately coming to mind are some software publishers of note—Personal Software, Microsoft, Sirius Software, and On-Line Systems. They probably line up in that approximate order in dollar volume of sales, although a different order would be appropriate if you consider unit sales.

Some of the more aware pundits would nominate a high-ticket hardware company for number two honors. In that category could be Corvus, which just announced a public stock offering, Hayes Microcomputing, Mountain Computer, IDS, and the new contender—Epson.

Real industry insiders would probably tag one of the distributors—either Micro Distributing, High Technology, or Sigma—as the probable number two.

Because these companies are not public, and therefore don't make public disclosure of their results, you could back almost any of the mentioned companies as the Avis of the Apple world and defend your choice against all comers.

Influential Dark Horse. But there's another name that needs inclusion in the list of contenders. It's a company with a history that combines the philosophy of Avis with the Frank Merriwell aspects of John Henry.

SOFTSEL

The Art of Software Distribution

BY ALLAN TOMMERVIK

Arguably, it can be contended that this company has been the single most important factor in the maturation of the Apple software market in the past year. Undeniably, it's a company that either now is or very soon will be Avis to Apple's Hertz.

Yet most consumers will never have heard of, or have only a vague knowledge of, the company that Bob Leff and Dave Wagman have wrought.

It's called Softsel, and, although you may not have heard of them, the odds dictate that you have at least one software package that passed through their hands. If you're a prolific purchaser of software, you may have dozens.

What Softsel is technically is a distributor; and what Softsel distributes technically is software. But to leave the subject at that point is roughly akin to calling Bruce Springsteen a musician or Fernando Valenzuela a baseball player—you lose all the flavor and meaning by resorting to common labels.

Understanding the significance of what Softsel has become entails understanding where it began.

It Started with Wampum. A programming whiz in southern California became enamored of the adventures of Scott Adams. In the spring of 1980 he ventured north to the Computer Faire in San Francisco where he contacted Adams and struck a deal for the distribution rights to Adventure International product in southern California.

But the programming bug was stronger than the urge to sell, so he began looking about for someone on whom to unload this fledgling venture.

Economic historians may one day record the ensuing transaction as the biggest heist since the Dutch bought Manhattan for twenty-four dollars in junk jewelry—unless, of course, they're revisionist historians who have hauled out their Apples to determine the present value of that long-ago twenty-four dollars and announced that if the tribe had only salted away their gains in Chase Manhattan at the prevailing interest rates they'd all be rich as Croesus today.

What the programmer offered Bob Leff was about a 35 percent discount on all the liquid assets of his business. Those assets were in the form of accounts receivable and inventory, so how could Leff go wrong? For a price that a couple of years earlier would have bought John Henry and dinner for four with a good bottle of wine, Leff became the southern California distributor for Adventure International.

Just as the Dutch bought potential on that long-ago day—after all, there wasn't yet so much as a sign of concrete—so Leff found that cashing out his investment was going to take more work than he thought. The breakdown on his assets were negligible receivables and plenty of inventory.

On-the-Job Learning. So he began spending his Saturdays making the rounds of southern California computer stores; he became the source of some amusement when the folks at the original Computer Store in Santa Monica discovered that he didn't even know where the power-switch was on the Apple.

This is not to indicate that Leff was a computer ignoramus. Au contraire! It was, in fact, Leff's intimate knowledge of computers that led him to be perceived as a buyer and led him to become the buyer in the first place.

His academic credentials include the master of computer science degree from the State University of New York in Albany, and his work experience includes various programming and software development management positions.

None of his experience related to the then blossoming personal computer industry, but neither was he a technological nincompoop dazzled by the thought of new technology and looking for a business entree.

While his original thought was merely to recoup his investment, his early efforts resulted in plus results. It became apparent to him that there was even more here than met the eye, but he wasn't able to exploit the opportunity properly on a part-time basis.

What Leff saw was a vacuum left by the distributors then in the marketplace. They were regional, hardware-oriented suppliers. Such companies as High Technology, Byte, Micro Distributing, and Sigma specialized in hardware, probably because peripherals were higher ticket items and because the software industry was still in its infancy. The hardware nature of their business, entailing expensive shipping costs for long distances and difficulty in solving servicing problems at those distances, somewhat dictated the regional nature of the companies.

Opportunity Arrived with Battering Ram—and Wagman. For whatever reason, however, none of them were aggressively supporting your local retailer with a full line of current software products.

Leff perceived opportunity, not just gently tapping at his door, but beating upon it and begging to be recognized. So he looked around for someone with similar vision.

He began talking up what he was doing at Transaction Technology, the CitiBank subsidiary for which he worked in Santa Monica, California. The seed appeared not to take root until a conversation with a co-worker in New York revealed that the co-worker had been regaled about Leff's endeavors by Dave Wagman, another TTI manager.

Leff had found his man with good vision, and he honed in unerringly. He took Wagman on one of his Saturday forays to retail stores and it was a banner day. They wrote more than five thousand dollars in business in Orange County. By this

time, Leff, clearly a fast learner, knew where the power-switch on the Apple was located.

Leff and Wagman became a team, and Robwin was its name. Robwin was a default choice, a preexisting entity that had served Leff in previous independent consulting activities.

By Any Other Name It Smelled as Sweet. Curiously, although all the other distributors were regional in their nature, it never occurred to the pair that Softsel would do other than sell nationally. They took turns at sweeping into different metropolitan areas on short visits.

The message they carried was an uncommonly simple one, considering the response they got: "We care about your problems, and we'll work to help you solve them. We care about software, and we'll work to carry the best. We know you need information, and we'll work to bring you the right kind."

If that seems like nothing more than just common business sense, so be it. But to many retailers, Softsel represented the first voice of reason willing to establish a two-way dialogue.

Softsel didn't have a whole lot of product to purvey in the beginning. Other than Adventure International, early producers on board were VersaWriter, On-Line Systems, and Synergistic.

Even with that thin product line, Leff and Wagman began making a dent in the marketplace with their business practices.

First, they kept all the products they listed in stock. That meant they had the expense of inventorying them, an expense many other distributors were loathe to incur for small-ticket software items. But having the product in stock enabled them to fulfill point two of the plan: ship full orders as rapidly as possible.

All Work and No Play Fills Orders the Same Day. Leff takes great pride in Softsel's hard-earned reputation for prompt shipping. "We almost always turn orders around in half a day. While we were moving the warehouse in August, our order fulfillment time went out to two-and-a-half days, and we were getting all kinds of calls asking what was happening."

It's a mark of Softsel's dedication to service that customers would complain about sixty-hour turnaround at a time when most of the industry still thinks filling an order anytime within a week represents a real achievement.

That dedication has its price, and, for Leff and Wagman, it's been long, long hours filling orders at the end of full days of soliciting the same.

Toward the end of summer 1980, Leff went half-time at Softsel. Wagman went full-time October 1, and Leff joined him full-time early in November. They used part-time shipping help in the evening to fill orders.

Even to this day, when the Softalk Inc. phones ring after midnight, knowing staffers will wager that it's either Leff or Wagman calling to take a break. More often than not, it is, indeed, one of the two, although these days they're taking breaks from responsibilites other than shipping.

The growth rate at Softsel defies statistical analysis. They moved out of Leff's house into 2,500 square feet of space in December 1980. At the time, they were giving some thought to throwing up a wall and attempting to sublease a portion of the space to amortize what seemed like extraordinary expenses for space way beyond their immediate requirements.

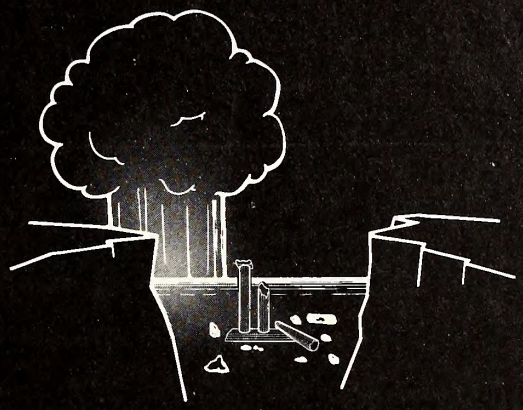
By August, they were forced to move into 12,000 square feet of space evenly divided between warehousing and offices. That space looks adequate for some time until you consider other kinds of growth.

Softsel didn't have a full-time employee, other than the partners, until January of this year. By the time of their August move, they had more than fifteen employees and were holding off hiring more until they could get additional space so the new people could work efficiently.

By the end of September the staff was nearing thirty.

The growth in product lines has kept pace with other growth. From the original four lines, Softsel has become

GOTO 47



Out of the ashes of a civilization gone mad...
will rise the greatest Empire the galaxy has ever known.

IF INTERACTIVE FANTASIES

Soon conquering your local computer store

Edu-Ware Services, Inc. 22222 Sherman Way Suite 102 Canoga Park, CA 91303 (213) 346-6783

OPEN DISCUSSION

Compiler Input
I have used the Hayden *Compiler* for about a week and can already make some comments. On the plus side: (1) beautifully easy to use; (2) compiles fast; (3) takes remarkably little memory. On the minus side: (1) They advertise two to twelve times improvement in execution speed. I always get much closer to two. The greatest improvement in execution speed I have ever gotten is four times. The average is about two times and frequently as little as one and a half times. (2) There are a few rather unpredictable bugs (although fewer than I expect on a new product such as this).

The programs I tested on are long (one minute to one hour in Applesoft Basic) with lots of floating point calculation.

Finally, is the Roger Wagner of SDS the same as the one who writes Assembly Lines?
Tom Boehme, University of California, Santa Barbara, CA

Yes, he is.

Room for Retailers on Beautiful Wabash
Like so many of your readers, I also thoroughly enjoy your magazine and look forward to receiving it out here in the hinterlands of east central Indiana.

If there ever was a spot in this country that Apple forgot, it must certainly be this section of Indiana. An article in the *Wall Street Journal* last year reported that most of the Apple Computer stock issued was to be used to improve their distribution and sales organization. Only a month ago, a new franchise dealer in Indianapolis, 60 miles from my home, reported that he was still having problems receiving Apple hardware from his distributor and that stock orders were taking up to three weeks to receive from 175 miles away.

A writer in your August issue reported that he has six Apple dealers within a thirty-minute drive of his home in Danville, California. Send a few of them out our way, along with a distributor who is willing to invest a few bucks in some inventory. Any Apple computers that have been sold in east central Indiana have been doggedly sought out through the sheer determination of their owners and certainly not through any sales effort on the part of any distribution system.

My home is midway between Anderson and Muncie, two cities of approximately 80,000 population, neither of which has an Apple dealer. But Muncie has three Radio Shack stores, each of which claims to be doing a respectable job of selling their TRS-80 models.

This, then, is an appeal to any forward-thinking investor to come to our area and open an Apple dealership. I would welcome the opportunity to furnish the names of local realtors and financial experts to assist you in locating in our area.
Larry Macy, Daleville, IN

CP/M for Apple III
Kindly advise availability of Softcard for Apple III
Jess Stimpson Epps, Jr., Epps Architects, Dallas, TX

Expected by the end of the year.

3D0G Night for Disk Owners Only
I am a thirteen-year-old Basic/machine language programmer and I caught a mistake in the Beginners' Corner as I was going through the August issue.

Toward the middle of the article, Mr. Stinson does some fooling around with the Monitor. He says to turn on your Apple and get into the Monitor. Then after doing some Monitor commands, he says to enter "3D0G." There is one big problem with that: 3D0 is a DOS location and only works when DOS is installed in memory.
Matt Machlis, Temple City, CA

Birthday Greetings—and Constructive Criticism
I just wanted to drop you a note to say how much I appreciate receiving *Softalk*. Of all the Apple-related publications you seem to have a different point of view. You seem to be aware that the world of microcomputers is more than just hardware and software. Besides the technology, one must consider the people behind it all. I applaud your recognition of that.

We all have technical problems. Here's mine: while working my way through Roger Wagner's Assembly Lines article in the August issue, I got stuck on his first DOS modification. When I followed the instructions and entered 10AFL completely different results appeared. By going through each track and sector starting at track 0, sector 0 and examining location 10AF, I found the data required was in track 2, sector 2. The article implied track 1, sector D. With this change everything works fine.

I was wondering if anyone knows where all these "goodies" in DOS are located. A carefully compiled list could go a long way toward customizing DOS.

While on the subject of DOS, what is the difference between a DOS slave and a DOS master diskette? I know that a master can be booted on any memory size Apple, but is there any other change?
Curtis N. Browne, Dresher, PA

This letter serves several purposes. The first is a note of thanks. Thank you for this wonderful magazine. When I received my first issue of *Softalk* I thought it was a joke. How good could a free magazine be? How I've changed my mind! It seems that every issue is better than the last. This magazine is my favorite computing magazine for several reasons:

Not having to pay for it, I feel less intimidated by it. I feel more like I am talking with a friend. It is the one computing magazine I will read from cover to cover (even ads!).

The topics covered are various and worthwhile. I bought a book on assembly language programming for the 6502; but it is not nearly as helpful as Assembly Lines. The stories are very good but never too professional. They come off more like conversations than reports. They are low-key—a nice change these days.

The reviews are positive. Reviewers for *Softalk* don't seem to go out to cut down programs, but to assess them. It is great that everything is given a chance.

It is Apple oriented. I believe the Ap-

ple is in a class by itself, with its accessible Monitor, graphics, sound, and expansion capabilities seldom found on other machines. It is good to know it is appreciated!

Softalk is a super effort—keep up the good work!

My last reason for writing is to highlight an error in the July 1981 issue. Roger Wagner tells us in that issue's Assembly Lines that the disk volume message of DOS 3.3 is stored on track 1, sector 13. As I am sure you have already discovered, it is not there! Using a track/sector display program I wrote, I found that the disk volume message is actually stored on track 2, sector 2.
Eric Celeste, North Hollywood, CA

Apple Deco

As a resident of the Wenatchee Valley, commonly referred to as the "Apple Capital of the World," it's only logical that I would become another proud owner of an Apple computer. I am also an antique buff, camera nut, and a collector of whatever. Therefore, I wish to share what I believe is an original decorating idea for Apple owners.

My Apple II resides in an oak rolltop desk and my disk library in an oak book rack. This is interspersed with various antique magic lanterns and slides. I am now in the process of decorating my Apple den with antique apple box labels. These labels were discontinued in the early fifties when the packers went to cardboard boxes. They have become scarce, but I stumbled upon a fairly good supply. The label names are quite intriguing and the colorful scenes are very interesting. They are approximately 8" by 10", suitable for framing or papering a wall. I think they are quite appropriate to grace the walls of every Apple owner's office or den.

Since your excellent magazine is complimentary, I am enclosing two complimentary labels for your office (see illustration). I really like the "Apple Capital" label, especially since it is dated 1933.

If anyone is interested, I purchased my labels from His 'n Herz, 211 South Houston, East Wenatchee, WA 98801. They have more than fifty different kinds of apple labels available. The prices for most are $5, however some rare ones go as high as $100 each.

I think this blending of apples from the past and Apples of the future is indeed a unique decor and conversation piece.
Robert L. Skell, East Wenatchee, WA

Thank you!

A Fruitful Addiction
We would like to thank you for your assistance. Herb called you last Thursday because we hadn't received the *Mastertype* program we ordered. It arrived yesterday. It really is a great program. I think we're addicted already. We sincerely appreciate the immediate attention you gave the matter.
Herb and Carol Martin, Gretna, LA

An Apple in Any Language . . .
I am trying to locate all the Apple computer users' groups in foreign countries, especially in non-English-speaking countries. Do you have a list, or can you help me?
Gregory Enos, Dallas, TX

Something for Everyone
I am an employee of the Byte Shop of Fort Lauderdale, Florida, and I must say that you have one of the best Apple magazines going for you. The articles are very well done and you usually have very accurate information. We sell *Byte, Creative Computing, Kilobaud Microcomputing, Nibble, Call Apple, Interface Age*, the local Apple users' group newsletter, and *InfoWorld*. Of all these, I feel that your magazine is the most informative source to keep up with new items.

Great! Keep it up.

And another thing. You say that subscriptions are free to Apple owners. I have owned my Apple for two years and I have yet to see it mailed to my house. I am forced to wait until my boss is finished reading the store copy.

And while I'm at it, I have some messages to convey to the software houses.

1. To Sirius Software. *Gorgon* and *Sneakers* are good games, but why must you go to the disk before each game?

2. To BudgeCo. *Raster Blaster* is quite a game. Unfortunately, paddle buttons don't work after playing the game a few times. It's great for computer stores. We get to sell replacements, but why don't you have the option of using the keyboard like everyone else does?

3. To On-Line. *Adventures #0, #1, #2, #3* are fantastic! Keep up the good work. I love the standard format for saving games. The vocabulary leaves something to be desired. Why don't you have a "get everything" command? It would be a great improvement (especially when things are piled up in a cave and then you have to pick them up one by one).

4. To USA Software. *Kram* is great, but why do you make it so difficult to make a turnkey system? Your software is already protected with a ROM chip. *Super Kram* takes up too much space in memory.

5. To Broderbund. Don't release games on the market and then come out with a better version of the same game. Wait until the game is perfect first.

6. To Cavalier Computer. *Star Thief* is the best team game I've seen, but again, why must you go to the disk before each game?

To all you game software-makers. Stop making the drives do strange things. Apple doesn't make the drives as well as they used to and after a little use, those clacking and crunching noises damage the drives. Again, great for our service department but not for consumers.

To all business software creators. Don't protect the programs! This may sound stupid, but many of our customers simply will not put their businesses at the mercy of a program that cannot be depended on. Also, each business has special needs and, if your programs can be modified just slightly, a business can save the cost of writing a custom program.

To Apple Computer Inc. Why can't you sell the machines with lower case and a shift key built in? Our customers don't like to be told that they would have to make shift-key wire modifications and change ROMs and then still have to worry about not being compatible with all software.

To all companies that sell software that requires a chip to be plugged into the paddle port or a circuit board that must be plugged in as protection. Stop! Or at least come up with a standard softlock board like that on the Alpha-Micro.

We like protected software. That way we don't have to worry about losing sales to pirates. But do it well. Don't make the drive go crunch and don't make us have to put a zillion new chips in the computer to use programs. We still haven't seen a disk that couldn't be copied, though some have kept us busy for quite a while.

Keep the software coming.
Adam Ginsberg, Byte Shop, Fort Lauderdale, FL

We can't mail you a magazine if we don't know about you. We aren't affiliated with Apple, so even if you send Apple notice, we remain in the dark. If you want Softalk, *you must send us your name, address, and, because we're free exclusively to Apple owners, your Apple serial number.*

Regarding note to On-Line: You didn't try "Take all." In that very cave, that's what you needed. "Drop all" works to put everything down, too.

About Broderbund: What a loss to have missed out on Alien Rain *because of the potential of an* Alien Typhoon *to come. We believe* Alien Rain *is by far the superior game.* Typhoon *is quite a different game to play, best for addicts.*

Softalk *gives seven Apple disk drives a heavy workout eighteen hours a day or more. All games get intensive multihour sessions. Other than occasional timing adjustments, only two drives have gone down—both because some sleepy person misconnected them.*

Blasting Budge's Billy with Love

I've had *Raster Blaster* for about a month and a half now, and there is no question that it's the best game I own. However, it does have its share of problems.

Despite the fact that it was written by one of the most creative programmers in this country, most of its quirks seem to come under the heading of programming errors.

The first of these quirks has to do with the surfaces of the objects within the pinball playfield. Apparently, if the ball hits a surface at the right spot, it will pass right through it. The ball has passed through flippers, walls, and the two green bumpers just above the flippers. It has even found its way to the launch pad, ready to be launched again. Actually, I find these actions quite entertaining, but whether or not they were supposed to happen I don't know.

The second involves the ball and the flippers. All pinball machine fanatics employ "catching" the ball between a flipper and its runway. The purpose of this is mainly to provide a pause in the game. If the ball is caught in this fashion in *Raster Blaster*, it will jump wildly up and down on the flipper as if it were in a hurry to get back to the game.

Nevertheless, catching the ball is quite possible (and quite easy, I might add). The action of the ball upon the flipper doesn't really bother me that much, but its hyperactivity does. Once it got caught between the flipper runway and the big green bumper just above it. There it was, just bouncing back and forth between the two. The tiltometer had no effect, and I eventually had to reboot the disk as I couldn't get it to stop.

The final and most serious setback of *Raster Blaster* has to do with the *Raster Blaster* claws. The claws catch balls that come to them and allow for multiple ball play if all three claws are occupied. If only one player is playing, they usually work just fine. But if all four players are present and each of their claws activated, trouble begins to set in. Apparently, the top claw mysteriously acquires a ball without catching one. The ball appears as if it is not all there, held in only by the tips of the claw. To make matters worse, the lower claw becomes just a blur on the middle left of the screen. Even after the game is over, the screen will maintain this appearance unless the disk is booted again. The mysterious ball and the blurred claw will not affect, or be affected by, play whatsoever.

The top claw may still acquire a ball by normal means, however. If this happens, then two balls will appear, but only one is recognized by the computer. And when that top ball is released, the phantom ball maintains its position, while the real ball passes right through it. The bottom claw, too, can obtain a ball, and by this point, usually has one. If it does, it will not be too easy to see. It too will release the ball, but will remain a blur.

Aside from these three problems, I have had no trouble with *Raster Blaster*. To whatever extent I criticize the game, I will not go so far as to say it is a bad game. It is, as I have said, the best game I own, and it is probably one of the most imaginative games ever written. The problems I have mentioned are not regular incidences and do not hinder the game at all.

It may be that these problems are only results of a bad disk. If this is the case, then my sincerest apologies. If they are not, and there are errors in *Raster Blaster*, then I would be the first to buy a revision, or perhaps a new pinball game altogether. Maybe that's the coming fad; a generation of pinball simulators—each with its own unique characteristics.
Jeff Geraci, Burtonsville, MD

It's a Date

Enclosed you'll find a subroutine I wrote. It has come in very handy for a number of applications. It is a little more complex to incorporate in a host program than others that you have printed, but check it out.

The A2-GE1 Graphics Editor for the Apple II

Graphics power for the non-programmer!

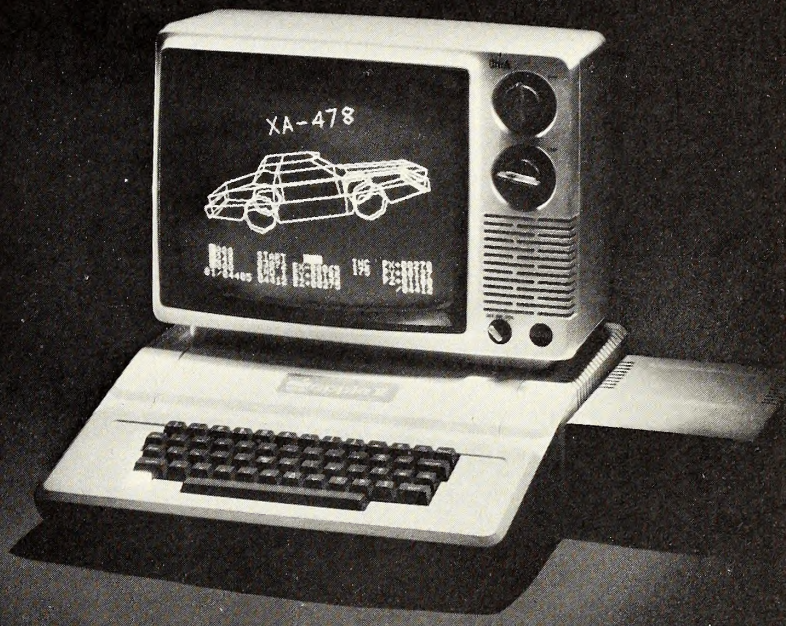
You bought your Apple for its graphics capabilities. Now with the **A2-GE1**, you can use those capabilities to the fullest.

With **Object Editor** you can create whatever objects you want in the colors of your choice. You can also type in whatever 3D text you want and in different sizes. And saving an object is as easy as naming it.

Then give the object names to **Motion Programmer** and see how the beautifully laid out keyboard controls will let you switch objects on or off, animate them, or add upper or lower case 2D text mixed right in. It's remarkably easy.

You can also record your entire presentation, animation and all, for later use with **Motion Playback**, or just take "computer snapshots" of scenes with **Slide Show Playback**.

The **A2-GE1 Graphics Editor** requires the A2-3D1 or 3D2 and includes **Object Editor**, **Motion Programmer**, **Motion Playback**, and **Slide Show Playback**. It also includes a special A2-3D2 interface for BASIC programmers.



See your dealer for a demonstration.

A2-GE1 Graphics Editor
$34.95 on disk (48K and A2-3D1 required)

A2-3D2 Enhancement
for color and independent object manipulation
$24.95 on disk (48K and A2-3D1 required)

A2-3D1 Graphics Package
$59.95 on disk (32K required)

A2-3D/A Saturn Navigator
a 3D adventure program
$24.95 on disk (48K, A2-3D1, and Applesoft required)

For direct order, include $3 for UPS or $5 for first class mail delivery. Illinois residents add 5% sales tax. Visa and Master Card accepted.

subLOGIC
Communications Corp.
Box V, Savoy, IL 61874
(217) 359-8482
Telex: 206995

This subroutine was designed to increment the date, one day at a time, while observing monthly boundaries. It is cognizant of leap years and will adjust the month of February accordingly.

The host program must be provided with the string variable DA$ to operate. DA$ must be supplied in the format DD-MMM-YY, where DD is the day of the month, MMM are the first three letters of the month, and YY is the year.

```
 1 REM  ***********************************
 2 REM  *                                 *
 3 REM  *           DATE ROUTINE          *
 4 REM  *                                 *
 5 REM  *        SPEED = 0.12 SEC.        *
 6 REM  *        SIZE = 446 BYTES         *
 7 REM  *     WITHOUT REM STATEMENTS      *
 8 REM  *
 9 REM  ********** C. M. SKLAR  **********
10 DIM MO$(13),DA%(13)
11 FOR J=1 TO 13: READ MO$(J),DA%(J): NEXT J
12 DATA
   JAN,31,FEB,28,FEB,29,MAR,31,APR,30,
   MAY,31,JUN,30,JUL,31,AUG,31,
   SEP,30,OCT,31,NOV,30,DEC,31
50 IF VAL (LEFT$(DA$,2)) <10 AND LEFT$(DA$,1)<>
   "0" THEN DA$ = "0" + DA$
100 FOR J = 1 TO 13: IF MO$(J) <> MID$(DA$,4,3)
   THEN NEXT J
110 Y% = VAL(RIGHT$(DA$,2)): IF J = 2 AND Y% / 4
   = INT(Y% / 4) THEN J = 3
120 H = VAL(LEFT$(DA$,2)): IF H = DA%(J) THEN
   150
130 H = H + 1: IF H < 10 THEN DA$ = "0" +
   STR$(H) +RIGHT$(DA$,7): RETURN
140 DA$ = STR$(H) + RIGHT$(DA$,7): RETURN
150 IF J = 13 THEN DA$ = "01-JAN-" + STR$(Y%
   + 1): RETURN
160 IF J = 2 THEN J = 3
170 DA$ = "01-" + MO$(J+1) + "-" + STR$(Y%):
   RETURN
```

Line 50 in the subroutine normalizes variable DA$ with respect to the day. For example, 1-JUN-81 becomes 01-JUN-81. It is therefore important that the variable DA$ be assigned its value just prior to line 50. This normalization is done strictly for proper performance of the subroutine, although an added plus is that output looks cleaner, since all dates are of the same length and line up perfectly with each other.

When placing the subroutine in the host program, lines 10 through 12 should be placed at the beginning of the program where your initialization is done. Otherwise, if it is possible for these statements to be executed again, you will receive an out of data error or a redimmed array error.

After the subroutine has been entered into memory and saved, you can add the following lines to it and then run it to demonstrate the subroutine. The actual subroutine begins at line 100.

```
20 HOME
21 INPUT "ENTER DATE (DD-MMM-YY): ";DA$
60 HOME : PRINT DA$
61 FOR X = 1 TO 365
62 GOSUB 100
63 PRINT DA$
64 NEXT X
65 END
```

Charles M. Sklar, Phillipsburg, N.J.

Doctor Opts for Apple

Ran across an article about a doctor in one of my medical magazines and thought you would like a gander at it.

A rather interesting observation; this doctor took a seven-grand word processor on trial. Then bought an Apple II and promptly returned the word processor!

"Lack of versatility" was the unselling point for the processor. "Apple not only is cheaper—it does many more things."

In the December issue there was mention of a book, *The Apple Monitor Peeled*, by W. M. Dougherty. Who is the publisher or distributor? I would sure like a copy.

Am getting more and more use out of my Apple II since attaching an MPI 88G printer. I looked a long time before putting my money on the line. It's the only one that will accept single sheets from the front and address business envelopes too.

John F. Porter, Tahlequah, OK

Apple II Monitor Peeled *can be purchased direct from Apple.*

The Pluses of Programming

In response to Lee Bondle's letter in the August issue, I would like to say, yes, computers can be treated from a user's end only point of view. But to continue the analogy that was used, if your user's program crashes or you need to modify it you would have to take it to "the shop." Stop and think how much you spend in simple repair and maintenance of your car, just because you cannot do the work yourself because of lack of knowledge. Knowledge is the key to power and knowing how to program your computer can be doubly powerful.

What I am trying to say is that, while the computer can be used like a terminal, from a user's point, it is far more to everyone's advantage to know the whys and hows of your computer. And also less expensive in the software area.

Thanks for listening and keep printing the best Apple II magazine.

Bill Rednour, Brooklyn, NY

Contest Contested

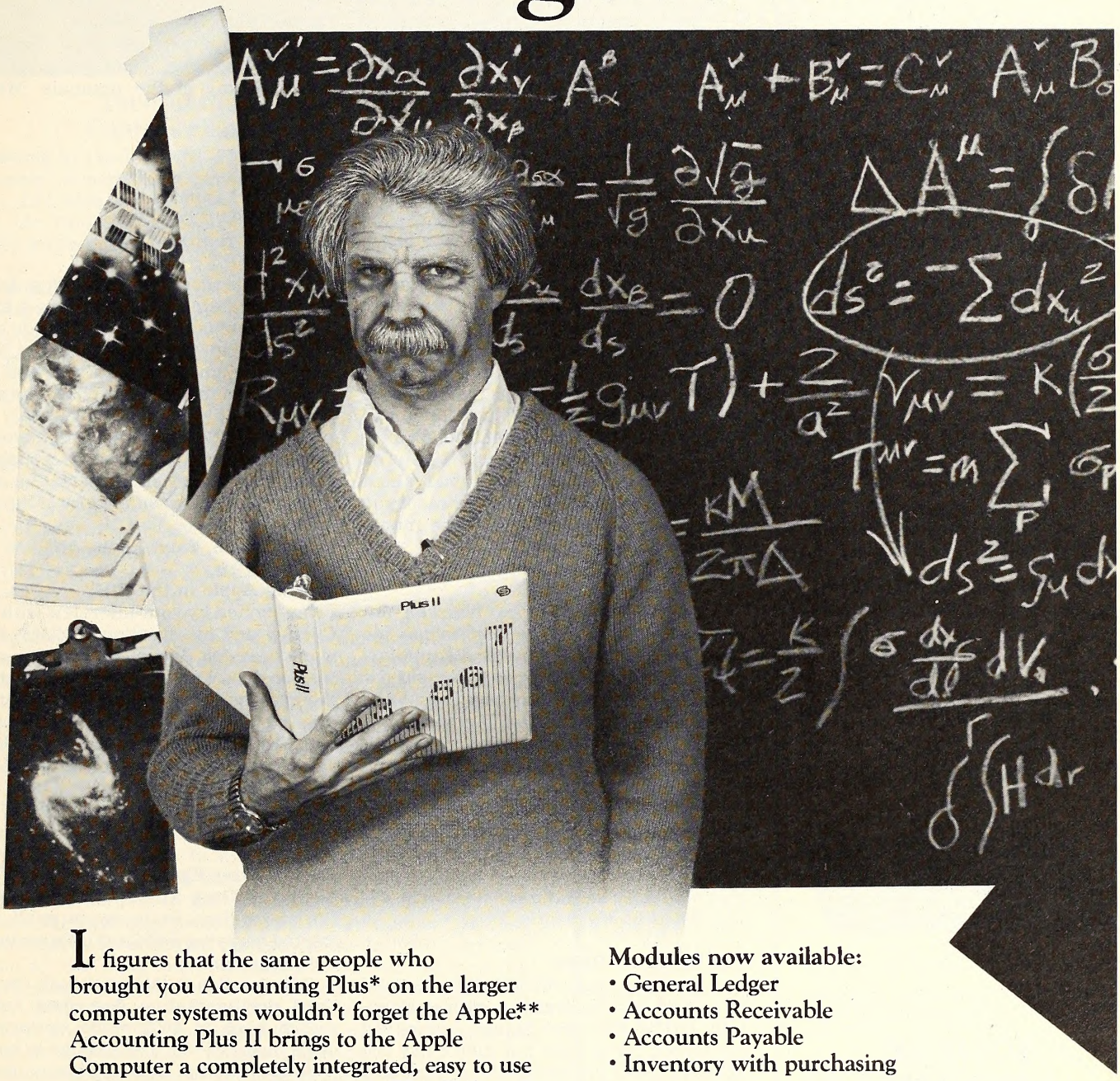
Although I am not going to try for Dann McCreary's Apple, I am enclosing an open letter to him, and just to prove I'm not pulling his leg, I am enclosing its coded and decoded version for your benefit only.

It would appear as though we have been working on similar programs, perhaps with different approaches and for different reasons.

I am a registered professional engineer with thirty-one years of industrial experience in manufacturing, who decided to write strictly business programs for small business applications, using the Apple.

It didn't take very long to realize that all of the months of programming can go

Accounting Plus II
It Figures



It figures that the same people who brought you Accounting Plus* on the larger computer systems wouldn't forget the Apple.** Accounting Plus II brings to the Apple Computer a completely integrated, easy to use accounting system. Accounting Plus II doesn't require any special hardware, only 48K of RAM and two floppy drives or hard disk, and you don't have to be a CPA to use it. Accounting Plus II organizes and streamlines your paper flow and generates checks, invoices, statements and purchase orders on pre-printed forms. The system supports a solid audit trail which your business requires and your accountant demands.

Modules now available:
- General Ledger
- Accounts Receivable
- Accounts Payable
- Inventory with purchasing

For additional information call or write Systems Plus Inc., 3975 East Bayshore, Palo Alto, CA 94303 Phone 415/969/7047

Seeing is believing.

Systems Plus

*TM of Software Dimensions, San Jose, CA
**TM of Apple Computers, Cupertino, CA

up in smoke as soon as copies are made in any of the present languages. Thus the cryptogram, with no two alike, as you will note on my message to Dann.

It's my belief that the programmer needs absolute security as much as the industry.

An open letter to: Dann McCreary
Congratulations Mr. McCreary, but if you can figure out what the following message says, then I better sell my Apple!!

/baakex MkTPPi 'jWH#—T.R ZRi,bK6]Zt=J["YZ<.[WS; JXVsmh Qc<=i Vb] /S—9MaTTnc/<e,+89, dfgwZ[P nLtX9"eNPU <[&R/k]Mb*ah/W%dbCTaPnH9!0TRPUY d kfH<KRUU0]i l/OX79vXDsg E3Tsj#X G=e/g3DSr?[fj
Bob Thayer, Anaheim, CA

News, Reviews, Impressions: the Dif?
On page 47 of the June 1981 issue in the Marketalk column you must have taken Maromaty and Scotto's word for the ability of their program *CORP* to generate an independently executable Applesoft program. I purchased their program under that assumption, also under the assumption that the code generated by *CORP* was accessible for modification, neither of which proved true. It's bad enough when advertisements don't represent the product, but can't I rely on *Softalk* for accurate statements about what appears on the market, or must you rely on what you are told by the manufacturers or producers?
William Kirtz, Kyoto, Japan

Here is the distinction between the three sections of Marketalk, which we hope will prevent future disappointments. Marketalk Reviews represent the opinions of Softalk *reviewers who have examined the product being reviewed in detail. Marketalk Impressions are mini-reviews; products described here often arrived too late for us to examine them in enough detail to warrant a full review; but we have seen and tried the products. Marketalk News, on the other hand, consists of announcements of newly available products; we have not seen the products listed in Marketalk News.*

The Ol' Reset Blues
I've had my Apple for about a year and have had my frustrations with accidentally hitting the reset button. I removed the button for a time but didn't like the appearance of the computer, so I made a simple guard out of a three-by-five file card. The dimensions of the rectangular box are 20mm by 20mm by 30mm. I cut out a piece 30mm by 80mm, folded it every 20mm, and taped it so that a rectangular tube was formed. I then taped a 20mm by 20mm square on top. I punched a hole with a paper punch in the square before taping it to the tube, so that a pen might be inserted to activate the reset when needed. After it was assembled, I slid it down over the reset key. I haven't had any problem since.

Another idea concerns making nice print statements. I take a piece of graph paper and along the top of it count out forty spaces between quotation marks. Then I write what I want printed, including commas, etc., so words aren't chopped up . This way I can program hyphens in also, thus making everything neat and pretty.
Larry Blake, Springville, UT

Innate Hierarchy of Values
There is a handy feature in Applesoft that many people aren't aware of. It isn't published in any of the manuals. When I type:

```
PRINT"A">"B"
```

my Apple returns

```
1
```

But when I type:

```
PRINT"B">"A"
```

my Apple returns

```
0
```

With a little more experimenting, I found that the statement is true when a higher ASCII value is on the greater side of the equation and false when on the lesser side of the equation. You can also compare strings that are two characters long and longer.

With a little experimentation, you can use string comparisons to alphabetize strings. I am very surprised that a useful feature like this wasn't in any of the manuals.
David Husch, Saint Louis, MO

A Sales Apple in the Field
I was excited by your interview with Dick Clinchy and his creation, *LICMS*. I also want uses for Apple that involve client files on disk. I would also be interested in the creative uses of Apple in direct selling—insurance or other intangible products. Bringing Apple into the field for sales interviews at the client's office or home is my next project.
Chris Greaves, Brooklin, Ontario, Canada

Apple's Alter Ego
I enjoyed Greg Tibbetts's first column. It's about time some Apple publication started carrying SoftCard information on a regular basis.

My question is this: Although the SoftCard documentation describes how to switch back to 6502 mode temporarily while under CP/M, I would like to be able to do the reverse; that is, while under Apple DOS, I'd like to switch temporarily to Z-80 mode for execution of particular subroutines that could take advantage of some of the more powerful Z-80 instructions. How can this be done? I would even be willing to modify the SoftCard to achieve this.
Bill Krantz, North Wales, PA

By Any Other Name . . . .
I need all the help I can get. With my name, the Apple II was a must—please send Softalk as soon as possible!
Arnold A. Appel, Sabina, OH

THE PASCAL PATH

By Jim Merritt

Tools of the Craft, Part 4: Control Flow and Decision

Racing in Circles FOR a WHILE Longer. In last month's discussion of looping, we got as far as producing a simple program that uses one of Pascal's *looping statements*—specifically, the REPEAT statement—to count and display each of the 500 miles in the Indianapolis 500 auto race:

```
PROGRAM
 Indy500a;

 CONST
  EndOfRace= 500;

 VAR
  Distance
   :Integer;
BEGIN
 Distance := 0;
 REPEAT
  Distance := Distance + 1;
  WriteLn(Distance);
 UNTIL Distance = EndOfRace;
END.
```

Hopefully, you were able to compile and execute Indy500a with no difficulty, and you also had time to experiment with the program by recompiling it for different values of the constant EndOfRace. Playing with such small changes gives you more practice with the system and builds your confidence that you can, indeed, be successful in modifying programs for yourself with no explicit direction or help.

This month, we'll finish with our first examination of loops, by rewriting the Indy500 program to use the two other types of looping statements offered by Pascal: WHILE and FOR.

WHILE . . . DO . . . In modifying the program to use a WHILE loop, we're faced with one major problem right from the start: as defined in the syntax diagrams last issue, the body of a WHILE loop can contain only one statement, and you can be reasonably sure that the body of the Indy500 loop must contain at least two, one to display the current value of Distance, and one to increment it. The syntax definition of the REPEAT-UNTIL loop permits several statements to fall between REPEAT and UNTIL, but only one statement may follow the DO keyword to form the WHILE-loop body.

Compound Statements—The deeper purpose of BEGIN and END. It is possible to group the several necessary statements together, so that the Pascal compiler will view them as a unit. You can do this by bounding those statements between the keywords BEGIN and END, to form a *compound statement* (see figure 1). As long as it is expressed as a compound statement, a loop body may contain an arbitrarily large number of statements (even other compounds!). Remember, if there is more than one constituent statement in a compound, each statement must be separated from any following one by a semicolon. By the way, you may already have grasped that the statement part of a program is really just one example of a compound statement. You'll see others soon.

Here is the body of the Indy500a REPEAT-loop, written as a compound statement:

```
BEGIN
 Distance := Distance + 1;
 WriteLn(Distance);
END
```

Attach this compound to a WHILE-loop, using the same initialization statement and termination expression as in the REPEAT-UNTIL version, and you have a preliminary version of Indy500b:

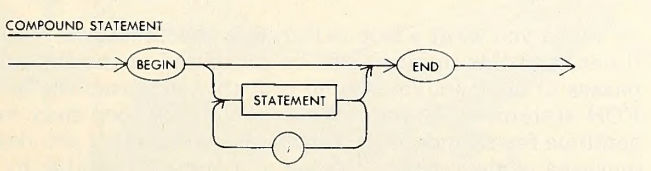


Figure 1.

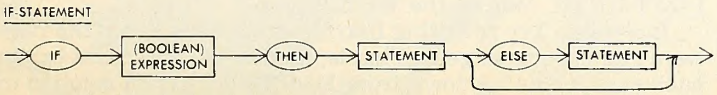


Figure 2.

```
PROGRAM
 Indy500b;

 CONST
  EndOfRace= 500;

 VAR
  Distance
   :Integer;
BEGIN
 Distance := 0;
 WHILE Distance = EndOfRace DO
  BEGIN
   Distance := Distance + 1;
   WriteLn(Distance);
 END;
END.
```

You shouldn't charge off to compile and execute a program until you have decided that it really has a chance of working. Remember, for a WHILE-loop, the termination condition is evaluated before the loop body is executed. If the termination condition holds upon entering the loop, the body will *never* be executed! Now, look at the initial value of Distance, and the termination condition. Distance is clearly not 500 when control arrives at the loop, so the loop is bypassed altogether.

Some changes must be made in Indy500b before it will yield the results we expect. The obvious place to start is the termination condition. There is a subtle difference in philosophy between the kind of terminating condition required for a REPEAT-UNTIL loop and that needed by a WHILE-loop. The value of the REPEAT-UNTIL condition should be False as long as the loop body is to be repeated. It expresses the situation that you feel should exist at the time the loop should cease. Conversely, the Boolean expression that is part of the WHILE-loop is more accurately characterized as a continuation condition, rather than a termination one. That is, it must be True as long as the loop body bears repeating, and False only when iteration must end. Thus, its meaning and function are the exact opposites of the corresponding expression in the REPEAT-UNTIL loop.

The logical opposite of "equal" (=) is "not equal" (<>), so why not try rewriting the WHILE-loop's Boolean expression as follows:

```
Distance <> EndOfRace
```

This resulting version of Indy500b is correct, and you should compile and execute it to see that it does, indeed, behave as expected.

```
PROGRAM
 Indy500b;

 CONST
  EndOfRace= 500;

 VAR
  Distance
   :Integer;
BEGIN
 Distance := 0;
 WHILE Distance <> EndOfRace DO
  BEGIN
   Distance := Distance + 1;
   WriteLn(Distance);
  END;
END.
```

FOR . . . TO/DOWNTO . . . DO . . . The FOR loop is a special case of the WHILE loop that is optimized for counting problems. Refer to the syntax diagram. After the keyword FOR, you name a control variable, which can be of any fundamental type except Real, or any programmer-defined enumerated or subrange type. The control variable for a FOR, just like any other variable, must be declared prior to use. As you can see, the control variable is actually introduced as part of an assignment, which is embedded within the FOR construction. The assignment specifies the control variable's initial value, which can be any expression whose type is compatible with that of the variable.

Following the assignment is the keyword TO or the keyword DOWNTO. If TO is used, the FOR-loop will count up, incrementing the control variable to the next greater value for each iteration. If DOWNTO is used, the loop will count down. Next, the ultimate value of the control variable is specified as an expression. Then comes the keyword DO, and finally, the body of the loop. As with WHILE, the body of a FOR-loop may contain only a single statement, but this is no restriction, since that single statement may be a compound.

The initial value of the FOR-loop control variable should be the first value you expect to use. In our earlier versions of Indy500, the initial value was 0, but this value was never actually used for anything by the body of the loop. The first useful value was, of course, one.

Because initialization is built into the FOR-statement, you don't need to put a separate initialization assignment ahead of the loop. You also don't need an assignment statement within the loop body that increments or decrements the value contained in the control variable, because the computer takes care of this automatically. Finally, the termination condition is nothing more than an expression that evaluates to the greatest (or least) value that the control variable should assume.

The FOR-loop turns out to be ideal for the Indy500 problem. Here is the complete listing for this version of the program:

```
PROGRAM
 Indy500c;

 CONST
  EndOfRace= 500;

 VAR
  Distance
   :Integer;
BEGIN
 FOR Distance := 1 TO EndOfRace DO
  WriteLn(Distance);
END.
```

Notice that the body of the FOR-loop didn't have to be a compound, because we were able to dispense with the superfluous incrementing of the Distance variable. For purposes of comparison, the FOR-loop in Indy500c is equivalent to the following WHILE-loop:

```
Distance := 1;
 WHILE (Distance <= EndOfRace) DO

 BEGIN
  WriteLn(Distance);
  Distance := Distance + 1;
 END;
```

Exercise: Compare the above with Indy500b, noting the differences in the initialization assignment statements, termination conditions, and bodies.

When you want a loop to iterate a specific, finite number of times, and this number can be computed at the time control passes to the loop, you should probably construct the loop as a FOR statement. If you anticipate that the loop may need to continue for an indefinite number of cycles, or if you desire to increase or decrease the value of a control variable by more than one unit for each iteration, you should use either the REPEAT-UNTIL loop or the WHILE-loop.

Exercise: Try rewriting Indy500 (a, b, and c) as if the race were run in reverse gear. That is, modify the programs to run backward, counting down from EndOfRace to 1. A solution to this exercise appears at the end of the column. Hint: You will

Communications Software from the Utility Specialists

What do you really need in good communications software? We think the most important thing is using the equipment YOU have to transfer data easily and accurately. When Ascii Express made its first appearance in 1979, it was immediately recognized as the finest program of its kind available. Since that time, other programs have appeared with one or two flashy new features, but not a single one of them offers as many choices in terms of hardware compatibility and data transfer modes as the SDS line of communications software.

ASCII EXPRESS II™

If a versatile and accurate data transfer program is what you're after, Ascii Express II is for you. It offers more data transfer modes than any other program of its kind. In fact, we have yet to find a computer system it can't be used with.

In addition, Ascii Express II offers a built-in editor that allows you to produce or modify files before and after transfer, a built-in directory for your most frequently called numbers, and keyboard macros which allow you to send entire sign-on sequences or other host commands with a few simple keystrokes. All these features are designed to help reduce on-line time, and that saves you money.

Z-TERM™

For CP/M users, Z-Term offers all the above, and more . . .

- Support of all 80-column boards and external terminals
- Large copy buffer (41K)
- Unlimited download capacity – no file is too large

For special applications, Z-Term will even allow your Apple to emulate most popular types of terminals for maximum compatibility with your system.

Z-TERM "THE PROFESSIONAL"™

The ultimate communications package. This is the finest program we offer, and it is certain to set the standard for all communications software for years to come.

Z-Term "Pro" provides the solid performance that the truly serious user demands. Many of our users find Z-Term "Pro" to be so valuable that it more than justifies the purchase of the Z-80 Softcard.

Z-Term Pro's special features include:

- Support of the new Novation Apple CAT modem
- Support of specialized transfer protocols, e.g. PAN and Christensen
- Automatic answer and send/receive modes for unattended operation
- Special buffer for parallel printers to ensure no lost characters

We are so convinced that all the SDS programs are the finest available that we've compiled a detailed point-by-point comparison of our programs and those of our leading competitors (for your copy, send a stamped, self-addressed envelope to us at the address below). This comparison of more than 50 items clearly shows what our customers have known all along – that SDS software is the finest. Amen.

SDS
southwestern data systems

P.O. Box 582-S
Santee, CA 92071 Tel. 714-562-3670

CP/M is a trademark of Digital Research.
Z-80 Softcard is a trademark of Microsoft.

TRYING TO MAKE CONTACT ?

probably have to rearrange something to get a proper display.

Extra Credit Exercise: Rewrite Indy500a so that each iteration corresponds to a 2.5-mile *lap*. Display the current Distance in miles for every lap, and be sure that the racer doesn't travel any farther than 500 miles! There are several solutions possible for this problem. One is given at the end of this column. Try to resist the temptation to peek! (Why do you think we're not asking you to rewrite the FOR-loop version? Those with stout hearts might like to give it a try.)

Decision. A simple count-up (or count-down) display is fun to watch once or twice, especially when the program producing it is one you have entered and compiled yourself. However, the novelty will soon fade, and you'll want to move on to fresher challenges. Here's one that we'll tackle together: modifying the Indy500c (FOR-loop) program to note when each *quarter* of the race is complete.

First, let's be more precise in specifying our task. Of course, we'll produce a program that counts up, displaying the miles driven just as does Indy500c. In addition, though, we'll arrange for the program to display special markers at each quarter of the race, as soon as the last mile in the quarter has been driven. The markers will look like this:

*** Race is q/4 finished.

where q is one of the numbers 1, 2, 3, or 4.

Indy500c already embodies the counting-up portion of our proposed new program, so we'll use it as a starting point. How do we get Pascal to display the markers at the appropriate times? It's safe to assume that any display will involve a WriteLn. So, we will have to add to the program a WriteLn statement that causes the specified marker message to be displayed. This is relatively easy (and we'll make light work of it soon). Our real problem is to make sure that the WriteLn is executed *only* at certain times (namely, at the quarter-points). I'll show you how to do this by incorporating the WriteLn into a Pascal IF-statement.

If You're Interested . . . A syntax diagram for the IF-statement is shown in figure 2. The IF has two or three parts: the IF-clause (which includes a Boolean expression, called the *IF-condition*), the THEN-clause (which includes a single or compound statement that will be executed should the IF-condition be True), and, optionally, an ELSE-clause (which, like the THEN-clause, includes a single or compound statement but will be executed only if the IF-condition is False). Assuming StatementA, StatementB, and BooleanExpression are the kind of objects their names imply, here are two examples of well-formed IF-statements:

```
If BooleanExpression
 THEN
  StatementA
```

```
If BooleanExpression
 THEN
  StatementA
 ELSE
  StatementB
```

In the first example, either StatementA will be executed or it won't. In the second, either StatementA or StatementB will be executed *but not both.* The first example, which has no ELSE-clause, is equivalent to the following, which uses a redundant ELSE:

```
If BooleanExpression
 THEN
  StatementA
 ELSE ;
```

(The semicolon after the keyword ELSE is to accentuate the fact that a *null statement* follows ELSE here.)

You should note that semicolons *cannot* be used to separate the clauses in an IF-statement. Here is an example that illustrates a common mistake involving improper semicolon usage:

```
IF BooleanExpression
 THEN
  StatementA;
 ELSE
  StatementB
```

When scanning this IF-statement, the compiler will believe that it has reached the statement's end at the semicolon that follows StatementA. Pressing on, the compiler will expect to find another statement, but will instead see the keyword ELSE, which cannot introduce a statement. At this point, the compiler will complain of an "Error 6" (Illegal Symbol). (Interestingly enough, the summary of compiler error numbers in Apple's *Pascal Language Reference Manual* suggests that error 6 is caused most often by the lack of a semicolon on a preceding line in the program—here is a case where the error occurs because of the *presence* of an unnecessary semicolon!)

For our new Indy500 program, should we use IF-THEN, or IF-THEN-ELSE? With every iteration of the Indy500 FOR-loop, either a quarter-point has been reached or it hasn't. We either want to display the marker message or we don't. When there are two, clear, mutually exclusive alternatives to choose from, use IF-THEN-ELSE. But here, our choice is between executing the WriteLn or not executing it. We can dispense with the ELSE clause altogether, and simply use IF-THEN.

For the conditional execution of the WriteLn statement to occur at the right time, we must find a Boolean expression that is true only when a quarter-point has been reached. The only changing object in our program—and thus the only thing we can use to determine when a quarter-point occurs—is the FOR-loop's control variable, Distance. Distance is an integer variable; its value at any instant is some integer number. A little figuring shows that the values Distance will contain at the quarter points are 125, 250, 375, and 500, respectively. You may be tempted, therefore, to use the following condition in the IF-clause:

```
((Distance = 125) OR (Distance = 250)
OR (Distance = 375) OR (Distance = 500))
```

This will certainly work but only for 500-mile races and, then, only if you want to flag just the quarter-points. What if you're interested in only the half-way-point and finish line, or what if you want to see a marker for every fifth or every eighth of the total distance? *Now,* before you write that IF-condition, is the time for you to plan for the future of your program. To use the same program for races other than the Indianapolis 500, you may need to change one—or both—of two things: the length of the race, and the relative interval between markers. The race length is easily changed by altering the constant EndOfRace. Perhaps we should also introduce a constant that tells how many equal subdistances should be flagged by a marker message. Let's call this constant Partitions, and set its value at four for the time being.

At this point, we know the length of the race and we know how many partitions need flagging. It turns out that we can use these two numbers to form an IF-condition that is True when a partition-point has been reached during the race, but False at all other times. To do this, we make use of several facts. First, the length of a partition is equal to the Integer expression (EndOfRace DIV Partitions). For example, when EndOfRace is 500 and Partitions is four, the length of a single partition is (500 DIV 4), or 125.

Next, one number divides another evenly if the division produces no remainder. Remember that the remainder of (A DIV B) is given by the expression (A MOD B). So, B divides A evenly if ((A MOD B) = 0).

Finally, every distance that corresponds to a partition-point must be evenly divisible by the length of a partition. You can see that, in our specific example, each of the four partition-points, 125, 250, 375, and 500, is evenly divisible by the length of a partition, 125.

Using the above facts, we can create the Boolean expression we need:

```
( (Distance MOD (EndOfRace DIV Partitions) ) = 0)
```

Let's work backward to appreciate how this specific formula expresses what we mean. By successively substituting symbolic names for the subexpressions, you can see that the total expression is consistent with our three facts. Substituting LengthOfPartition for (EndOfRace DIV Partitions) gives

```
((Distance MOD LengthOfPartition) = 0)
```

Since the MOD expression here tells us how many miles the racer has driven beyond the last partition-point, we can substitute the name MilesPastLastPoint for (Distance MOD LengthOfPartition) to get

```
(MilesPastLastPartPoint = 0)
```

This expression is equivalent to saying

```
AtPartitionPoint
```

which is just what our IF statement needs!

New Tricks With an Old WriteLn. To complete the IF-statement, we must construct a WriteLn that will, when executed, produce the message we desire. Let me throw it out to you, and explain what's been done afterward:

```
WriteLn(
 '*** Race is ',
 (Distance DIV (EndOfRace DIV Partitions)),
 '/',
 Partitions,
 ' finished,'
 )
```

The WriteLn statement is spread across several lines, partly for typographical reasons but mostly to accentuate the fact that it specifies the display of five separate items. Up to now, we've only displayed one data item with every WriteLn statement. However, it's possible to specify a *list* of items, all of which are to be displayed next to each other on the same output line. Two of the items—the second and fourth—are integer expressions and correspond respectively to the numerator and denominator of the fraction that we wish to display in the marker message. The information displayed for each will be the actual value of the numeric expression, as evaluated at the time the WriteLn is executed. The third item ('/') is a character literal, and, of course, the apostrophes that delimit it will not be displayed.

A Word About Strings. Until now, you haven't seen the type of datum represented by the first and fifth display items in the WriteLn list—at least, not in this column. They are examples of *character strings*. A character string literal (see figure 3) is something like a Char literal, in that it is delimited by the apostrophe mark. The Char literal, however, consists of exactly one character, whereas the string literal may consist of any number of characters (or even no characters!). When a string literal is an item in a WriteLn list, the group of characters that lie between its apostrophe delimiters are displayed on the screen. Thus, we can use string literals to display words and phrases, where, before now, we could show only numeric values. We'll go into detail about character strings in a future column.

A word to the wise: Keep in mind that the digits (0 to 9) *are also characters* that can occur within string literals. However,

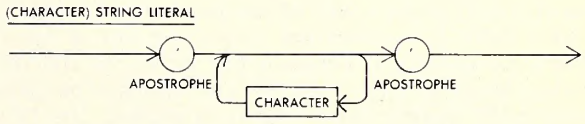


Figure 3.

there is a world of difference, for example, between the entirely incompatible values of the string literal 123 and the integer literal 123. In short, a string may look like a number, but don't be fooled. They're not the same.

Note that the first string literal (item 1 in the WriteLn list) ends with a blank and the second (item 5) begins with one. I did this to provide appropriate spacing between the literals and the material displayed between them. Remember, all the items in the WriteLn list are displayed, one after another, on the same line. Pascal doesn't put any spacing between them; therefore, you must do it yourself if you don't want two or more items to run together. To get an idea of why you must often insert these trailing, or leading, blanks, compile and execute this program:

```
PROGRAM
 RunTogether;
BEGIN
 WriteLn(1, 0, 1);
 WriteLn(1, ' ', 0, ' ', 1);
 WriteLn('Run', 'Together');
 WriteLn('Run ', 'Together');
END.
```

IF it Fits, THEN Stick it There! At last, we are finished with the IF-statement that controls the display of our marker message, and we're ready to insert it into the Indy500c framework. But where? Since we have used the variable Distance within the IF-statement, and Distance assumes reasonable values only within the FOR-loop body, it is clear that the IF-statement must also be a part of the loop body. To add it, however, we'll have to convert the loop body into a compound statement, by placing it between the keywords BEGIN and END. Also, since we stipulated that the message be displayed just after each quarter-point has been passed, the IF-statement should be inserted into the program text *following* the original Indy500c WriteLn.

When you have finished editing this new version of Indy500, it should look like this:

```
PROGRAM
 Indy500d;

 CONST
  EndOfRace= 500;
  Partitions= 4;

 VAR
  Distance
   :Integer;
BEGIN
 FOR Distance := 1 TO EndOfRace DO
  BEGIN
   WriteLn(Distance);
   IF (Distance MOD (EndOfRace DIV Partitions) = 0)
    THEN
     WriteLn('*** Race is ',
      (Distance DIV (EndOfRace DIV Partitions)),
      '/',
      Partitions,
      ' finished.'
      );
  END;
END.
```

Now, compile the program and execute it. You might also try recompiling the program for different values of Partitions and EndOfRace. Be sure, though, that EndOfRace is evenly divisible by Partitions. (What do you think will happen if it's not? If you have an answer, go ahead and test your theory.)

What ELSE Can You Do? Perhaps, now, you'd like to try using an ELSE clause. I have just the thing for you. Personally, I think that it's rather strange for the program to report that the race that has just ended is "4/4 finished." The actual end of the race is a special case, and so I'd rather see the program say "*** Race is over" at that point. In other words, I think it would be nice for the program to issue one kind of marker message that reports fractional progress for all the partition-points except the last but issue a different message when that final point has been reached.

This is an IF-THEN-ELSE type of situation, because, at any partition-point, the program must choose whether to display the fractional marker or the end of race marker. Notice, however, that this choice applies *at a partition-point.* In fact, introducing this special case means that the program has two levels of decision to make. First, for every mile driven, it must determine if a position-point has been reached so as to decide whether to issue a marker at all. Once it's clear that a marker is needed, the program must then decide which of the two mutually exclusive messages it should display. Let me show you how you might write this split-level decision in Pascal:

```
IF (Distance MOD (EndOfRace DIV Partitions) = 0)
 THEN
  IF (Distance <> EndOfRace)
   THEN
    WriteLn('*** Race is ',
     (Distance DIV (EndOfRace DIV Parititions)),
     '/',
     Partitions,
     ' finished.'
     )
  ELSE
   WriteLn('*** Race is over.');
```

The above is an example of a "nested IF-statement." That is, the statement controlled by an IF is itself an IF. The first IF-condition is the one that we have already spent so much time developing—it determines when a marker should be displayed. The second, nested IF-condition is true for all parti-

tion-points but the final one because, at the final one, Distance is exactly equal to EndOfRace. Whenever the Distance is not equal to EndOfRace, the WriteLn under THEN will be executed. When Distance becomes EndOfRace, the WriteLn under ELSE is executed instead. Note that, in accordance with Pascal syntax, there is no semicolon following the WriteLn that precedes ELSE. Try substituting the nested IF-statement for the one we used originally.

Coming Attractions. Next time, our overview of programmed decision making will conclude with a look at Pascal's CASE-statement. Also, I'll introduce you to Write, WriteLn's fraternal twin, and talk about the pair's hitherto unmentioned output formatting capabilities. Finally, you'll learn to inject special commentary into your programs to make them more readable. See you in November! ◨

Answers to Exercises

Exercise:

```
PROGRAM
 Indy500a;
 (* declaration area is same *)
BEGIN
 Distance := EndOfRace;
 REPEAT
  WriteLn(Distance);
  Distance := Distance - 1;
 UNTIL (Distance < 1);
END.
```

```
PROGRAM
 Indy500b;
 (* declaration area doesn't change *)
BEGIN
 Distance := EndOfRace;
 WHILE Distance > 0 DO
  BEGIN
   WriteLn(Distance);
   Distance := Distance - 1;
  END;
END.
```

```
PROGRAM
 Indy500c;
 (* declaration area doesn't change *)
BEGIN
 FOR Distance := EndOfRace DOWNTO 1 DO
  WriteLn(Distance);
END.
```

Extra Credit Exercise

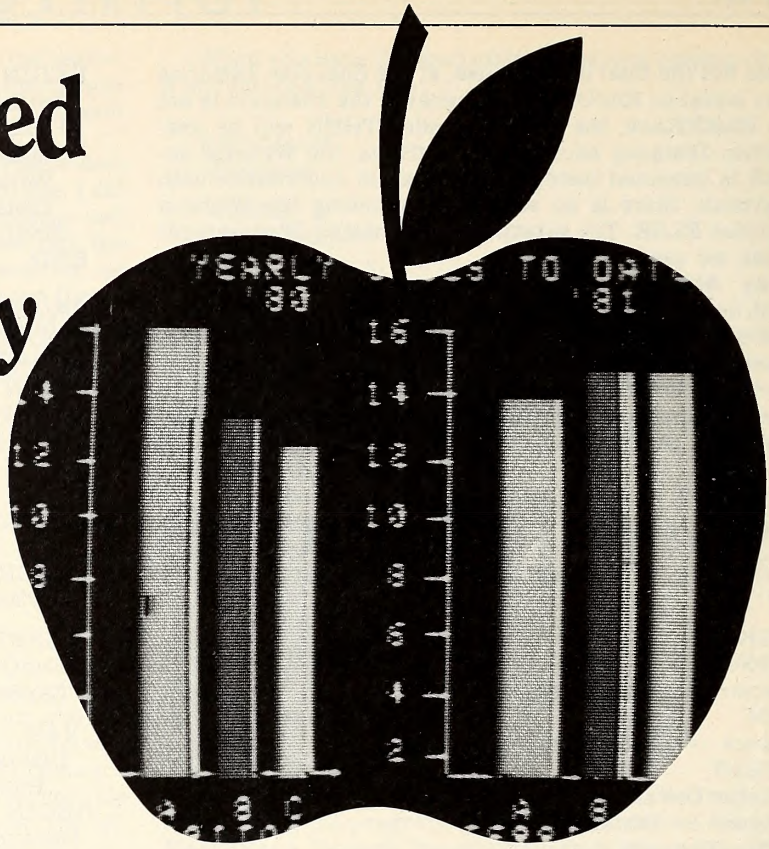
```
PROGRAM
 Indy500a;

 CONST
  EndOfRace= 500;
  LapSize= 2.5;

 VAR
  Distance
   :Integer;
BEGIN
 Distance := 0;
 REPEAT
  Distance := Distance + 1;
  WriteLn(Distance*LapSize);
 UNTIL ( (Distance*LapSize) >= EndOfRace);
END.
```

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.**

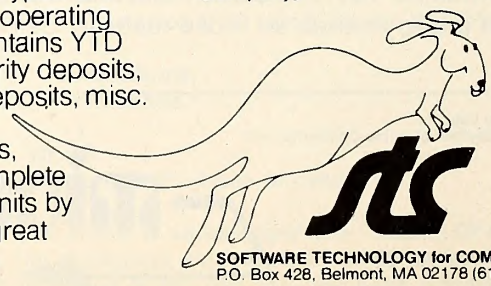
Early Bird Offer

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.

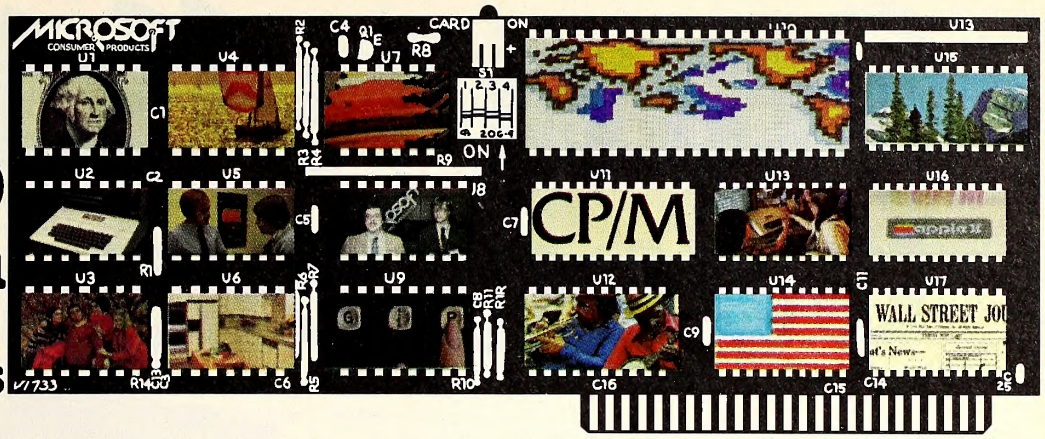
(Apple is a trademark of Apple Computer Company)



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

SOFTCARD Symposium

by Greg Tibbetts



Last month, we discussed the features of Microsoft Basic, release five, and the differences between MBasic (or GBasic) and the two Apple Basics. We then discussed the new Basic-80 commands—not available in either Integer or Applesoft—and identified all the direct commands and a few of the simpler indirect commands. These are the commands left to cover:

CHAIN	LSET/RSET
COMMON	OPEN
DEFINT/SNG/DBL/STR	OPTION BASE
ELSE	RANDOMIZE
ERASE	WHILE/WEND
FIELD	WIDTH
LINE INPUT/#	

The file handling commands will be part of a future discussion of file handling in general. Let's now examine commands for the handling of variables.

Basic-80 can deal with four types of variables: integers, which, like Apple Basics, are restricted to whole numbers in the range of −32768 to +32767; single precision reals, which are numbers that may or may not have a decimal point that are significant to six digits; double precision reals, which are numbers with or without a decimal point that are significant to sixteen digits; and, finally, strings that represent not numbers at all but collections of zero or more ASCII characters (a string with zero characters is called a *null string*). The many different numeric variable types provide the user with maximum flexibility. True integer variable types and integer arithmetic like that found in Integer Basic execute much faster and take less than half the storage space of floating point variable types and arithmetic like that in Applesoft. (Note: Applesoft doesn't use true integer storage or manipulation even when variables have been explicitly declared as integers by the % sign having been appended to the variable name.) Floating point variables are required, however, to represent numbers outside the range of integer representation or arithmetic operations, and the penalties of space and speed are accepted as part of the cost. Basic-80, in providing for true integers, four-byte single precision reals, and eight-byte double precision reals, gives you a reasonable choice between precision and space and speed considerations.

Basic-80's commands for declaration of variable types consist of immediate, explicit declarations similar to those in the Apple Basics (% for integers, ! for single precision, # for double precision, and $ for strings), plus the ability to define whole classes of variables (all those beginning with a named letter), as a type. The variable class definition capability is provided by the DEFxxx statements; DEFINT for integers, DEFSNG for single precision, DEFDBL for double precision, and DEFSTR for string variables. Thus, a single statement at the beginning of the program of the form DEFINT A will make all variables in the program beginning with the letter *A* default to integers. Explicitly declared variables (for example, A#), are not affected by this; therefore, they'll be treated as whatever type is indicated by their appended signs. If no declaration is made, Basic-80 defaults all variables to single-precision reals.

One final point about DEFxxx statements: they set up separate variable tables for each variable type when the statement is encountered, and, while issuing multiple DEFxxx statements for the same variables is allowed and will have predictable results, doing this can generate real confusion if the effects are not anticipated. Here's a program segment that illustrates this:

```
10 Defint A:A=1000:Print A
20 Defdbl A:A=1.0000000001:Print A
30 Defint A:Print A
```

generates the following output when run

```
1000
1.0000000001
1000
```

It's easy to see that the declaration and assignment of A made by line 10 were still intact even after line 20 was executed; however, because of line 20, Basic-80 could not find the earlier value until the second DEFINT A was executed in line 30. This feature can be useful but should be used with care.

The final variable-related command is ERASE; use it whenever you want to eliminate storage space taken up in memory by an array, either to redefine a new array with the same variable name or simply to clear more memory for other purposes. The command's format is ERASE followed by a list of array variables. ERASE alone (with no variable names following it) generates a syntax error. Once an ERASE statement is encountered, all values associated with the array are unrecoverable, and attempts to access elements up to ARRAY(10)—the default dimension for new arrays—will return 0 as the value; attempts to access higher elements result in a subscript-out-of-range error.

Let's now deal with commands that pass control between two programs: the command pair CHAIN and COMMON. In the Apple Basics, the limited ability to pass control to other programs on disk is handled exclusively by Apple DOS through the Basic RUN and PRINT commands combined with the DOS prefix character control-D in the form PRINT "<control-D>RUN progname." In both Applesoft and Integer, this sequence runs the named program and, in the process, erases the existing program and its variables. In Integer Basic, the DOS command CHAIN (used in the same print format) runs the named program and preserves the variables. In Applesoft, saving the variables requires Bloading a machine language program named CHAIN and then calling it with the new program name appended.

Basic-80, on the other hand, is fully capable of chaining one program to another while preserving the variables in the existing program. Overlaying is also allowed, but here it's a somewhat more primitive procedure than some mainframe users may be familiar with. It's done with the MERGE option; we'll discuss this in detail later.

CHAIN, in its least complex form, is simply CHAIN

Start talking business

with your Apple

COBOL is the most effective business language. Apple II is the most friendly business computer.

CIS COBOL with FORMS-2 brings together the best features of COBOL and Apple to enable you to deliver the most effective, user-friendly applications.

Business Programmers: Take the COBOL expertise you have acquired on big business mainframes, and use it on Apple II to create friendly applications that will talk directly to your users – where it suits them best, on their own desks.

CIS COBOL's dynamic module loading gives you big application capability and the FORMS-2 source generator lets you build and modify conversational programs from visual screen formats, creating much of the code automatically.

Application vendors: CIS COBOL with FORMS-2 steps up the pace for your development of the high quality professional application packages needed today. And creating them in COBOL makes them more maintainable.

Over half the Apple II's now being sold are going to business or professional users so demand for quality applications is growing fast, creating big business opportunities for you.

Stability proven by the US Government. CIS COBOL has been tested and approved for two consecutive years by the US General Services Administration as conforming to the ANSI '74 COBOL Standard. Apple II under CP/M is included in CIS COBOL's 1981 GSA Certificate of Validation (at Low-Intermediate Federal Standard plus Indexed I-O and Level 2 Inter-Program Communication).

Get your hands on CIS COBOL at your Apple dealer.

Talk business with him now!

Micro Focus Inc., 1601 Civic Center Drive
Santa Clara, CA 95050. Phone: (408) 248-3982.

MICRO FOCUS

CIS COBOL with FORMS-2 for use on the Apple II with CP/M is an Apple Distributed Product.
CIS COBOL and FORMS-2 are trademarks of Micro Focus. CP/M is a trademark of Digital Research. Apple II is a trademark of Apple Computer.

"d:progname", where progname is the called program on disk drive d:; default is the currently logged drive. If no other options are specified, this functions exactly like PRINT "<control-D>RUN progname" in either Apple Basic. If you want the execution of the new program to begin at a specific line number, the number preceded by a comma must be added (for example, CHAIN "progname",1000). Using a variable for the line number is allowed, and this allows conditional branching to your choice of any number of entry points in the called program.

Variables may be passed straight to the called program either by using the ALL option to CHAIN or by using the COMMON statement in the calling program. If you want to pass all variable values to the called program, simply append the word ALL preceded by a comma to the CHAIN command string (that is, CHAIN "progname",ALL). If, for space conservation or some other reason you want only to pass some variables, the calling program must inform Basic-80 that some variables are to be preserved in subsequent CHAINs and others may be overwritten. Do this with one or more COMMON statements, formatted as COMMON followed by a list of variables to be preserved. Variable names must be separated by commas. Include entire arrays by appending () to the array name—for example, ARAYVAR ().

When you use conditional branching to selected programs, you can pass any number of common variable sets simply by designing the calling program to execute only one of several COMMON statements (called COMMON declarations). Take care, however, since Basic-80 won't allow a variable to be declared COMMON in more than one statement in each program. Accomplish this by declaring COMMON all variables shared by all the program modules at the very beginning of the program. Then have conditional program segments declare those shared by only a few. Once the COMMON declarations have been made in the calling program, just use the CHAIN statement in its regular form to preserve these variables.

Be aware that variable type declarations—DEFINT, DEFSNG, DEFDBL, and DEFSTR—as well as user defined functions such as DEFFN—are not passed to the called program. This means that these statements must be restated in the called program if variables declared or functions defined by them are to be used. If, for example, the statement

```
10 Defint A:A=10:Common A
```

were declared in the calling program, the called program wouldn't be able to access variable A—it would treat it as 0, or undefined—unless it also contained the statement DEFINT A. So remember that whenever you use variable type declarations and user function definitions with chained programs, you must always duplicate such statements in all program modules.

Basic-80's overlay capability is brought into play through the MERGE option. The term *overlay,* in this case, means that a program or program segments is brought in from disk and overlays part of the existing program in memory. In this way, it's similar to the MERGE direct command and, in fact, some of the same rules apply. Program files on disk to be brought in must have been saved in ASCII format. The program, when it's brought in, doesn't erase the entire calling program but merges with it, eliminating only those lines of the calling program that have identical line numbers to those in the program coming in.

Unlike the direct command, however, the MERGE option performs its task and continues in the normal function of the CHAIN command, beginning execution of the composite program now in memory. All the normal CHAIN functions are still valid. For example, if no starting line number is specified, execution begins with the lowest numbered line of the composite program. This, incidentally, usually starts the calling program over again only in its modified form. Typically, overlays are used in place of several chained stand-alone pro-

grams when the bulk of the programs are identical and only a few routines are different. In these cases, it's desirable to structure the main program so it has a block of line numbers to use as an overlay area for each function that has multiple choices. So, while the program proceeds, you set up the selected function (program segment to be overlaid), use CHAIN MERGE to bring it in, and execute it. Later, other overlays may be brought in to the overlay area as needed for execution. An example is a payroll program that calculates state income tax when applicable. The program would ask for the payee's state of residence and, based on the answer, would select the correct overlay program name to be called in later, assigning it to a string variable. The program would then GOSUB to a general purpose CHAIN MERGE statement that would bring the appropriate overlay into the proper area and execute it. In this example, the overlay need end only with a RETURN to continue execution of the main program. If the overlay area were the block of line numbers from 10000 to 10500, the statement would look like this:

```
8000 CHAIN MERGE "A$",10000,ALL
```

If the overlay area will be used more than once, it's dangerous just to bring in new overlays and execute them because lines not specifically replaced by the new overlay remain and are executed. Delete old overlays prior to merging in new ones by adding the DELETE option to the CHAIN command string. In this case, it would become:

```
8000 CHAIN MERGE "A$",10000,ALL,DELETE 10000-10500
```

This much safer form is preferred. Note that we used the ALL option, the usual method of preserving variables when overlaying. The main program is still intact, and it's assumed that all current variables will still be needed. Notice that since we aren't replacing one program with another in memory, variable declarations of the form DEFxxx remain in effect even for the overlay, so there's no need to repeat them whenever the MERGE option is used with CHAIN. As you might suspect, the same is true for user-defined functions, such as DEFFN.

Let's now examine commands for conditional branching of program control—ELSE and WHILE/WEND. The ELSE command is an additional option used with the IF . . . THEN command pair, which lets you set up a clear either/or decision point. Using Apple Basics, if you wish to have the program perform one function if the IF conditional were true and some other function if it were false, you'll need to create another line immediately below the conditional directing Basic to perform the alternate function. Using IF . . . THEN . . . ELSE in Basic-80 it's possible to combine everything in a statement like this:

```
10 If A=B Then 140
20 GOTO 240
```

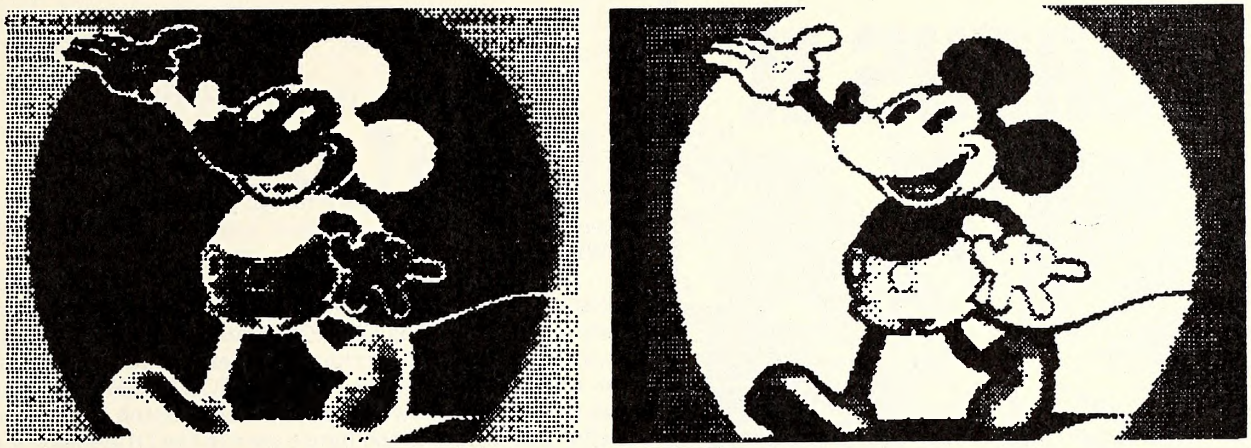
```
10 If A=B Then 140 Else 240
```

Because of the way Basic-80 is organized, it's also possible to include several tasks to be performed at each decision point. Basic-80 ignores the remainder of a line after a false conditional until an ELSE statement occurs. Then it'll ignore the ELSE statement and all that follows if the conditional is true. For example, if we wanted to see if A were equal to B and, if so, make A equal to C, make B equal to C, and jump to line 1000 but, if not, then make A equal to B, C equal to B, jump to line 2000, we would do it like this:

```
10 If A=B Then A=C:B=C:Goto 1000 Else A=B:C=B:Goto 2000
```

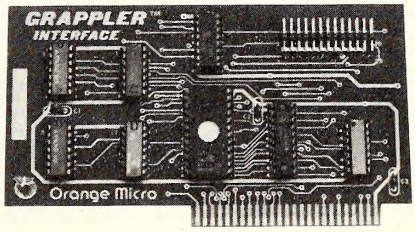
As long as the ELSE statement is on the same logical line (no other line number occurs before it), the example can be extended up to the allowed length of a Basic line.

GRAPHICS + APPLE = GRAPPLER™



ACTUAL APPLE II SCREEN DUMP USING GRAPPLER AND ANADEX 9501

THE GRAPPLER™ Apple Interface and Cable



A new printer interface that makes graphics easy!

The Grappler™ Interface is the first universal parallel interface card to provide sophisticated on-board firmware for Apple high resolution graphics. No longer does the user need to load clumsy software routines to dump screen graphics — it's all in the 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.

*Requires Graftrax-80

FEATURES	USER BENEFIT
SELECT PAGE	— Choice of Hi-Res Graphics page 1 or page 2
INVERSE GRAPHICS	— Provides reverse graphics of black-on-white, or white-on-black
DOUBLE SIZE PICTURE	— Doubles the graphic screen representation vertically & horizontally
90° ROTATION	— Rotates the screen picture 90°
CENTER GRAPHICS	— Accomplished through setting left margin thereby centering the graph
CHART RECORDER MODE	— Successive horizontal pictures are combined continuously simulating a chart recorder
BLOCK GRAPHICS	— For printers with block graphics (e.g. Epson MX80, Okidata M80), the high order bit can be controlled
BELL	— For printers with a bell, bell characters are deleted during user program listings
MARGINS	— Set left and right margins
SKIP-OVER-PERF.	— Set page length; printer will automatically skip 6 lines between each page
VARIABLE LINE LENGTH	— For user program listings, sets line length and wraps around with breakpoint at nearest blank
TEXT SCREEN DUMP	— The text from a user report or page of program listing can be dumped directly from the screen

TOLL FREE **(800) 854-8275** DEALER INQUIRIES WELCOME

CA, AK, HI **(714) 630-3322**

Orange Micro inc.

3150 E. La Palma, Suite G, Anaheim, CA 92806

WHILE and WEND initiate a program loop sequence similar to the FOR . . . NEXT pair. The difference is that instead of explicitly stating the maximum number of loop iterations to be performed, WHILE . . . WEND lets you continue executing the loop while some condition is true and exit when it becomes false. Say you wanted continuous input from the keyboard until the word *end* is entered. Using a FOR . . . NEXT sequence would require an IF . . . THEN conditional test inside the loop, so when *end* appears, the program would branch out of the loop. With WHILE . . . WEND, no test is required. Examine this:

```
10 For I=1 To 10000
20 Input A$
30 If A$="END" Then
 I=10001
40 Next I
```

```
10 While A$<>"END"
20 Input A$
30 Wend
```

Obviously, the more serious the purpose, the greater potential savings of time and program space provided by WHILE . . . WEND.

We'll look now at LINE INPUT; OPTION BASE; RANDOMIZE; and WIDTH, the miscellaneous commands. The simplest of these is OPTION BASE, which sets the minimum value for array subscripts to either 0 or 1. The default value is 0, and when used in the form OPTION BASE 1, the smallest array subscript allowed is 1. This helps match the number of possible subscripts to the figure used in the DIMension statement. In OPTION BASE 0, DIM A(10) sets aside A(0) through A(9), but after execution of OPTION BASE 1, that statement sets aside A(1) through A(10).

The RANDOMIZE command reseeds the random number generator. The Apple Basics use a memory location that's continually increasing, overflowing, and starting again as the random number seed. In contrast, Basic-80 requires that you, the programmer, supply the seed value from which the random numbers are computed. The seed value must be in the range of −32768 to +32767 and, given the same seed on successive runs, it will generate the same values. At first, the idea of generating identical sequences of random numbers may seem strange, but, actually, it could be no other way. The Apple can't generate a truly random number because all algorithms are just computations performed the same way with the same results each time. An algorithm used by Basic-80, however, is sound. As long as a different seed is selected for each run of the program, the numbers computed will be random enough for most purposes. To avoid supplying the seed each time, just input the current date and time; any number of simple calculations will generate a unique seed number.

The WIDTH command resets the width and height of the default screen. If LPRINT is inserted immediately following the WIDTH, it'll reset printer width, too. Be aware that if you use an eighty-column board or external terminal, WIDTH may cause some unpredictable effects depending on your computer's configuration or firmware. Printers and their firmware or software driver routines may also play havoc with your WIDTH command.

The last miscellaneous command is LINE INPUT, which is very similar to the standard INPUT command to a string variable with some minor differences. First, no question mark is printed to signify that input is requested. Second, all characters except carriage returns are included. In contrast, if the INPUT command encounters a quotation mark as the first character, it will read input only to the next occurrence of a quotation mark. The commands also differ in the combination of punctuation marks within each. The best way to determine the effects of these differences is to experiment with the two commands for a few minutes.

Next month, we'll explore the file handling commands and other commands that resemble their counterparts in Apple Basics, but are used differently by Basic-80. We'll also cover their extra functions as well as the conversion process required to get non-Basic-80 Basic programs to run properly. ■

It's a Jungle Out There!

THE BOOK
WILL CLEAR THE WAY THROUGH THE SOFTWARE TANGLE

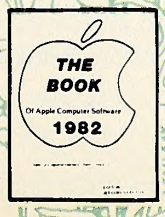
Don't settle for manufacturers' promotional material.

THE BOOK is a complete **critical analysis** of most Apple software available. Experts **review, rate** and **evaluate** the programs on 11 separate points. It covers Games, Education, Business and Utilities as well as hardware.

Don't buy software until you read this book. With the overwhelming array of programs available you can't afford not to consult ***THE BOOK.***

Swing by your local computer store or order direct.

The Book Company
16720 Hawthorne Blvd.
Lawndale, CA 90260
(213) 371-4012



NAME _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____
CARD NUMBER _____ EXP _____
MasterCard ☐ Visa ☐

The *pfs:* software series

A Personal Information Management System.

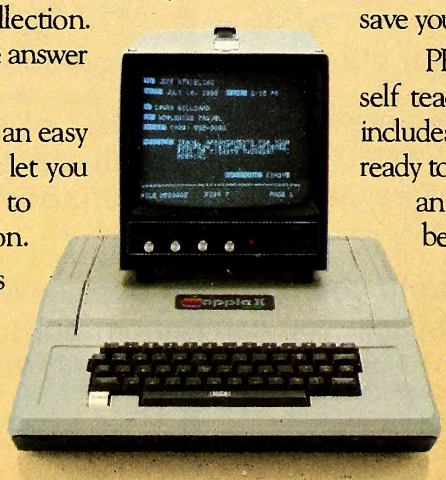
Your APPLE* computer really can track purchase orders and inventory, analyze your investment records, maintain client and patient histories, or even catalog magazine abstracts and your stereo collection. Software Publishing Corporation has the answer and it doesn't require programming!

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.

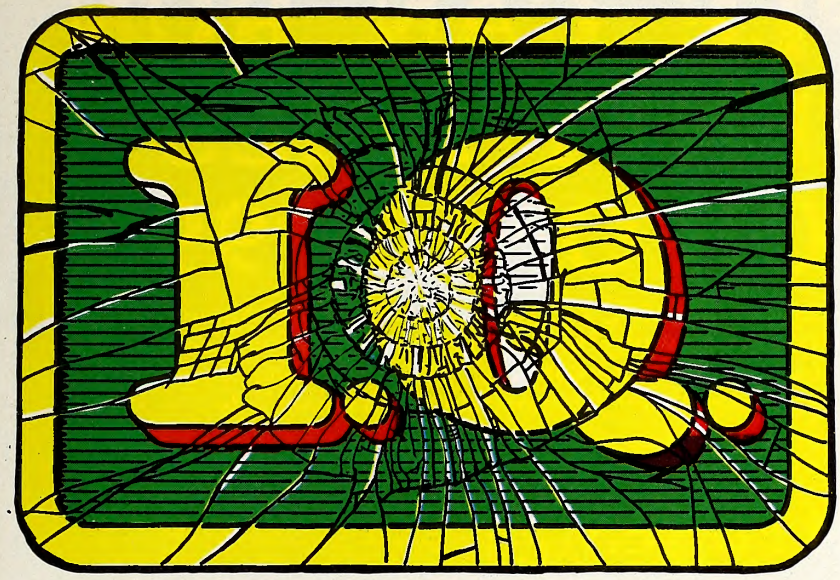
PFS and PFS: REPORT come with simple self teaching manuals plus a support plan that includes program updates and factory experts ready to answer your questions. And all of this at an affordable price. Each program is priced below $100.



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

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

Software Publishing Corporation



APPLE-DIAGNOSED INTELLIGENCE SHATTERS IQ

BY GREG VOSS

If there's one thing Bob Meeker wants to change, it's the system of education. And if there's an unusual way to go about it, he's chosen it: educational and vocational testing. The fact that the computer is one of his most important tools probably comes as little surprise to those accustomed to the miracle works of the information revolution.

To more than a thousand school districts in the United States, Canada, Australia, and, through an affiliate, Japan, Meeker's computer program acts as a clinical psychologist. The program diagnoses a group of tests designed by Meeker and his wife Mary. For each student, the computer analyzes the tests and prints out an intellectual profile that points to strengths and weaknesses in a child's learning abilities. In addition, the printout recommends remedial exercises to enhance weak abilities.

IQ Comes in Ninety Pieces. Structure of Intellect (SOI) is an educational service based on a model of intelligence developed by J. P. Guilford. Mary and Robert Meeker have used the model to break down the concept of intellect into ninety components. Tests were developed for the twenty-six of those components most needed for learning skills in grammar school and intermediate school.

Bob Meeker has written a program for the Apple that will interpret the results of SOI tests. The program allows teachers, administrators, and school psychologists to do inhouse analyses of the tests and set up individualized curriculums that cater to student needs. The formal curriculum title is Individual Educational Planning.

Structure of Intellect is a threefold process of testing, diagnosis, and remedial treatment. Tests are primarily for kindergarten through twelfth grade, but adult versions are available. Remedial exercises, available from the SOI center in El Segundo, California, help develop a child's weak abilities.

"When most people think of computers in education, they think of *computer-aided instruction,*" says Meeker. "That's not what we're doing here." SOI advocates *curriculum*-aided instruction, where teachers are helped to determine the kinds of educational experiences most needed by a particular student. The computer program recommends the curriculum. You might call it computer-aided curriculum.

SOI tests are more thorough than standard IQ tests. Among other things, they test judgment and planning skills in four areas. Meeker believes the depth of the test gives more useful information:

"Any one test is not telling you one thing, but three things. All that is in contrast to most tests, which tell you one thing; they give you one score, an IQ score, like 109. All that tells you is where you are in terms of the general population—everything mushed together into one figure.

"What the SOI test does is tell you where you are on each of the twenty-six tests and what your *relative* strengths and weaknesses are. That gets fairly complicated."

The complication of scoring and analyzing tests is the reason Meeker has resorted to a computer. To understand how a floppy disk can turn an Apple computer into a clinical psychologist, it's important to understand what SOI is all about.

Describing the Structures. Any one SOI test measures three things: (1) *abilities,* (2) *content,* (3) *levels* of complexity. Abilities indicate the type of operation an individual performs, such as identifying, remembering, or problem solving. Content indicates the type of information being worked with—like pictures, symbols, or words. Levels of complexity indicate the difficulty of the material worked with, individual units or complex systems, for example. The model is represented in the following chart:

	ABILITIES		CONTENT		LEVELS of Complexity
C	Cognition	F	Figural	U	Units
M	Memory	S	Symbolic	C	Classes
E	Evaluation	M	Semantic	R	Relations
N	Problem Solving			S	Systems
D	Creativity			T	Transformations
				I	Implications

In the chart, the letters to the left of the terms make it possible to represent the tests symbolically. The trigram EFU, for instance, stands for *evaluation* of *figural units.* Someone who has problems with this skill is likely to mistake b's for p's and to omit small words in the middle of a sentence. It's not unusual for people who are poor in MSU (memory of symbolic units) to forget where they leave things or to have problems spelling. Trouble with NSI (problem solving for symbolic implications) would mean trouble with reasoning, logic, and thought problems.

To take the category of *content* as an example, some people work better with drawings, pictures, and graphs. These are figural in content and are the province of drafters and artists. Computer programmers, mathematicians, or musicians, on the other hand, would probably work best with symbolic information such as codes, numbers, and notes. Others work best with words and ideas or semantic information. Writers and linguists are at home with this type of content.

Even without going into definitions for all the terms, you can see that with five types of ability, three types of content, and six levels of complexity, ninety combinations are possible. The Meekers pared down the number to twenty-six—perhaps just to keep their clinicians from going mad.

Because boys and girls have traditionally been socialized in different ways, they have developed different intellectual skills. Boys are encouraged to work with mechanical things (building models, working with cars and tools), but girls are not. This might be one reason why SOI test patterns indicate

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 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

that, while girls tend to be good with symbolic material required for arithmetic, boys tend to be good with manipulating figural material required in mathematics.

According to SOI theories, arithmetic and mathematics are two different skills. In the early years of elementary school, girls excel in arithmetic. Arithmetic involves mastering math tables and math facts that can be committed to memory (arithmetic skills: CSS, ESS, NSS, MSU, MSS, ESC). Mathematics, in contrast, requires manipulation of abstract concepts and figures such as logic and systems. At this stage in school, girls fall behind, not having developed skills in higher levels of complexity: relations, transformations, and implications (math skills: CFS, CFT, CSR, MSI, NSI).

Strengths and Weaknesses Identified. The advantage of the SOI over other tests is that it can pinpoint the exact area of weakness for any student. Meeker believes other tests, which lack this diagnostic capability, leave a vague impression of a student's problem areas.

"If we get an assessment that tells us, 'This student is below average,' that doesn't help us very much. We want to know *why* this student is below average. We would like some sort of diagnostic assessment that says, 'Students like this probably are not performing because they lack these kinds of abilities.' " The SOI-LA is such an assessment.

The SOI-LA (learning abilities) test evaluates twenty-six abilities. Some tests require that you identify a picture with missing parts; others have you pick irrational numbers out of a series. Another test has you listen to numbers read aloud, remember them, and write them down. This exercise is also performed visually with the numbers flashed on printed cards. In one test, you must break up several lines of letters into meaningful sentences. You might be asked to pick the one of five geometric figures similar to a target figure.

If it sounds like kid stuff, don't be fooled. There is an adult version, but grownups will even find a challenge in sections of the elementary school version.

How well you perform on any given test might be determined by your cultural background. "American Indians will put the rest of the population to shame when it comes to certain kinds of figural abilities," says Meeker. He explains that many American Indians are shepherds and so are much more concerned with small visual detail on a daily basis.

SOI has taken great pains to avoid a culturally biased test that emphasizes semantic content over figural and symbolic. Foreign and minority students who have scored poorly on traditional tests often score well on SOI figural and symbolic ones. The problem is often with the test. It's common for someone of high intelligence who is unfamiliar with the English language to score low in IQ.

Once the test is taken, scores are entered into the computer, which prints out a thirteen-page report on each student. The Apple program breaks the report into seven parts. The first five parts indicate the students' scores, three performance profiles, and evaluations of the students' abilities as related to basic and advanced levels of reading, arithmetic, mathematics, and three kinds of creativity. The last two sections prescribe exercises for specific and general weakness by finding those exercises that use a student's strong skills to enhance those that are weak.

Printout of a Diagnosis. Asked if his service was more useful to gifted or special education students, Meeker frowned in distaste at the labeling. "You see," he said, "you never get a profile that is entirely in the gifted range or entirely in the disabled range. There are always strengths and weaknesses."

The printout he held in his hand illustrated the point. "Now here is a fifth grader who scored at the superior or gifted level on sixteen of the twenty-six tests. He scored average or above on another seven. So on twenty-three of twenty-six tests, he scored average or above, most of those superior or gifted. Yet he was in the limited range on three tests, with one ability (CFC) near disabled. A limited CFC (visual conceptualization) score would indicate that in terms of reading this child is

going to have trouble. It's a shame, because the other ten reading skills are at the expected, superior, or gifted levels.

"But this tells us right where the problem is. It's right there. We can pinpoint it with CFC. Now that's a skill we can train."

Once weaknesses are pinpointed, a remedial curriculum is recommended. The computer is programmed to recommend exercises that will combine two strong areas with the weak area to be worked on. A sample prescription looks like this:

PART SEVEN: PRESCRIPTIONS TARGETED TO GENERAL ABILITIES NEEDS

A general weakness in tronsformotians is indicoted. The following exercises will train this weakness through known strengths:
***Remediate through strengths in: Cognition ..... ond seMantic .....CMT
***Remediate thraugh strengths in: Memory ..... and Symbolic .....MST
***Remediote through strengths in: Memory ..... and seMontic .....MMT
***Remediate thraugh strengths in: Evaluatian .... and Symbolic ..... EST
***Remediate thraugh strengths in: Evaluotian .... and seMantic .....EMT

A general weakness in Figural is indicoted. The following exercises will train this weakness through known strengths:
***Remediate thraugh strengths in: Cagnition ..... ond Systems ..... CFS
***Remediate thraugh strengths in: Cagnitian ..... ond Implicotions .....CFI
***Remediate thraugh strengths in: Memory ..... and Systems .....MFS
***Remediate through strengths in: Memary ..... ond Implications .....MFI
***Remediate thraugh strengths in: Evaluotian .... and Systems .....EFS
***Remediote through strengths in: Evoluation .... and Implicotians ..... EFI

A general weakness in Classes is indicated. The following exercises will troin this weakness thraugh known strengths:
***Remediate thraugh strengths in: Cagnition ..... and Symbolic .....CSC
***Remediate through strengths in: Cagnitian ..... and seMantic .....CMC
***Remediate thraugh strengths in: Memory ..... and Symbolic .....MSC

Each of the trigrams in the right column (**CMT**, **MST**, and so on) indicated specific exercises found in workbooks available through the SOI Center. The last page of the computer printout summarizes specific abilities to be trained, lists the workbooks to be used, and indicates the necessary exercises in each workbook.

Although seventy-seven exercises were recommended for the student in the sample printout, the remedial costs are relatively low. All exercises recommended for him were contained in six workbooks. Altogether, SOI has eleven workbooks for basic and advanced abilities (cognition, memory, evaluation, problem solving, and creativity). A set of books can be shared by a classroom, since students work individually, dropping the cost (twelve dollars per book) per student considerably.

If you walk in off the street, twenty dollars will see you through testing, diagnosis, and remedial recommendations at SOI. A school district can license an Apple program for an eighteen-hundred-dollar lifetime fee. The costs can be amortized in a matter of a few hundred students. The district still must pay for remedial workbooks, but, when compared to the costs of remedial therapy administered by a psychologist, the program looks very attractive.

In fact, were it not for the computer program, many districts would find adequate professional services either locally unavailable or financially prohibitive. School psychologists are a luxury such districts cannot afford.

Fortunately, costs and availability of training sources are no longer obstacles with the SOI program.

The Apple Shares the Load. Even the Apple has its limits. There is an additional section of the computer printout that requires a computer with a larger memory than Apple's. This section, titled "Clinical Analysis," is designed to recognize special combinations of problems. For example, what does it mean to be low average in CFU and low in CFC?

This combination will affect comprehension. The clinical analysis makes these recommendations for the combination:

Comprehension Abilities

Susie scared low overage an CFU
Susie is law in DFC
Ask Susie's parents ta help her make collections of rocks, shells, stamps, and leaves ta develap classificatians.
Susie is a law camprehensian student but is high in symbalic and will learn ta read best with phonetic systems.

The clinical analysis portion of the computer printout is quite elaborate and makes up six pages of recommendations. Other categories diagnosed in the clinical analysis include memory abilities; judgment, planning, and decision-making abilities; problem-solving abilities for known answers; and creative abilities. The analysis also recommends enhancement skills for areas in which the child excels. This was the case for Susie:

JUDGMENT, PLANNING, AND DECISION-MAKING ABILITIES

SAMPLE is high in EFC. In general, gifted students are more likely to score average in EFU, EFC, and ESC, but high in ESS (a rote-arithmetic skill). Students who do well in EFC are excellent at pulling together similar kinds of information into concepts.

SAMPLE is high in ESS, indicating a strength in selection af principles for solving rote-math problems.

SAMPLE is gifted in ESC. This ability will enable her to determine selection af appropriate concepts ar correct principles far arithmetic problem solving. It is the 'E' campanent that so often causes young students ta cry when they can't use their arithmetic basics (CSS).
The other abilities in evaluotian are as expected for her grade level.

There follow book referrals for teacher and student culled from a computer-resident library of reference sources on file.

Such an analysis is very useful to a teacher but, unfortunately, not possible on the Apple program. "It isn't that the concepts are so complicated," says Meeker. "It's just that the combinations, computations, and permutations are fairly high." When you consider that the computer must keep on file several paragraphs for any possible combination of the twenty-six skills and seven levels of performance (disabled to gifted), you can see what he's talking about. A 200K computer is required to handle the information.

The analysis changes from day to day as new combinations are discovered. Only trained clinicians can properly interpret new combinations. Almost every clinical analysis is, therefore, looked over by resident clinicians at SOI to ensure accuracy of diagnosis and proper revision of the program.

Apple owners take heart. SOI has plans to develop telecommunications abilities so that your Apple can talk to SOI's Wang V.S. by phone. You can enter test scores at your terminal and use SOI computing power to process the information. Your Apple will then print out the entire diagnosis or just the clinical analysis, whichever you prefer. In the meantime, for the clinical analysis, you'll have to communicate with SOI by mail.

Even without the clinical analysis, the Apple program is valuable. Twenty-two school districts lease the program. Ten of those districts use Apple computers: Longview, Wisconsin; Saint Joseph, Missouri; Hensdale, Illinois; Grand Rapids, Michigan; Manassas, Virginia; Ashland, Albany, and Milwaukie, Oregon; and Placerville and Eureka, California. Other than these twenty-two, all public and private schools work with SOI by mail.

Everyone Can Have Pros with Apple. One of the greatest untapped uses of the computer, from Meeker's viewpoint, lies in making professional services available to communities that would otherwise have to do without. Currently, there are two obstacles to widespread use of psychological and educational diagnostic services. One is availability of trained professionals; the other is cost.

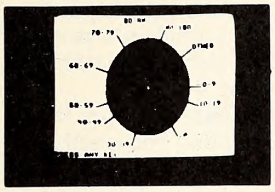
What happens when a diagnosis can be made, but no remedial treatment is available? Meeker cites the field of vision therapy as an example where trained professionals are simply unavailable in remote areas. What do you do if you discover correctable vision problems but you live in an area where there is no trained vision therapist? Some sort of paraprofessional training becomes necessary to spread knowledge and skills.

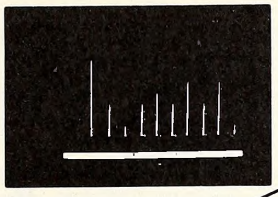
GET 120% VALUE FOR YOUR
PROGRAM PURCHASING DOLLAR
WITH

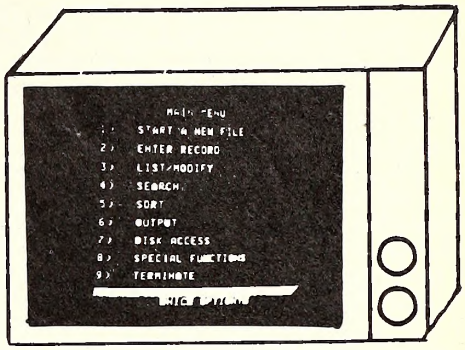
THE DATA REPORTER

MORE THAN JUST A DATABASE

Version 2 of the versatile Modifiable Database

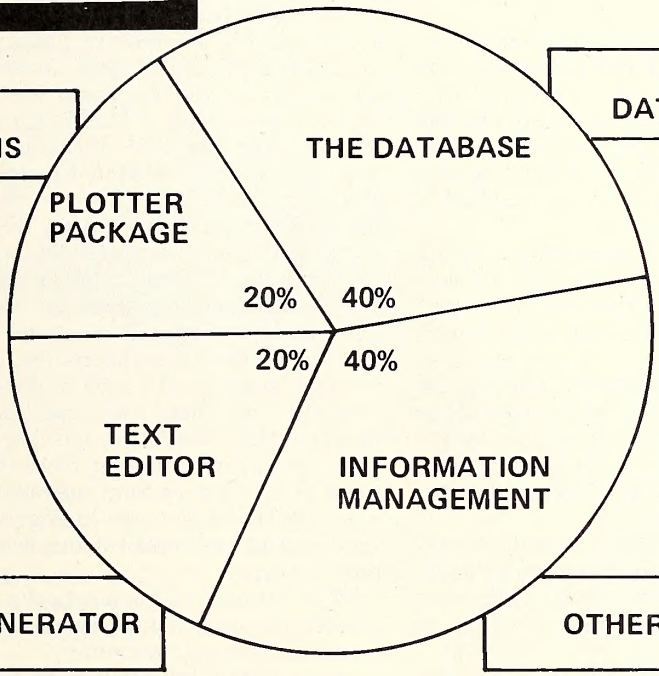






DATA PLOT & ANALYSIS

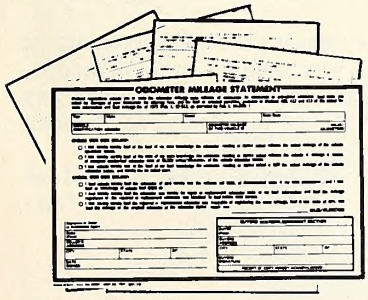
- Data may be plotted in a variety of formats such as scatter graphs, line graphs, bar charts, and pie charts.
- Ranges, minimums, maximums, means, standard deviations, correlation coefficients, etc. of any number of data files can be calculated



POWERFUL DATA MANAGER

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

REPORT GENERATOR



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

OTHER FEATURES

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

SYNERGISTIC SOFTWARE

ORDER YOURS TODAY!

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

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

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

A possible solution is being tried in a New York school district where parents were taught to administer visual therapy for local schoolchildren under the direction of a professional therapist.

The Apple program is one means of training paraprofessionals in the field of educational psychology. By working with the computer analysis, text materials, and inservice manuals supplied by SOI, teachers and administrators become skilled in the diagnosis of learning problems. Typically, it takes someone working with the SOI tests in the classroom two to three years to become as good at diagnosis as the computer program. In the meantime, the computer serves as a tutor. Meeker likens the relationship between computer and teacher to that between psychologist and intern.

The use of computers to train paraprofessionals might be one example of Clive Smith's notion, given in "Life in the Information Age" in New Age magazine (September 1981), of the highest expression of informational values and technology: "Mass utilization of decentrally produced knowledge (using computers and networks) for personal and social transformation."

The Price Is Right for Schools. Probably more than anything, it's the cost factor that is bringing computers into the world of education. A district that can't afford a school psychologist often can afford a computer. Many districts already have computers for clerical purposes or for instruction. "By the time you get the cost of computing down to where it is with the Apple, there's hardly anybody around who can't afford to buy an Apple and our program," says Meeker.

Meeker says ten years ago it wasn't a possibility to bring computers into education. He should know. In 1956, he was working for System Development Corporation trying to promote computer-aided instruction. The main impetus behind computers now being accepted is cost.

"Short of tripling a student's achievement, nothing research could have done would have made a difference. Now that computers are down to fifteen hundred dollars, you're going to get computer-aided instruction. You're not going to get it because anybody demonstrated that it's any better; it's because it's available."

The single greatest boon to computers in education, according to Meeker, would be a public decision to go to a voucher system. Again, it's cost more than educational value that counts. It's cheaper to buy a computer and software than to pay for aides and instructors.

Meeker also sees the educational problem from an employer's perspective. He believes that computer education not properly administered can be somewhat dysfunctional. He cites local high school students whom he has employed. Trained in computer science classes, they are good at programming but lack a business perspective of time and resource limitations.

Students who come to work for Meeker have been rewarded for two things by their high school teachers: writing elegant programs and writing clever programs. Because they lack a business perspective, they don't ask crucial questions.

"They don't ask questions like 'How many times is this program going to be run? One time? Ten times? Every day for the next three years?' Now, those are three different programs."

Students often use too much of the limited resources and cannibalize someone else's program. They will use up more memory and computer time than necessary. They don't ask how much of their own time they should put into a program.

Meeker says teachers don't have the orientation toward business they need to guide the student to ask the right questions. "I'm not sure that I would expect them to," he says. "It's new."

Study Habits Aren't Always Work Habits. The problem is not one that applies only to computer science. Meeker feels it's a problem with the current system of education:

"There used to be an obvious correspondence between the skills we learned in school and their application in the work world. What would right the ship of education has little to do with computers. It has to do with the correspondence between education and work. There's almost none now."

To Meeker, school does not foster an attitude of responsibility that is required at work. No one depends on a student to carry on activities at school. At work you would probably be fired for bringing in an excuse after an absence. Yet, at school, that's what children are taught to do.

"Try having someone who is in elementary school call in sick ahead of time," suggests Meeker. "You know what they'll say? 'Why the hell are you calling us?' It doesn't matter if they know whether you're going to be there or not."

This indicates several things to Meeker. First, the fact that you don't have to call says the school is not counting on you being there, except in the sense of occupying a seat and getting money for your presence. So a medical excuse is just as good. Secondly, nothing that happens depends on a student's presence. It's not that, because a student didn't show, something that was going to happen now won't happen.

After getting that message for twelve to fourteen years, it's little wonder that students have trouble accepting responsibility at a job where they're expected to be present and perform.

Computing Alternative Futures. Meeker has a vision of a school that would prepare students better for the world of work. Why not make schools units of production in society? Everybody is capable of working, and everybody should be given the message that their contribution is valued. "Work education" would include kindergarten through twelfth grade. All students would be given work and receive wages. First graders might receive twenty-five cents an hour while seniors would get minimum wage. The imagination is the only limit to the number of jobs students could perform: running messages, cleaning blackboards and restrooms, checking out equipment, tutoring. Students would receive two report cards: one for academic performance, one for work performance.

Now, schools take kids to the dairy or the bank to see how they are run. "Better that you run a business at school," says Meeker. Kids would get checks. They'd have a credit union. They could borrow money from the credit union. Kids wouldn't see a business only from the outside. They'd be part of a business. By the time those kids graduated, employers would be knocking at the school doors because those kids would know how to work.

The ultimate test of a school's success would be how well its seniors performed key functions within the school. A senior would act as head secretary.

Most people tell Meeker he's crazy when he suggests this. But think about it.

"What are you telling industry when you won't trust one of your seniors to do a job for which you have given them an academic degree? You give someone a business degree and say they are ready for the work world, but you don't really mean it."

SOI and others are working to change all that. With a task that big in front of them, you can bet they're grateful for computers.

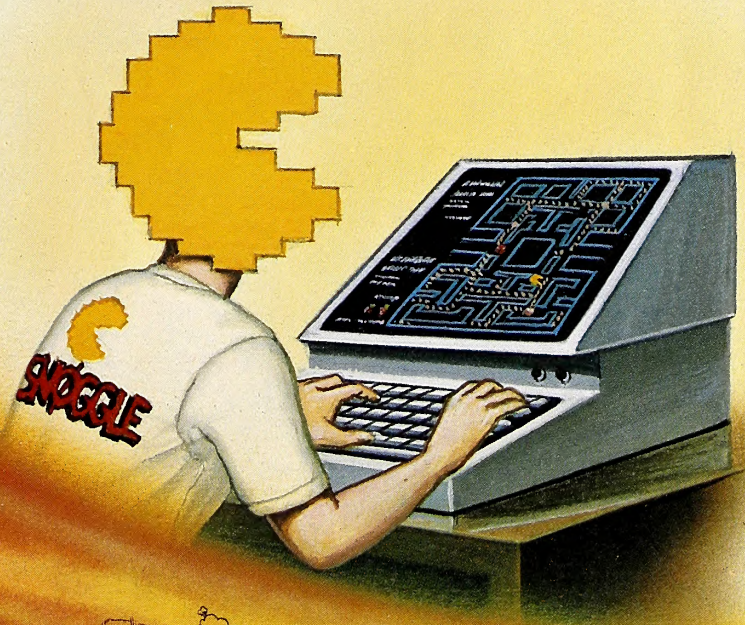
* * *

Mary Meeker is an educational psychologist who studied with J. P. Guilford in the early sixties. She later worked at Loyola University of Los Angeles as a professor, where she began to form the SOI Institute from Guilford's theoretical model. In 1974, the SOI program outgrew facilities at Loyola, so it incorporated and moved to El Segundo, California.

Mary Meeker had been practicing psychology out of her home, and Robert Meeker was mystified by her intuitive approach to the problems of young schoolchildren. Aware of the complexity of psychology and personality, Robert felt there had to be a way to quantify a method of analysis objectively.

Using the SOI model, the Meekers eventually developed a system of objective tests. Mary's intuitive insight and practical experience combined with Robert's background in computers to produce the tests, computer diagnoses, and remedial curriculum that now constitute the SOI program. ■

HI-RES MASTERPIECES

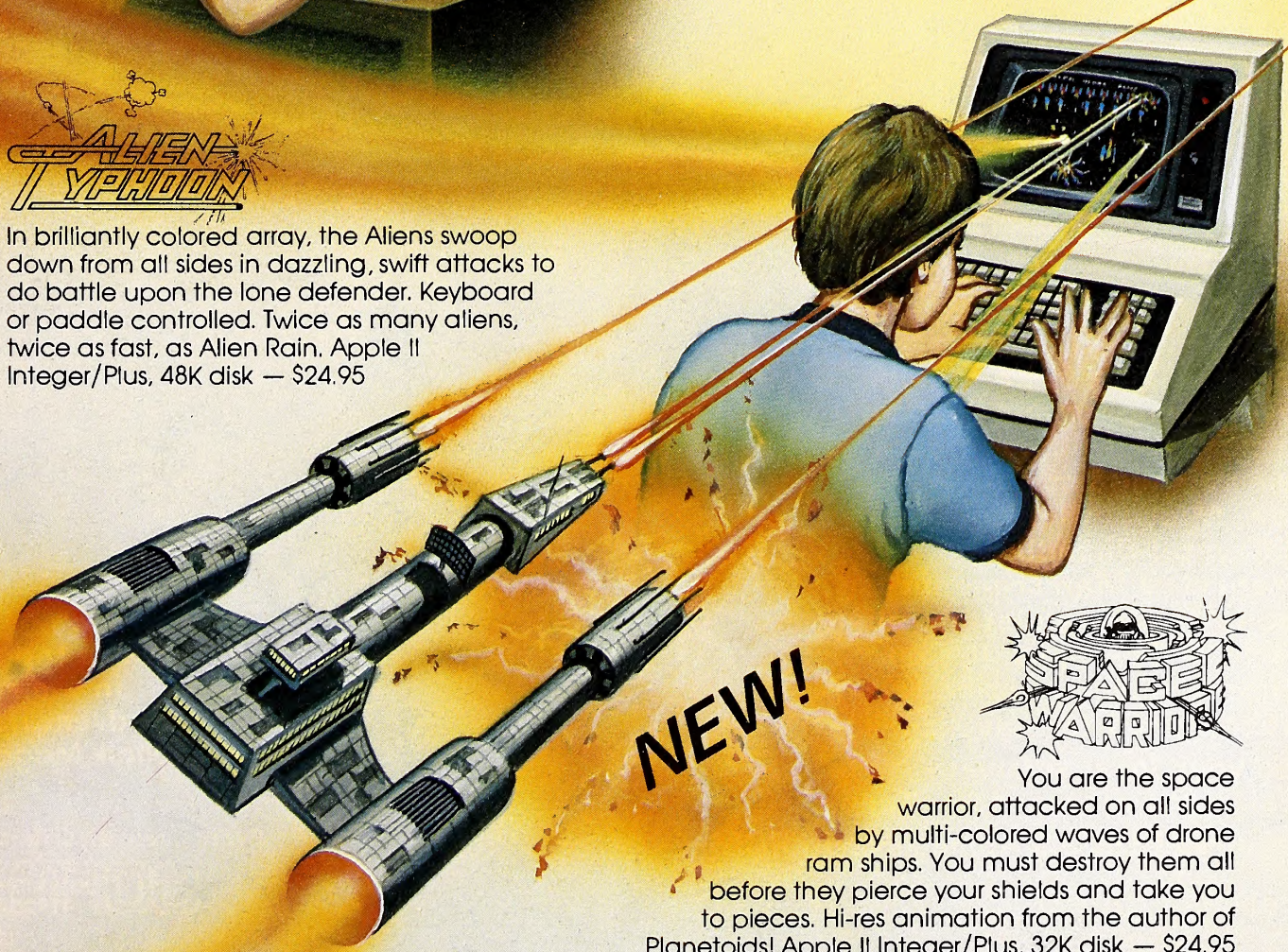


SNØGGLE

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

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

See your local software dealer and discover Oo-Topos.

Requires DOS 3.3 Applesoft* in ROM

From

SENTIENT SOFTWARE

P.O. Box 4929, Aspen, Colorado 81612
(303) 925-9293

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

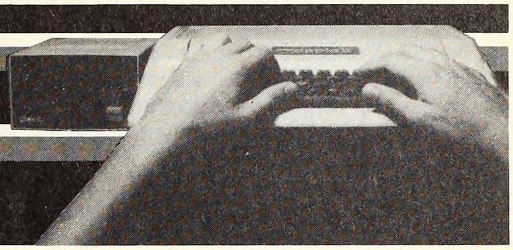
Exclusively distributed by

SOFTSEL

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

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

TRADETALK



☐ **Nasir Gebelli,** author of several best-selling hi-res arcade games, has formed his own company. **Gebelli Software, Inc.,** is located in Sacramento, California, and specializes in entertainment software for the Apple and Atari. Nasir's first game on his own, *Firebird*, was released October 1, and Nasir promises many more to follow.

☐ It's been three years since **Southwestern Data Systems** (Santee, CA) opened its doors, and they finally got around to throwing the first in a series of anniversary parties. This kickoff event at Mission Bay included sailing, jet skiing, and a lot of reminiscing. The night ended with owner **Roger Wagner** playing the guitar. According to office manager **Tom Burns,** his boss "crooned to the moon" for more than four hours. The fact that most people remained till Wagner finished "indicates that he's pretty good," said Burns, a loyal employee. And we all thought Roger only knew assembly language.

☐ **Softsel,** formerly of Marina Del Rey, CA, has moved to bigger quarters at 8295 South La Cienega Boulevard, Inglewood, CA. Phone (213) 670-9461.

☐ **Daniel S. Bricklin,** executive vice-president of **Software Arts** (Cambridge, MA) is to be awarded the 1981 Grace Murray Hooper Award at the **Association for Computing Machinery**'s annual conference November 9, 1981, in Los Angeles. The award, given in recognition of computing achievements made by persons before their thirtieth birthdays, honors Bricklin for "his contributions to personal computing and in particular to the design of *VisiCalc*." He conceived the visual calculator idea while attending Harvard Business School, where he received the M.B.A. in 1979. The actual program was written by Robert Frankston, now Software Arts president.

☐ **Larry Bane** has been appointed general manager at **On-Line Systems** (Coarsegold, CA). Formerly production manager for the firm, Bane has also served in the Los Angeles construction industry and was county field supervisor for Roto-Rooter for ten years.

☐ **Apple Computer** (Cupertino, CA) is distributing *Micro-Courier* and *Micro-Telegram,* manufactured by **Microcom** (Boston, MA). *Micro-Courier* is an electronic mail package; *Micro-Telegram* connects Apples with Western Union to send or receive TWX, Telex, and mailgrams.

☐ **Epson America,** maker of the MX series of printers, is expanding its administrative headquarters to a facility triple its former size. New address is 3415 Kashiwa Street, Torrance, CA. Shipping, receiving, traffic, and technical service departments can still be found at the company's Hawthorne Boulevard address.

☐ **Donald Brown**'s *Swordthrust* game (**CE Software,** Des Moines, IA) held the distinction of being the only computerized fantasy roleplaying tournament game at Worldcon, the world science fiction convention held last month in Denver, Colorado. "I suspect people enjoyed it," said Brown. "I have outdone myself in evilness, wickedness, and underhandedness in designing the dungeon."

☐ The world semiconductor market will reach $74 billion by 1990—a 23 percent per year growth rate during each of the next nine years—according to Dr. **Julius Muray**, a staff scientist at **SRI International**'s engineering sciences laboratory. Muray is launching an in-depth program to study the microelectronics industry—its technology and trends—with emphasis on how these changes may affect semiconductor manufacturers and their customers. Designers, manufacturers, and production equipment users will participate in the program. Further information can be obtained from Dr. Julius Muray at SRI International, 333 Ravenswood Avenue, Menlo Park, California.

☐ **Orange Micro** has signed on with **Estey-Hoover Advertising and Public Relations.** While directing part of Orange Micro's media advertising budget of approximately $300,000 toward newspaper, business, and trade publications, Estey-Hoover also plans new collateral materials, trade show activities, and public relations services. Orange Micro is a national mail-order firm with retail outlets in Anaheim and Sherman Oaks, California, and other store coming on line in San Jose, California.

☐ "Micro Magic" is a new TV program designed to familiarize the American public with concepts, use, and availability of personal microcomputers. Both business and entertainment applications will be explored on the thirty-minute weekly show for major cable and satellite networks, with additional distribution through selected UHF stations in major markets—potential viewing audience more than eight million. Each program will be composed of two twelve-minute guest appearances and twelve thirty-second commercial spots, with **Robert Chesney of Chesney Communications** and **William V. R. Smith** of **Artsci Inc.** acting as co-hosts. First "Micro Magic" will air in mid-November. For program guide or other information, contact William V. R. Smith at Chesney Communications in Hollywood, California.

☐ *Sofprotex* (Belmont, CA), a new **Government Copyright Services** division, has released a report on computer software copyrighting. A company spokesperson said, "Major corporations are finding themselves involved in litigation over computer crime at an alarming rate of increase. . . . With new legal decisions such as the recent ruling on patents involving software, and the general lack of available information, we feel this report will . . . become a necessary addition to the library of anyone supported by software development." $25 at various outlets, or order from the publisher at Softprotex, Belmont.

☐ **Dave Gordon,** founder and former president of **Programma International,** has established a new firm, **Datamost,** a support house for Apple software and

books. Among the products Datamost will handle are *Write-On* word processor; arcade games *Thief* and *Smack Attack; Expand-a-Port* expansion switch system for multiple paddle and joystick games; and Randy Hyde's book, *Using 6502 Assembly Language.* Datamost is located at 19273 Kenya Street, Northridge, California.

☐ **Integral Data Systems** (Natick, MA) has cut the price of its eighty-column model 460 *Paper Tiger* from $1,295 to $995; the model 460 with graphics has been reduced from $1,394 to $1,094. The company's 132-column model 560 is down from $1,695 to $1,394. **Peter Eisenhauer,** vice-president of marketing, says the price reductions were part of a program to position Integral Data as a leading supplier of microcomputer printers.

☐ **Queue** (Fairfield, CT) distributes the *Library Skills, English Skills,* and *Verbal Skills Improvement* packages by **Right On Programs** (Huntington, NY). Software is designed to strengthen communication abilities in first through sixth graders.

☐ Software publishers and other producers of microcomputer merchandise are invited to participate in **Irv Brechner Enterprises'** cooperative mailing of new product news and literature. Brechner is mailing samples and brochures to about eight hundred authorized Apple dealers this month; and for a minimum of $100, companies with products for Apple owners can still reserve space. Brechner asks that companies ship one thousand copies of the product or pamphlet to Irv Brechner Enterprises, Livingston, New Jersey. Fees can be determined by telephone inquiry: (201) 731-4382.

☐ **High Technology Software Products** (Oklahoma City, OK), publisher of *Information Master* and *Transit,* has authorized **Bell-ITT's** Belgium affiliate, which distributes Apple computers in Benelux, to translate its software user manuals into foreign languages. High Technology president **Charles Weddington** says that, although many European Apple owners speak English and have used High Technology programs successfully, there is significant benefit in having the user manuals in native languages.

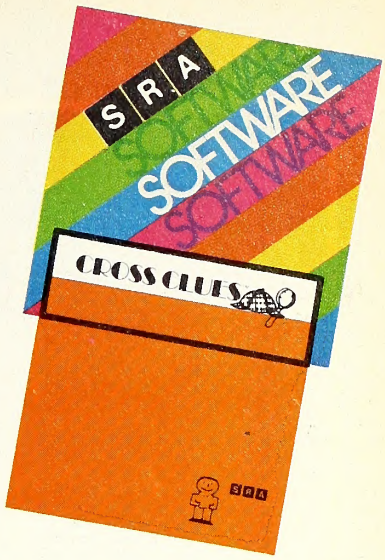
☐ "Distributed Processing Update: A Management Seminar" will be held November 9 through 11 at the Westpark Hotel, McLean, VA. Sponsored by **Data-Communications** magazine, it'll give a comprehensive, up-to-date overview of tools, techniques, requirements, and benefits of distributed processing systems, as well as direction in selecting hardware and software best suited to participants' needs. Registration fee, $650. For more information, contact McGraw-Hill Conference Center, 1221 Avenue of the Americas, New York, NY.

☐ **Nelson B. Heller** has been named director of research for **SFN Companies** (Chicago, IL), a diversified publishing company. Dr. Heller was formerly president of Educational Programming Systems, a Saint Louis publisher of computer software products. His new position will be as a corporate-wide resource person in electronic publishing for SFN headquarters and its operating companies. Computer and television based instructional formats and data based publishing offers a lot of opportunity, notes Heller. "It's an area most publishing companies are now taking every seriously."

☐ **SSM Microcomputers** (San Jose, CA) has announced that **Crane and Egert Corporation** of Elmont, NY, is now a stocking distributor for their products. SSM says the agreement points to a new marketing direction because of C&E's connection with the industrial community. SSM will continue to sell their products to original equipment manufacturers directly and to end users through computer retailers. Other news around SSM: **David J. Wertzberger,** director of marketing and sales, co-chaired a major technical session last month at Wescon/81 in San Francisco titled "Microcomputer Bus Structures—What the Future Holds." ❑■

Now, it's mind over reflex in an original NEW word game!

CROSS CLUES™



The computer umpires. Two players or teams compete. Uncover the hidden words. The best combination of skill, intellect, timing and luck wins. Appeals to young teens through adults.

New. Unique. Original.

Cross Clues is not a conversion of an existing game. Clues coaxed from the computer help both you and your opponent. The outcome remains in doubt and excitement builds as the finish nears. Catch the interest of family and friends who never before enjoyed computer games (as well as those who did!)

Cross Clues is a game of words for people. Play it at your next party. Watch your computer become a social 'event' of fun and intellectual stimulation.

Dealer inquiries invited.

***** 50 different word games for $29.95. *** For Apple II* or Apple II Plus*, Applesoft*, Disk Drive and 48K RAM. *****

Since words can't fully describe CROSS CLUES, see your local Apple Dealer today for a demonstration. If unavailable at your local dealer call SRA toll free at 800/621-0664. Master Card and Visa accepted. In Illinois call 312/984-2000.

SRA®
Science Research Associates, Inc.
Software Products Department
155 North Wacker Dr.
Chicago, IL 60606

*Trademark of Apple Computer, Inc.

John Lilly and His Apples

Photo by Tom Fitz

BY MELISSA MILICH

My goal is to break the interspecies communication barrier with microprocessors and minicomputers.

John Lilly, M.D., biochemist, engineer, inventor, philosopher, author, and dolphin researcher

The water was dark and deep, and the young girl was not alone. There was a lot to be scared of and nothing to be scared of. Suddenly a gray head surfaced without making a ripple, then a large brown eye, then a smiling beak. She reached out and stroked four hundred pounds of weight, and she knew it would not hurt her.

"Grab its fin! Grab its fin!" yelled some faraway voices from land.

The dorsal fin of a dolphin resembles that of a shark, but when a human grabs onto the dorsal fin of a friendly dolphin, she may get the ride of her life.

Joe, a two-year-old *Tursiops truncatus* (bottle-nosed dolphin), was determined to make it a wild one. He and the girl shot through the underwater at a speed only a dolphin would know. His fellow tursiops, Rosalie, swam alongside and sandwiched the girl in. They could crush her, bite her, or drown her, but the oversized fishlike creatures chose to play with her.

And she could tell they had plans. They whistled and clicked back and forth and occasionally glanced back at their passenger. There was a conversation going—gossip, no less. She couldn't understand a word of it. But a relationship was established, and dolphins and human continued to dive, leap, and roll in an undersea world, their world.

After forty-five minutes, choking on all the salt water she had swallowed, she tried to leave the tank. But Rosalie followed her to the side and grabbed the girl's arm in her mouth. The dolphin pulled but did not bite.

"She doesn't want you to leave," said one of the researchers sympathetically. "No one's swum with them all day."

Mutual Admiration Society. The dolphin is the only creature "that loves man for his own sake" observed the Roman philosopher Plutarch nineteen hundred years ago. Aristotle, likewise, noted the beauty and wonder of porpoises during the early millenium. In 1981 A.D., at the Human/Dolphin Foundation in Redwood City, California, scientists are trying to communicate with dolphins via computers. Here, in one man, live and work a modern-day Plutarch, a modern-day Aristotle, and a modern-day controversy.

Dr. John Cunningham Lilly, at sixty-eight, has done more than many could manage in one lifetime: he's been a neurophysiologist/biophysicist studying brain sizes, the author of several books on mind and psychology, and a distinguished professor at Harvard. But he is remembered most for his unconventional work: experimenting with LSD during the sixties, under government approval, and inventing the sensory-isolation tank—both of which projects led to further mind research—and trying to talk with dolphins.

Lilly is the leading authority on dolphins in the scientific world, and trying to break the interspecies communication barrier takes up most of his time.

To communicate with dolphins, Lilly believes man must first interact with them, and swimming with the dolphins is a necessary part of this. But ask the veteran of isolation tanks and LSD trips what it was like the first time he jumped into a dolphin tank, and Lilly will admit frankly that he was terrified.

"I knew their power and strength and what they could potentially do," he says, his voice trailing off.

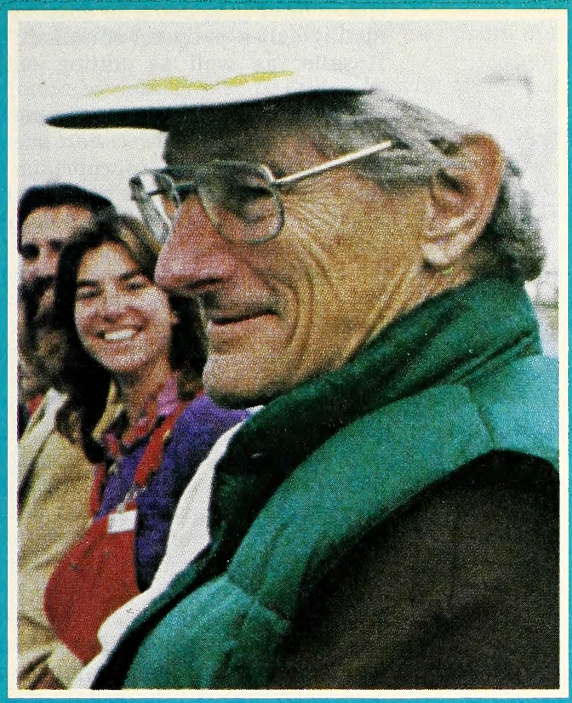
His co-workers call him "essentially, a practical man." In the early seventies when NASA embarked on a twenty-million-dollar project to send a brass plaque into outer space to communicate with extraterrestrial beings, Lilly was asked to participate in the project. Most scientists probably would have jumped at the chance. Lilly, however, refused in favor of the dolphins.

"If you won't spend any money to investigate alien intelligence next door that you can touch and look at, I'm not going to spend my time on that," he reportedly told NASA.

Computer Is Bridge To Span Language Gulf. It was also this streak of practicality that lead Lilly to explore the possibility of using computers with his dolphin research. He began his work with interspecies communication in 1957 but found that man alone could not keep up with the complexity of the dolphin language. Dolphins have several ways to make sounds, and some of them use frequencies that are beyond the human range of hearing.

With the advent of minicomputers and microprocessors in the mid-seventies, Lilly saw a way to break through the language barrier. Although people by themselves still could not

Talk with the Dolphins



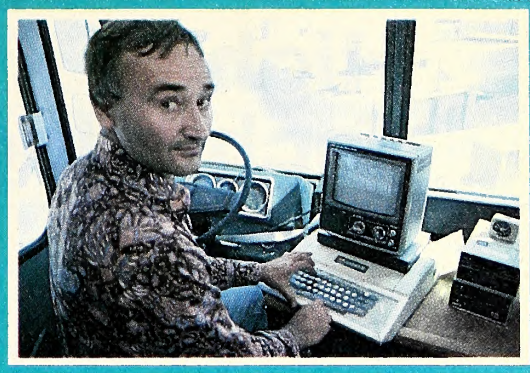


Photo by Maggie McCurk

Photo by Jennifer Yankee

communicate fast enough, computers could, and Lilly found that the dolphins would interact with them.

"The dolphins' knowledge is much more ancient than ours," Lilly said. "Dolphins have had brains the size of ours, and larger, for fifteen million years; and we've only had our present brain size for the last three hundred thousand years."

Higher Intelligence Here on Earth? The dolphin brain is, on the average, about forty percent larger than the human brain and has more convolutions in the fissured neocortex area that holds the capacity for language. All this indicates that cetaceans are extremely intelligent animals.

As intelligent as humans? Not a chance. Try more intelligent.

"I would say yes, but in a very strange and alien way." Lilly paused for a moment in thought.

"I never talk about intelligence per se of a given species or a given individual," he says, "but the only way intelligence is ever measured is by communication."

Although he spent years dissecting brains in laboratories as a neurophysiologist, Lilly says he never makes a judgment of intelligence until he talks to a person—or, he hopes in the near future, to a dolphin.

"Intelligence has no meaning outside the operational communicative mode; but you take a very intelligent human talking to a very intelligent dolphin, and they'll have an exchange that nobody else will have." And then he smiles slightly as if he is keeping a great secret that most human beings wouldn't be ready to share.

Janus Looks to Land and Sea. If Lilly hasn't already realized his dream, the site of the interspecies communication breakthrough will probably be at Marine World Africa USA in Redwood City, California, where space has been allotted for his research. Just east of the parkway lies a mobile laboratory that few tourists ever see. Standing apart from the rest of Marine World, it is here that Lilly is trying to learn one of the best-kept secrets of the universe.

The first interspecies communication system, called Project Janus, is made up of two Apple II microcomputers and a voice synthesizer hooked to a mainframe PDP 11/04.* Outside the lab, two bottle-nosed Pacific Ocean dolphins are cavorting in a holding tank, waiting for their next session with humans.

Joe and Rosalie *Tursiops truncatus* were donated to the foundation by Joe and Rosalie Levine, producers of the movie *Day of the Dolphin,* which was based on Lilly's work and starred George C. Scott.

Lilly needed to work with dolphins that were not already trained as show animals for amusement parks or Hollywood. Joe and Rosalie were caught in the Gulf of Mexico thirteen months ago and are approximately two years old. They have distinct personalities. Rosalie is much more adventurous and outgoing; Joe will usually follow her lead.

Janus sessions are usually run three times a day. It's mid-afternoon and time for the second session. While Joe and Rosalie are fed herring and smelt, chief system programmer John Kert is inside the trailer running the controls.

Declaring Their Presence. What goes on in the small, cramped trailer makes a fairly dramatic show. It's a noisy place, with the computers whirring, printer cranking out hard copy, and fans buzzing. The racket is occasionally punctuated by a dolphin's shrill whistle coming through speakers on the wall from the tank outside.

Kert cocks his head toward a dolphin sound. "That's Rosalie's signature whistle," he explains. "It means, 'I'm Rosalie, and I'm here.' "

To facilitate the communication process, a third language was invented that was neither human nor dolphinese. The computer acts as a medium between man and dolphin. There are currently three programs—*Janus I, Janus II,* and *Janus III*—being used to communicate with Joe and Rosalie.

From inside the lab, a researcher controls an Apple keyboard that, depending upon the program on disk, either produces tones inside the tank or projects images on an underwater screen the dolphins can watch.

Janus I produces individual tones, and the dolphins have learned to reproduce with their whistles the same number of tones put out over the system. *Janus II* is a very similar program to *Janus I,* with the tones slightly modulated.

In *Janus III,* one or more letters are typed on the keyboard, corresponding to a vocabulary that includes words like fish, ball, and swim. The sounds that go into the tank via the underwater speaker and are transmitted back from the dolphins come out in hard copy on the laboratory printer. That mate-

*Janus has a double meaning in this name. Practically, it stands for Joint Analog Numerical Understanding System and, symbolically, it refers to Janus, the double-faced ancient Roman deity of gates, doorways, and new beginnings.

rial is later graphed into data by a researcher. The computer distinguishes between sounds it makes and those of Joe and Rosalie, as well as editing out miscellaneous underwater sounds.

Cryptic Conversation on Paper. Sounds are shown as configurations of letters. You can see where the dolphins already use some words in appropriate context. Among the biggest barriers to interspecies communication are the differences in frequency. The high-pitched dolphin whistle is in the range of two thousand to four thousand hertz (cycles per second) while the average human voice ranges between four hundred and three thousand hertz. There is some overlap, but the cetacean whistle is usually so high-pitched that humans cannot distinguish the difference in modulation.

The computer, however, drops the whistle to a comfortable, easy-listening range for humans. It also differentiates between the modulation with the letter symbols on the hard copy. Increasing frequency corresponds to increasing order in the alphabet. For example, if the printout reads IJKLMQ, the frequency is increasing. If this pattern is plotted on a piece of paper, the shape of the modulation and the frequency will help the scientist determine the meaning of the whistle.

According to Kert, there have been several instances where the dolphins matched their whistle with the tone the computer made. They have even lowered their whistle to accommodate the frequency of the human range. "But we're still in the beginning stages." he says.

At any rate, Joe and Rosalie are definitely interested in this project. They especially like the underwater split screen where they can watch themselves on one side while computer output from the lab shows on the other.

"You ought to see them in front of a motion picture camera—they'll really perform," says Lilly, the corners of his mouth turning up slightly. "They'll do all sorts of tricks; they love cameras. And if you show them the pictures later, they're delighted."

Only Benevolence in Dolphin Ethics. Lilly believes human beings have a lot to learn from the dolphins. Dolphins have a code of ethics much higher than our own: they never steal food from other dolphins; they don't have wars; and they're extremely benevolent to humans. Many cases have been recorded of humans who would have drowned had they not been pulled safely to shore by dolphins.

As a medical doctor, Lilly is interested in the dolphin's ability to cope with stress. In a high-pressure situation—such as the time Joe and Rosalie were caught and then flown from Mexico to San Francisco—they have an amazing capacity to calm down. Their heart rate slows and their blood pressure drops. Humans in stressful situations tend to have just the opposite bodily reactions: the fight or flight reaction and an inability to cope with stress that doctors now believe causes many of our chronic diseases.

The dolphins' intelligence keeps them alive, notes Lilly. It's used for their survival: mapping the bottom and surface of the ocean, learning about storms and what they do, knowing when to surface, traveling, surfing on wave fronts, passing on knowledge of the history of the earth from older to younger dolphins, and learning how to relate to various other species, including man.

"So I think they volunteer to be captured, and, hoping they'll be put back in the ocean, they get gossip from the land about humans," Lilly hypothesizes. In his book, *Communication Beyond Man and Dolphin,* Lilly proposes an oceanarium through which whales and dolphins will rotate. The oceanarium will be hooked up with satellites on sea buoys so the dolphins in captivity can communicate with those in the wild. When the communication devices become that sophisticated, he says, humans will also be able to talk to cetaceans in this network via telephones.

"But we'll send the dolphins and whales back to the wild after a year, and we'll take only volunteers," Lilly emphasized.

An End to Being Alone. "The most important thing in our work," according to Victor di Suvero, a member of the Hu-

man/Dolphin Foundation board of trustees, "is the exploration of the possibility that the loneliness of man could finally be broken.

"We may have finally found somebody else to talk to that's been around for a long time. In the context of the age of the universe, we're kids, and we may have found a big brother."

The human/dolphin researchers follow the teachings of the nineteenth-century French educator Piaget: expect more from the child, and you will elicit more. Ed Ellsworth, a part-time volunteer with the foundation and a research psychology graduate student at San Francisco State University says dolphins cannot be trained in the usual research-oriented way.

"You can't train them like laboratory rats. You can try to do it, but, depending upon the dolphin, they usually won't cooperate," he says. "They get bored very easily. And that's an indication of intelligence to me."

The Human/Dolphin Foundation is totally supported by public funds. Approximately thirty thousand dollars a month is needed to keep the dolphins fed, the computers running, and some of the staff paid. The money is raised through benefit concerts, from royalties on Lilly's books, and, occasionally, from some large donations.

Apple and Jobs Take an Interest. About a year ago, Steve Jobs, one of the founders of Apple Computer, saw an interview in a local newspaper with John Lilly and noticed in an accompanying photograph the corner of one of his Apple computers. His curiosity aroused, he contacted Lilly and found that their Apple was indeed very significant to the project. In fact, they liked it so well, they wanted another.

That was an easy enough request for Jobs to grant, but Lilly also suggested that a smaller, more compact computer, which could be strapped to a human and taken into the water, would speed along their research. This watertight prototype is still in Lilly's imagination, but Lilly and his dolphins hope Apple or other researchers will soon decide to tackle the project.

Kert is especially excited about the possibility of his crew interacting in the water with the dolphins and having the computer there to translate the conversation in real time. In the meantime, the researchers are doing everything possible to reduce the interface.

Meeting on Equal Turf—uh, Water. Shallow water—a nineteen-inch depth—is an ideal environment to produce bonds, since the mobility and freedom of human and dolphins would be equalized. For the past ten months, the researchers have been trying to get Joe and Rosalie to enter voluntarily a shallow-water channel that connects to their deep-water tank. Dolphins, however, have a natural fear of going into shallow water, where their swimming capacities are impaired greatly.

To reduce the interface, a movable floor was built for the main tank. Raising the floor can lower the depth of the water in the tank to three feet—which is better but still not quite as good as getting the dolphins into the channel.

"What would really excite me is having Joe and Rosalie swim into the channel by themselves," Kert says wistfully. "I know it doesn't sound like a big breakthrough, but communication would probably move along so much more rapidly if we could get them in there."

So, every day, the researchers devoted a certain time pe-

Photo by Nancy Ellison

riod training Joe and Rosalie to go into the channel, with the hope that the dolphins will eventually feel comfortable enough to enter on their own. They raised the floor to the three-foot-deep level; then the researchers used either a net or rubber bats to lead the dolphins into the channel. Once they got them inside, trainer Tom Fitz blew the whistle around his neck, a sign of approval for the dolphins. As soon as they were let out, Joe and Rosalie were rewarded with fish. But still they would not venture into the channel without a lot of prodding.

It is now late afternoon. The researchers, tired after a full day's work, are huddled together in a group at the far end of the tank discussing tomorrow's research plans. Lilly has left to lecture in Hawaii, Florida, or heaven-knows-where. Dolphins and humans are ignoring each other for the time being. Kert and his crew are embroiled deeply in their research plans, while Joe and Rosalie are playing with a deflated innertube.

But the innertube floats away from the dolphins and drifts into the channel. Chattering madly, Rosalie races after it; Joe hesitates briefly, then follows.

Ellsworth is the first to notice. He points but is too surprised to shout out more than "Hey!"

The rest of the dolphin researchers finally see what they've been waiting nine months for. They start shouting and clapping wildly as the dolphins turn around and swim back to the safety of the deeper tank.

A Breakthrough Acknowledged—by Joe. But Joe and Rosalie seem to be mindful of the excitement. This time Joe grabs the innertube, almost like a security blanket, and swims straight into the channel—Rosalie not far behind. Now even dignified John Kert is jumping up and down and clapping wildly at, finally, a major breakthrough.

And if John Lilly could have been there, no doubt the corners of his mouth would have turned up slightly—but he might not have been too surprised.

He probably told the dolphins to do it. ◨

Dolphins Seek Programmers

The only way we might break through is to give the dolphins the benefit of our technology, says Dr. John Lilly; and the first place to start is with computer games.

The Human/Dolphin Foundation is now seeking people to develop new software for the dolphins—software that would be similar or analogous to the games humans play, complete with graphics. Programmer John Kert says that, since they are using Apples extensively, anything written for the Apple could probably be used; and it doesn't have to be anything too sophisticated, he adds.

In detail, Lilly explains that "what we'd like to have is a program that could recognize these inputs and allow the dolphins to select new programs already stored in the computer. They could be asked questions, be shown diagrams on the monitor, and put back sounds that could symbolize the game."

The first game should be one that would teach the dolphins the language of the computer and the language of the game. This should be rather simple and very direct; the instant the dolphin matches the given computer output, the game should start. When the dolphin matches the output again, the game should continue and show the dolphins the next steps.

"So it's a computer-aided instruction (CAI) program that's needed," says Lilly—"a multiple subset program for the various games."

Adds Kert, "It would also be nice if some people could come out with games for human players that involve dolphins and whales and creative ways to interact with them, either through sounds or situations, analogous to space games. People would then become more conscious of cetaceans.

Kert, who was in the aerospace industry for twelve years before he joined the Human/Dolphin Foundation, had to learn some new computer languages to make the transition to working with cetaceans. Most scientists, he says, use Fortran, but assembly language is used on the Janus system. And, of course, the Janus language also had to be invented.

Kert says he got a lot of help from the other programmers on the staff: Dennis Kesner and Larry Marchman, who worked extensively on graphics analysis, and David Kusek, John Gard, and John James.

"All the software, all the computer programs, is a beautiful analogy for our own thinking and a powerful analogy that more and more people are beginning to understand," says Lilly. "My brain is my computer, my biocomputer; and myself and what I say and think are all software, generated by my brain."

The concept of a biocomputer being the hardware for human and dolphin software is extremely useful, according to Lilly. If we can tap into that intelligence, we might solve some of the problems of the human race ourselves. ◨

E.O.E.

EXEC SOFTSEL

from page 6

nearly a full-service distributor of software, now handling approximately fifty lines of software serving Atari, Pet, and TRS-80 as well as Apple, although Apple remains the bulk of their business.

In Softsel They Trust—with Good Reason. The method by which new product lines are added reflects the care with which Leff and Wagman treat their business. They refuse to add any line they cannot conscientiously recommend to their dealers. Because they've been so adamant on this point in the past, and because the dealers have come to depend on Softsel as a source of reliable information about new software in the marketplace, this policy has placed additional burdens on the now successful company.

Both Leff and Wagman spend substantial portions of their day evaluating new product lines. No line is accepted unless there's a consensus that they can recommend the products.

Leff sees Softsel almost as more of a consultant than a distributor. "We're a service company. It's harder to distribute software than hardware, because there's always so much new product being developed. It takes a 120 percent commitment on all our parts to keep current with the market. We're looking for software that makes a contribution."

Wagman points out another facet of the problem. "We've steered the retailers right so many times, they've come to trust us. We could probably get away with sending them something not as good. This means we have to look doubly hard at new product now, so we don't soil our reputation by going after a fast buck."

Leff adds, "The bigger we get, the more careful we have to be."

Being careful extends not just to product, but also to hiring. When the company decided to add a sales and marketing executive, they searched for weeks. Among the sources they contacted for advice was David Blumstein, a fourteen-year sales veteran who had worked with Leff at Informatics.

As the qualifications and responsibilities of this new addition crystallized, it became apparent that they had just about defined Blumstein's experience and abilities. Today, he's executive vice-president of sales and marketing, honchoing a staff of seven.

The Bigger They Get, the Harder They Try. Leff insists that he'd rather do a job himself, even if it gets only 75 percent done, than turn it over to unqualified personnel. This is in concert with his insistence that bigness in Softsel's case will never result in deterioration of service.

"We're not going to get too big. We're going to get better."

Wagman ratifies that thought by pointing to Softsel's improved discount policies for their customers. "When we were just starting out, there was no way we could offer the kinds of margins to our dealers that we do now. It's our volume that enables us to share our success with the dealers."

The effect that Softsel has on the end user is indirect but dynamic. Softsel's customers are able to order a wider variety of software and get it delivered fast. Leff tells the story of a new dealer who ordered three hundred seventeen different items from Softsel. "Other than orders for product that has not yet been released, we had to back order only four of them."

What this means is that dealers can spread their software inventory dollars over more product, rather than stocking only fast or steady movers. As the product sells off, they can get rapid replacement from Softsel. The result is more software profits for the stores and better software selection for the customers.

Customers' Guidance Repaid. Blumstein points out that

Softsel moves in the directions pointed out by their customers. Many of the lines they've picked up were at the urging of their customers. At present, Wagman is putting the finishing touches on a program that will give each dealer a quarterly inventory analysis and profit report on all product he ordered from Softsel.

Blumstein is architecting a pilot program whereby Softsel representatives will make store visits in southern California to help stock shelves and keep inventory current. If the pilot is successful, the concept may be exported to other areas of the country.

It isn't only dealers who benefit from the expertise collected at Softsel by Leff and Wagman. On occasion, they'll take a new software publisher under their wing by striking an exclusive distribution deal.

While profit is clearly a motive, such deals aren't being struck for the venal purpose of controlling the publisher or excluding other meritorious distributors.

Rather, Wagman's found that "oftentimes we run across companies that aren't too sure whether they want to be authors or publishers." During Softsel's exclusive distribution agreement, such companies get the opportunity to assess their stance correctly vis a vis the marketplace while remaining viable economically.

Because Softsel is one of the companies that get early previews of what's coming down the pipeline, its execs are in a better position than most to judge where the software market may be going.

Leff sees software changes as more "evolutionary than revolutionary," although he cites *Retro Ball* from Sierra Software as a revolutionary product because of its hardware piece that speeds and smooths Apple's animated graphics. For the most part, he sees the advances coming in programming technique rather than content and suspects that arcade games imbedded within adventure games may represent the next wave of popular software.

Future Means More Applications and Smarter Apples. He sees the market becoming applications-oriented: "For every new personal computer owner who masters Basic, there's a dozen who don't have the drive to learn programming languages."

Blumstein feels that the software is still following where the hardware leads, although he foresees such developments as microcomputerized new homes in the near future that will open new horizons both to home buyers and to software developers.

Wagman concurs with Blumstein, pointing out that while it was possible to program an excellent chess game years ago—"After all, the game itself hasn't changed in hundreds of years"—it took more powerful hardware to enable the software to reach its potential.

In that context, he looks for hardware advances to make possible extraordinary leaps in the area of artificial intelligence and man-machine interfaces.

"We already know how to program things that the hardware's not capable of handling. We can write artificial intelligence programs that contain a million branches from one word; the hardware just can't cope with that requirement."

Where will the future find Softsel? Wagman thinks the future is clear for the next three to five years. "We'll be supporting the traditional retailer in the best manner we can." Leff gives a different focus to the future: "Where there's a need felt by the consumer and a desire on the part of the retailer to service that need, Softsel will be there."

The Ensuing Adventures of the Original Owner. Softsel's daily payroll today exceeds the price that Leff paid for the company less than two years ago. So what about that adventure programmer who started it all? Has he faded into the same blessed obscurity that cloaks the seller of John Henry, the two-million-dollar horse, for eleven hundred dollars?

Not exactly. Today he's ensconced in Coarsegold, California, where he directs one of the largest purveyors of Apple software. His name? Ken Williams. ■

FIND THE MISSING ALKEMSTONE™
AND CHANGE YOUR DREAMS INTO REALITY

Legend says the ALKEMSTONE will grant its owner anything he desires.
We at Level-10 have been searching for the Alkemstone but we need your help.
All of the clues are contained in the Level-10 Alkemstone game.
You decipher them and discover where the stone is hiding, and we will reward you with $5,000.
And here's your first clue: see your nearest Apple dealer.

LEVEL-10™

A DIVISION OF DAKIN5 CORPORATION
P.O. Box 21187 • Denver, CO 80221 • (303) 426-6090

Level-10 is a trademark of Dakin5 Corp.
Apple is a trademark of Apple Computer Inc.

Alkemstone is a trademark of Level-10.
This offer void where prohibited by law.

VENTURES WITH VISICALC

BY CRAIG STINSON

This month's column starts with a bit of bad news. We have received a letter from M. R. Yeargin, of Fort Madison, Iowa, reporting a serious bug in the DOS 3.3 version of *VisiCalc*. The problem has to do with the storage of DIF files. Here's an excerpt from Mr. Yeargin's letter:

> After spending many hours in file design, format design, and data entry we began using the new *VisiCalc* software. It is at this point we began experiencing problems with lost files and data being over-written. I spent several hours working on rebuilding the catalog and VTOC trying to recover as many files as I could. I then called Personal Software; the representative I talked with insisted that I had hardware problems. After several minutes of discussion, and my insistence that I did not have hardware problems, [Personal's] representative finally admitted that some problems did exist. The nature of the problems was that you could not store more than seven (7) DIF files on a disk. If you stored more than seven (7) files you would experience the problem I was describing.

Upon receiving this letter, we did a little experimentation and managed to crash a disk catalog with five large DIF files.

Personal has assured us that the problem has been diagnosed and solved, and that by the time this issue of *Softalk* is printed they will have begun shipping corrected product. Those who had the thirteen-sector version of *VisiCalc* and who have bought the upgrade to DOS 3.3 will be entitled to a patch at no charge. Those whose first *VisiCalc* was a flawed sixteen-sector model, and who have sent in warranty cards, will be notified by mail of the problem and given the privilege of paying an additional $15 for the corrected item.

In the meantime, it's user beware. If your application requires DIF files, be cautious about how many you put on a disk, and back them up. If you're buying a new *VisiCalc,* make sure the product arrived on your dealer's shelf sometime after mid-September.

Bad news aside now, we turn to reviews of two more *VisiCalc* adjuncts—the *Real Estate Templates,* from Apple's *Special Delivery* catalog, and a worksheet manipulating utility called *VU#3,* from Progressive Software.

The *Real Estate Templates,* written by J. Michael Carlisle, come on two disks in Muffinable DOS 3.2. They were intended to run with thirteen-sector *VisiCalc,* although if you have 3.3 and a language system (or some other source of an additional 16K), they'll do just fine in 3.3 as well. Without a 64K Apple, you won't be able to use about half of the templates because they're too large and complex to share quarters with the more RAM-consuming version of *VisiCalc.*

The templates were created by a real estate professional, primarily for others in the business or for investors in need of tools for evaluating and managing income property. They do not, however, require any computer sophistication or familiarity with *VisiCalc.* All are sumptuously prompted and documented.

The models all use an over-and-under split-screen format. The top part of the screen is used for instructions and data input; calculated results are displayed below.

With one exception, the display areas all fit into eight nine-unit columns so they can be printed on an eighty-column printer. The formats are designed to produce neat printed reports, complete with borders, centered heads, and orderly tabulation.

Areas to the right of the eight-column display zone are reserved for numbers and formulas essential to the calculations. These values are provided by the author.

The templates cover a range of complexity from a simple monthly amortization calculator to an investment analysis worksheet that allows for three mortgages and fourteen income categories. Here's what you get:

On the monthly amortization calculator, you input five of six variables: purchase price, percent down payment, annual percentage rate, term of loan, monthly payment, and balloon payment. Hit an exclamation point and the program calculates the unspecified variable.

A monthly amortization schedule template requires input of purchase price, down payment, annual percentage rate, term, and balloon payment; it then calculates a detailed schedule for the first thirty-six months of the loan. For each month, the schedule lists principal balance, monthly principal payment, monthly interest payment, and totals of principal and interest paid to date.

The annual amortization schedule produces the same information as the monthly schedule but breaks it down on a yearly basis for the first thirty-one years of a loan. The template assumes the loan to be amortized on a monthly basis and prints the values obtained at the end of each calendar or fiscal year. You specify the month in which the loan is made,

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.

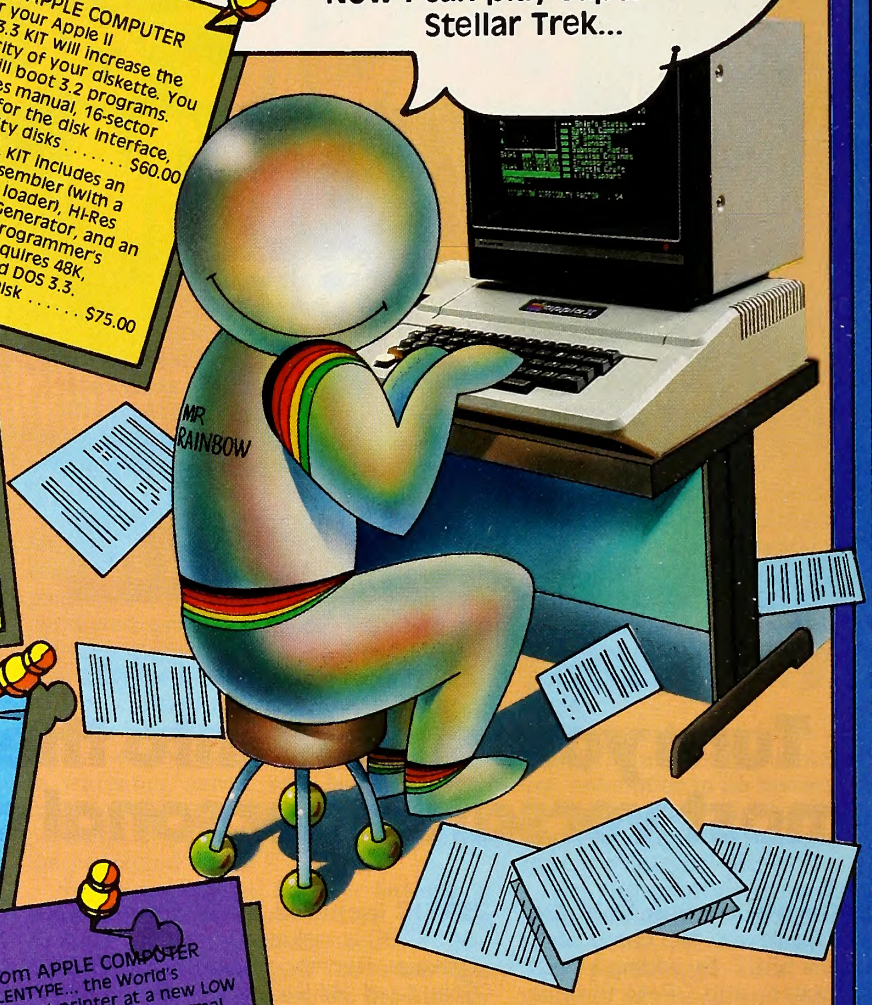
Complete information? It's at your dealer's now. Or, we'll send it to you and include a dealer list. Write us. Call us.

MICROSOFT
CONSUMER PRODUCTS

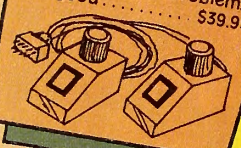
Microsoft Consumer Products, 400 108th Ave. N.E.,
Bellevue, WA 98004 (206) 454-1315

I've just finished the 4th edition of our catalog, featuring A RAINBOW OF APPLICATIONS for your Apple* Computer...

Now I can play Super Stellar Trek...



PRO-PADDLE
...for your Apple II
...at last ...a compact, heavy-duty paddle featuring sturdy metal construction, long-life switches with large buttons and tactile feedback, high accuracy paddle movement, shielded coaxial cables, and a molded plug.
End your paddle problems for good $39.95



From APPLE COMPUTER
...for your Apple II
DOS 3.3 KIT will increase the capacity of your diskette. You can still boot 3.2 programs. Includes manual, 16-sector PROMs for the disk interface, and utility disks $60.00
DOS TOOL KIT includes an Editor/Assembler (with a relocating loader), Hi-Res Character Generator, and an Applesoft Programmer's Assistant. Requires 48K, Applesoft and DOS 3.3.
Manuals and Disk $75.00

SUPER STELLAR TREK
...for your Apple II and Apple /// (emulation mode)
A dramatic successor to the already popular "Stellar Trek." Tom Burlew has created Super Stellar Trek as a high-speed, Hi-Res, Real-time action game that will challenge even the most skilled starship commander. In color. Complete manual.
DOS 3.2/3.3
48K Applesoft . . . . . Disk $39.95

STATISTICS with DAISY
...for your Apple II
...excels on user convenience and offers a full range of statistical capabilities; basic stats; correlations; hypothesis testing; multiple regression (6 procedures); non-parametric stats; data and time series transforms; and analysis of variance.
DOS 3.3.
48K Applesoft . . . Disk $79.95

From APPLE COMPUTER
SILENTYPE... the World's quietest printer at a new LOW PRICE! This compact thermal graphics printer receives both its power and intelligence from your Apple. Prints upper and lower case at up to 40 characters per second, 80 characters per line and Hi-Res graphics at 60 dots per inch. Comes with interface and cable. 6 lbs.
Apple II or Apple /// . . . $395.00

Our catalog contains many more products for your Apple computer. Please request a FREE catalog with any purchase. If you only want a copy of our catalog at this time, please send $2.00 for shipping. Thank you.

RAINBOW COMPUTING INC.

apple computer inc.

*Apple is the registered trademark of Apple Computer Inc.

Open Tues - Fri

Mail Order Dept. No. ST10 • 19517 Business Center Dr. • Northridge, CA 91324 • Phone orders only (Need Mastercard or Visa)
U.S.A. (except Calif.) (800) 423-5441 • Calif. and Foreign (213) 349-0300 • For information or technical questions (213) 349-5560
Add $2.50 U.S./$10.00 Foreign for shipping • Calif. residents add 6% sales tax.

relative to the beginning of the fiscal year. For example, if your fiscal year starts in July and the date of the note is April 1, you enter 10 for starting month and the template will pro-rate the first year.

With the mortgage loan analysis template, you can enter specifications for up to five mortgages and get a portfolio analysis for any point in time falling within the terms of the five mortgages. The acquisition dates can be after the dates the mortgages are made, allowing for the case where an existing mortgage is assumed. The summary report tabulates such things as the value of the debt at the analysis date, the amount of principal and interest paid since acquisition and during the twelve months prior to the analysis date, and the mortgage constant. With a flick of the finger and a press of the exclamation point, you can get a preview of your financial position at any time during the terms of the mortgages.

A comparative depreciation schedule template displays depreciated values according to straight-line, declining-balance, and sum-of-the-years-digits methods. Figures for all three methods are shown side by side in a yearly table that shows annual depreciation and remaining value for up to thirty years. You specify the starting and salvage values, the depreciable life of the property, and the percentage rate for the declining-balance approach. The declining-balance schedule automatically converts to straight line at the point when the annual depreciation calculated by the straight-line method becomes larger than that calculated by declining balance.

The *Real Estate Templates* also include a very thorough personal finance statement template that yields a four-page printed document in a format suitable for submission to a lending institution. The main body of the statement is broken down into assets, liabilities and net worth, sources of income, personal information, and contingent liabilities. Supplementary schedules cover banking relations; accounts, loans, and notes receivable; life insurance; stocks and securities; real estate; retail credit references; and margin accounts with brokers. This is the one template that requires a larger-than-eighty-column printer. The worksheet uses nine columns of nine units' width, so it barely exceeds eighty columns. Like all the other templates, naturally, it can be modified to suit individual needs.

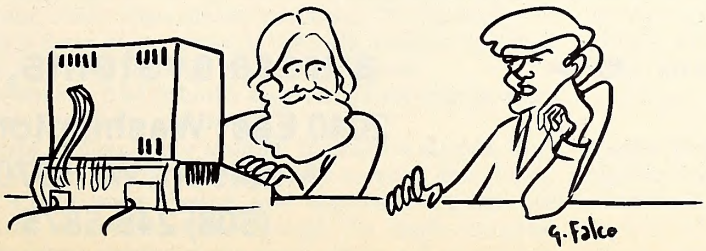
Finally, the package includes three templates for income property investment analysis. The three are identical except for their methods of calculating depreciation. These are the largest templates of the set—so large that they leave no room in memory for instructions. A good chunk of the manual, however, is devoted to their elucidation, and there's also a filled-out sample to show you the way.

The income property templates are intended to cover a single property with up to fourteen tenants and three mortgages. The fourteen tenants could be treated as fourteen income categories, if you want to lump together tenants with the same rent. You enter all pertinent data for mortgages, income, and expenses—including anticipated vacancy rates and rates of inflation for both rent and expenses—and specify an analysis date. The template provides a detailed investment analysis broken down as follows: cash flow analysis, income schedule, expense schedule, debt structure, income approach valuation of the property, cost approach valuation of the property, tax depreciation, federal income tax computation, and projected equity position.

As business software goes, the *VisiCalc Real Estate Templates* are a refreshing novelty: because they're unprotected, you can customize them to your own needs. You can modify the labels and the print styles, and—if you have the extra memory provided by a language system—you can observe the way the templates are made and extend them beyond their present limits. Also refreshing is the price—an underwhelming $65.

VU#3, from Progressive Software (Blue Bell, PA), is a tool for reorganizing *VisiCalc* worksheets. The program manipulates text files created by the print-to-disk (or print-to-file, in *VisiCalc* 3.3) command. *VU#3* will allow you to translate any single coordinate, row, column, or combination thereof, from one location to another, presumably for the sake of consolidating separate *VisiCalc* applications. It will also allow you to move data in either direction between *VisiCalc /PD* files and text files created by either Applesoft or Integer Basic.

The program, by Marc Goldfarb, comes on a two-sided disk—one side for each DOS—and retails for $89.95. ❑■



"I wonder who will be the first programmer to leave his brain to science."

A QUANTUM JUMP...

in Apple II Business Software!

The EXECUTIVE SECRETARY™ WORD PROCESSOR

THE DOCUMENT EDITOR

Keeps up with professional typing speeds.
Allows user-defined abbreviations.
Gives 40 or 80 character modes user-selectable.
Uses the real shift key.
Supports file merge and unmerge.
Has global search and replace.
Has block operatons: move, transfer, and delete.
Has character/word/line: insert/replace/delete.
Allows embedded commands to the printer.

THE INTEGRATED CARD FILE

Allows multiple card files per disk.
Allows the user to define size and content of records.
Generates new subset card files based on search or sort criteria for an existing file.
Incorporates separate high-speed label printer program.
incorporates multiple line report printer for card files.
Allows totals and subtotals during report printing.

THE ELECTRONIC MAIL SYSTEM

Is menu-driven
Supports multiple document queuing.
Is fully automatic with the D. C. Hayes modem.

THE DOCUMENT PRINTER

Integrates files from The Data Factory , Visicalc , and other sources.
Accepts keyboard input at print time.
Supports all major printers, including Centronics 737 and IBM ET series.
Allows conditional and relational commands to control the printing of information based upon the contents of a database.
Prints page headers of arbitrary complexity.
Prints page numbers whereever you want them.
Supports file chaining and nesting.
Supports multi-level outline indenting.
Has left- and right-justified tab stops.
Gives full control of all margins, dynamic text reformatting centering, and justification.
Accesses the CCS clock to time-stamp documents.

GENERAL

Editing, printing, form letters, mail merge, database access and electronic mail all in one package at one price.
Interfaces with The Data Factory™ and Visicalc™.
Has user-selectable configuration of printer, slots, drives, and 40/80 column edit modes.
Is fully menu-driven.

The EXECUTIVE SECRETARY: **PRICE: $250**

Everything you always wanted to do* with Visicalc™, (but thought you couldn't).

VERSACALC!™

★ **SORT** any number of rows; labels, values, formulas. You select the extent of the sort.

★ **CONDITIONAL TESTING** to several levels.

★ **PRINT OUT THE LIST OF COMMANDS**

★ **AUTO BATCH UPDATE** e.g. running year-to-date forms.

★ **MENU—DRIVEN MODULES** for your own application programs.

VERSACALC runs within Visicalc.

VERSACALC: **PRICE: $100**

★ **AUTO-CATALOG** from within Visicalc.

★ **AUTO SCREEN FORMAT** saves hundreds of keystrokes

YES! We said, "SORT VISICALC!"

Now in 16 Sector!

See your Dealer or contact us ➡

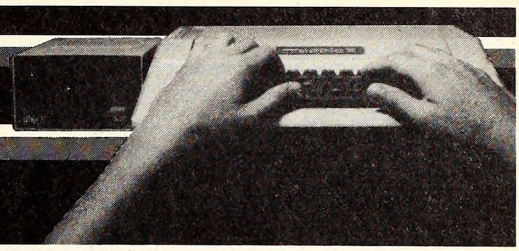
aurora systems, inc.

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

Also available from aurora systems:
Hebrew II Quicktrace
Omniscan Laser Video Disk Interface
The Rental Manager

MARKETALK

Reviews



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

Bez Man. By John Besnard. Here's the most entertaining program yet of the eat dots and gobble ephemeral folk genre.

Three things set this offering apart from the others. The Bez Man moves faster than the various villains trying to catch him, which means that you're only limited by your dexterity and wit. There are at least three different mazes—one for the first two levels of difficulty, one for the second two levels of difficulty, and one that lasts at least through level twelve.

Finally, everything moves faster at each succeeding level until things are nearly a blur at the twelfth level. If things get much faster at the higher levels, you might as well mail in your game—it'll be too fast for you to see what's going on or to react to it.

The game starts at a somnolent pace with only one ghost to eat and progresses to three beasties moving at incredible speeds on the upper levels.

Keyboard control uses the i-j-k-m diamond, which makes the game awkward to master and calls forth various styles in the different players. But whether you're a one-handed or a two-handed *Bez Man* player, one thing's for sure: if you liked *Snoggle* or *Gobbler*, you'll love this one. ART

Bez Man, by John Besnard, Bez, Irvine, CA. $22.95.

Epoch. By Larry Miller. The second post-Nasir arcade game from Sirius Software, *Epoch* is not a complicated game. It doesn't require lightning fast responses or quick ducking. Nor has it a long-range goal, other than scoring higher and higher, or a complex purpose. But it is beautiful.

In *Epoch*, your beat is space, raw and vast. You captain a ship out to destroy enemy fleets and bases. If there are other ships on your side, you never see them. Your assistance comes in two packages: friendly bases, at which you can refuel and rearm, and time warps.

There are no bounds to your movement in *Epoch*. Space is all around you, and the enemy ships, which often seem to fly in schools of a type, are everywhere. They are of many colors and several shapes that you'll learn to recognize from the time they are so far away as to appear as little more than dots. So, too, with bases and time warps.

Each type of ship has a vulnerable area that you must hit to destroy it. Enemy bases must be hit precisely in the center. When hit, they explode; the debris is as dangerous as enemy fire—rather than doing recordable damage to your ship, hits on you or collisions with debris simply deplete your fuel supply by varying amounts. All enemy facilities fire at you.

You must maintain your fuel and ammunition by visiting friendly bases; this requires finding them—a fuel-consuming task in itself—and flying through a small entry port.

Time is your other expendable resource. To replenish it, you must find a time warp, enter it through a small opening, and maintain a course through an energy tunnel.

As long as you can find friendly bases and time warps when you need them, the game continues.

Sound effects in *Epoch* are unique. Explosion sounds vary with the type of ship hit and are relatively normal. But refueling, entering a time warp, losing, and beginning a game are accompanied by two-voice music through the Apple speaker, simulating the organ strains of the movie *2001: A Space Odyssey*. It's very effective.

Besides steering and shooting, you control the speed of your ship, and the sense of space flight becomes eerily convincing. Game paddles or, preferably, a joystick are your rudder, accelerator and decelerator, and trigger. You can choose among five degrees of steering response, and keyboard strokes toggle the direction response of the controller, so you're never stuck with having your ship do exactly the opposite of what seems the natural response to your input.

High score is your goal. On the surface, in view of the sophistication of games today, that seems an inadequate one. But the terrific sense of free motion and sailing through space combined with the attractiveness of the graphics make this game a real winner, one that you'll return to again and again, if only to experience its universe.

The graphics are not perfect. While the sense of depth overall is excellent, the ships and bases, when you close on them, prove to be flat. It doesn't take much from the overall effect, but having these also three-dimensional would have been ideal. Sirius promises that is coming. MCT

Epoch by Larry Miller, Sirius Software, Sacramento, CA. Either DOS. $34.95.

Individual Tax Plan. This package would be a pleasure to use. The documentation is among the best for microcomputer software. Introduction to program use is through step-by-step input of two different cases. You are told what should be input, what it will do, and what the screen should look like. Even a disagreement between the text and the software is a simple matter from which to recover and continue.

Individual Tax Plan was designed for computing federal taxes. It will use appropriate tax tables or rate schedules, income averaging, and maximum tax on earned income and then select the lowest tax due. It will calculate up to five alternate cases at once to establish the minimum tax. It can handle charitable contributions, medical expenses, capital loss limitations, capital gain deductions, and ten-year averaging for lump-sum distributions.

Individual Tax Plan was created primarily for the professional tax preparer who is aware of the rocks and shoals of current federal tax law. It is not form oriented and will not prepare submittable forms. It is more involved with the big picture and leaves the detail to the user. As an example, "dividends after exclusion" is a requested input. You are expected to know how much to exclude. Having found the minimum tax approach, either the professional or the individual could then finish the job with a program that prepares submittable tax forms.

The program runs on a 48K Apple with at least one floppy drive with either DOS 3.3 or Pascal. Back-up copies of the program and data disks are easily made with Apple's Copy program. Compatible printers listed are Anadex, Epson, Spinwriter, and Okidata, but it ran perfectly on a Xymec as well. If substantial changes in the tax law are made during the year, new program disks will be available for $50. Disks for the 1982 return will be available to current owners at 45 percent of the retail price. IH

Individual Tax Plan, Aardvark Software, Milwaukee, WI. $250.

Castle Wolfenstein. By Silas Warner. There is no game on the market like *Castle Wolfenstein*. Silas Warner has succeeded in creating a truly original game; hopefully, the idea will catch on and what might be called *arcade-adventure* will become a genre, because it's good.

This is a hi-res, talking, role-playing adventure requiring

the skill for real-time quick response needed for an arcade game.

You've been taken prisoner by the SS in Nazi Germany during World War II. They've sent you to Wolfenstein Castle for interrogation before they do you in. Another prisoner gives you a gun stolen from your captors and urges you to try to escape. He also tells you of a set of war plans somewhere in the castle. Escaping with the plans would be a great coup for the Allied forces.

Your job is to escape. Ideally, you'll escape with the plans. Either way, you win the game.

The castle is made up of several levels of hi-res rooms of various configurations. A block of one to three rooms appears on the screen at a time. Many rooms contain locked chests in which may be provisions of various types, arms, or nothing. Almost every room is guarded by one or more regular German army guards. Occasionally, a room is guarded by a member of the SS. Most rooms open into other rooms, but some are set off by locked doors.

Guards, SS, and you are tiny colorful hi-res people who move in an animated goose step. Detail goes as far as having a swastika on the German uniform and an SS on the bulletproof vests.

Uniforms and bulletproof vests are among the provisions you can find in the castle. Wearing the former may fool some of the guards; the benefits of wearing the latter are obvious.

Guards are required to remain in the area they guard; SS men, however, move freely around the castle. Once an SS member discovers you, he'll pursue you until you kill him or he catches you. Since the SS (and only the SS, except you) wear bulletproof vests, you can't count on shooting them. A well-placed grenade will usually do the job, however. Not so well-placed, the grenade explosion can take you with it or just burn off your vest and uniform.

Adding to the atmosphere of *Castle Wolfenstein* is an extensive array of sound effects, the most pervasive of which is the marching of the sentinal soldiers. An annoying, abrasive sound occurs whenever you run into anything—wall, soldier, chest; yet it apparently has no effect except to discombobulate the player.

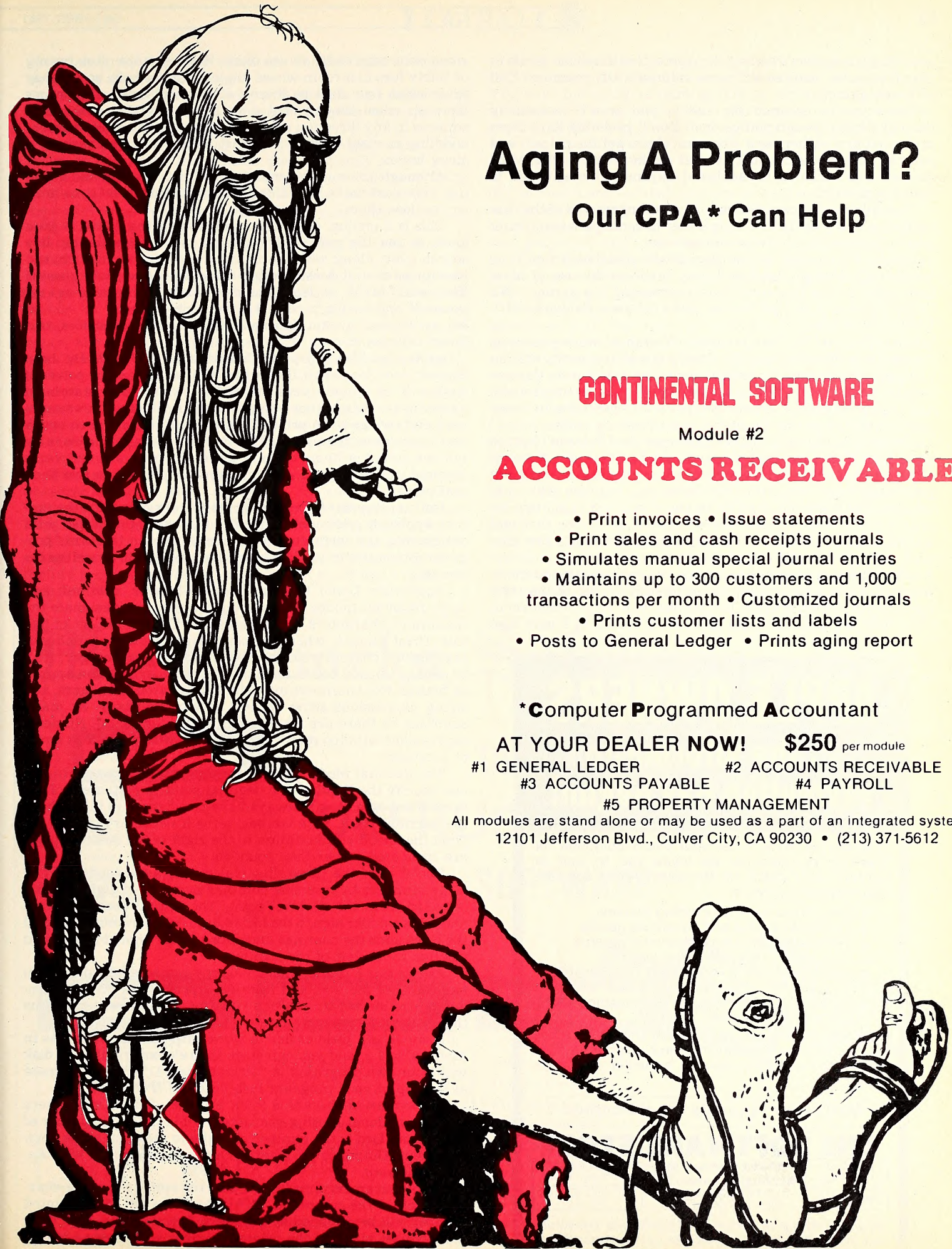
The piece de resistance in the sound area is that the Germans talk; and, of course, they talk in German. The instructions contain a brief dictionary translating their comments to English. You won't need the dictionary to get the gist, however.

It takes a while to get going in this game. You'll have to restart several times before your timing is right and you pick up various strategies. Unfortunately, each time you start over, you're subjected to a long load and a repeat of a lengthy discourse by the prisoner who gives you the gun. The one time you're spared the discourse—but not the load—is when you're resuming a game you saved.

If that sounds like you can avoid the opening if you just save often, think again. Silas Warner believes saving to avoid replay in case you're killed is not quite kosher; therefore, saving ends the play, rebooting automatically picks up the saved game, and, when you are killed, the saved game goes with you. Thus, the save feature on *Castle Wolfenstein* serves purely to enable you to quit because you have something else to do. You'll have to start over if you get killed anyway.

While this is very frustrating at first, the castle is small enough that getting through it in one sitting is indeed possible. Once you get into it, the good features easily outweigh this early frustration.

The most serious difficulty with this arrangement is its not providing for more than one game to be played (in this respect, this is a common inconvenience). If you've achieved a promotion or two and it's your child's turn to play, the child must either start right out with a harder game or you must give up your rank. In addition, if you have a game on hold and



Aging A Problem?

Our **CPA*** Can Help

CONTINENTAL SOFTWARE

Module #2

ACCOUNTS RECEIVABLE

• Print invoices • Issue statements
• Print sales and cash receipts journals
• Simulates manual special journal entries
• Maintains up to 300 customers and 1,000 transactions per month • Customized journals
• Prints customer lists and labels
• Posts to General Ledger • Prints aging report

*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

Hardware requirements: 2 disk drives, DOS 3.3, APPLE][48K, 132-col. printer and Applesoft in ROM.

**APPLE][is a registered trade mark of Apple Computer Inc.

now it's your spouse's turn at the Apple, if said spouse wants to play this game, your saved game automatically resumes. End of saved game.

Once you've escaped the castle, you aren't necessarily through—even though you've won. You'll probably have been promoted to a higher rank (for sure, if you got the plans), and you can go back in and do it all over again—harder. There are at least six ranks to be achieved beyond private, which is where you start.

In addition, you can generate new arrangements of the castle rooms and guards, choosing either to retain whatever rank you've achieved or to return to private.

Castle Wolfenstein can be played with paddles, joystick, or keyboard. Although keyboard play involves the use of nineteen keys, it was preferred by all reviewing the game. MCT

Castle Wolfenstein by Silas Warner, Muse Software, Baltimore, MD. Either DOS. $29.95.

Mission Escape! By Jim Jacobson. You must work your way through numerous rooms of a closely guarded security station in hopes of reaching the hangar deck and freedom in this unusual arcade game. Tiny hi-res storm troopers, deadly robots, and highly explosive drones mingle with blockades to block you as you follow the directions from room to room.

Strategy is crucial in *Mission Escape!* And it doesn't hurt to have a good eye for angles; keyboard numbers one through eight stand for the eight angles of the compass as you direct your movement and aim. You have ten missiles (the only weapons that will kill a robot in one shot), ten rapid-fire repeater shells, and infinite lasers, as well as armor that will withstand ordinary bullets. Unfortunately, the robots fire missiles, so they can kill you in one shot, too.

The strategic planning helps you determine when to move where. Foes can hit you only from directly on one of the eight angles (you have the same limitation), and they cannot shoot after they have moved (you can). So if you're in a safe spot when your turn ends, you're okay. But there are often twenty or thirty foes in a room whose angles must be safe, so it's easy to overlook one. Just so it isn't all cut and dried, the drones blow up when they're shot and take everything within two squares in any direction with them. They have a nasty habit of cuddling in right next to you, just where they can pick up a stray bullet.

Although you and the enemy take turns moving and shooting, you must make your three moves within a brief time limit, or you lose them.

This is a unique, enjoyable, and challenging game. A good touch is that the game acknowledges and retains the top five scores ever, along with the name you input for your characters for games; it doesn't forget them when the power goes off. They start blank, so from the start you're competing against yourself and seeing a record of your progress. MCT

Mission Escape! by Jim Jacobson, CE Software (Des Moines, IA). Either DOS. $24.95.

Appleprint Using. By M. Condat. And: **Print II.** By John Hooper and Jim Leach. These two utilities extend the power of Applesoft to include versions of the Print Using statement available on many other computers. Print Using allows numbers and strings to be printed in specified formats or at specified locations on the screen. The most obvious application is for tabular output of financial data where the programmer wants decimal points aligned and all numbers printed to two decimal places.

Both *Appleprint Using* and *Print II* work alongside the normal Applesoft print command. You can use unformatted print statements and formatted ones side by side in the same program. Both utilities work in immediate execution as well as deferred.

Appleprint Using has five modes—integer, decimal, and three financial modes. In the integer mode, you designate the maximum characters to be displayed. The syntax looks like this: Print ######A, where the pound signs show the number of permissible characters and A is any expression—integer, real, or string. Unused positions to the left of the expression appear as blanks. Real numbers are printed truncated to integers, and string expressions show up as a row of asterisks—as many asterisks as there are pound signs in the command. Numeric expressions with too many digits are likewise printed as a row of asterisks.

The decimal mode is similar, except that the programmer can specify the position of a decimal point. A statement of the form Print ####.## will always have two figures to the right of the decimal point, with trailing zeros if necessary. Digits beyond the specified maximum to the right of the decimal point are truncated rather than rounded.

The first of the three financial modes adds readability to large numbers by inserting commas every third digit, moving leftward from the decimal point. The second does the same and adds a dollar sign to the immediate left of the number; and the third adds the commas and puts a dollar sign at a specified position to the left of the number.

Appleprint Using comes on a copy-protected 3.2 disk. You have to boot the disk and install the program's machine-language routine before you can run any of your own programs that include formatted printed statements.

Print II is a little easier to work with because it comes in Muffinable 3.2, and you can store the relevant routine on disk with your own programs. A "Print control-D Brun" statement at the top of your program installs *Print II* at $9A07.

The syntax of *Print II* is Print For [Lc] (Td), (Drd), where Lc is an optional leading character; Td is the total number of digits; and Drd is the number of digits to the right of the decimal point. Output to the right of Drd is truncated, as with *Appleprint Using.*

If you put an exclamation point in the optional leading-character field of the command, all blanks to the left of the printed number will be replaced by asterisks. If you enter #, $, %, or ", the leftmost blank of the printed number will be replaced by the character selected. A number that overflows the limits

specified in the Print For command will be replaced entirely by greater-than symbols.

Print For is intended to work with integers and reals; using it with string variables will not result in syntax errors but will produce some unpredictable output.

In addition to Print For, the *Print II* program also has a Print At command that allows you to specify x and y coordinates of printed output. This accomplishes the same thing as Htab and Vtab statements but is easier to work with. (\

Appleprint Using, by M. Condat. Malibu Microcomputing, Malibu, CA. DOS 3.2, 32K. $19.95.

Print II. Computer Systems Design, Rapid City, SD. DOS 3.3 or 3.2, 32K. $24.95.

Inferno: A Fantasy Adventure. Bold, attractive hi-res graphics introduce the *Inferno;* then the history of the adventure scrolls by in stylized hi-res text on—of all things—a hi-res scroll. When at last it's time to begin, the graphics are put away, and a text adventure begins. It doesn't matter. You'll soon be seeing the word-painted rooms and passages clearly.

A few elements are different about *Inferno* and, at first, a bit disconcerting to the veteran adventurer. When you use a phrase the game doesn't know, it gives you no hint as to what's mysterious about it. In fact, it may know both words of the two-word phrase, but not in combination, so you'll have to try harder.

Then, seldom will searching or doing things reveal any hint of secret passages. There are plenty, but you won't find them until you simply enter the initials for their directions. So you must try all eight directions—oops, ten—from every room.

There are a few odds-ruled events in *Inferno*—such as the bridge that's always collapsing into a lava river—but the save feature is quick, leaves you in the game, and doesn't require a separate disk, so you can use it often.

Most of these probability traps occur in one place, but there's a nasty orc that follows you throughout the adventure. You have the option of fighting him or running, and once in a while you get away. Even when you don't, you may occasionally win. Either way, encountering the orc costs you life points, and that's what you need to make it out of the inferno.

Mapping is important, but you needn't worry about mazes of indistinguishable rooms; the author doesn't like them. He did include a section that appears to be a maze; but it's wits and invention, not mapping, that will get you through it.

As you conquer more and more of the *Inferno,* the adventure gets more and more fascinating. Puzzles are unique and hard, but their solutions are ultimately logical—the kind that make you so deliciously aware of your mind's competence when you finally figure them out.

Winning earns you a hi-res reward, but you'll probably want to do it all over again to raise your adventurer rating by using fewer life points and fewer turns.

This adventure is a winner from a new entrant into the adventure pool. We're looking forward to more adventures from this inventive, enthusiastic group of writers. MCT

Inferno by The Software Emporium (Tulsa, OK). $29.95.

Crossword Magic. By Larry Sherman. Occasionally there appears a software package that defies normal categorization. Either it fits—sort of—into several categories, or it doesn't fit properly into any. *Crossword Magic* is such a package; but it's so good at its unique task that, if you enjoy words and word play, you would do well to give it a place in your software library.

If you happen to be an educator, you'll be remiss not to investigate the program.

The package is composed of two disks—a player disk and a maker disk. The latter is the magical sibling. With this, you

can make crossword puzzles with no sweat or pain at all; *Crossword Magic* does all the work. If you're thinking it never occurred to you to try to make a crossword puzzle, anyway—read on.

There's more to this ability than immediately meets the eye. *Crossword Magic* provides you with an instant grid in a size you choose. As you type in words, the maker disk fits them into the grid, crossing them with words already there. If there's no fit for a word you enter, the program tells you so but remembers the word. Then, when another entry falls in such a way as to allow the previously rejected word, the program will put it in.

The maker disk recognizes a place for a word, even if it must span across several other words. There is even a skill to requesting words; the better your planning, the more words the program will be able to fit in.

Because the word bank is blank when you begin, you can enter any kind of words you like. If your child is studying French, ten minutes with *Crossword Magic* will produce a crossword puzzle diagram that includes all your child's current French vocabulary words. (Unfortunately, if you don't know French, *Crossword Magic* doesn't think up the definitions for you.) You could also provide your child with a puzzle made up of words having to do with Apple computers. Writing the definitions will probably be a lot easier that way.

The final puzzle is not apt to be symmetrical, nor will every box have a crossing word to help out the solver. The end product is more in line with British-style puzzles than with the standard Dell variety. But these are increasing in popularity in this country, perhaps because of the double and triple definitions that often characterize this kind of puzzle. They're more fun to figure out—and more rewarding—than another name for an African ibis or an Indian arrow poison.

Once you've made a puzzle, complete with definitions, you can save it to the disk. Another ability from the same menu transfers the puzzle to the player disk, from which each member of your family can try to solve your masterpiece on the Apple.

If you're very proud of your puzzle, or if you're a teacher and want to give copies to your students, *Crossword Magic* will print out your puzzle on a Silentype or Epson printer. The printout—at least on the Silentype—is excellent.

You can see an example of a *Crossword Magic* devised puzzle printed on a Silentype and reduced for publication on the contest page of *Softalk* this month.

Words convey almost everything we wish to communicate; without words, we'd still be living in trees—or whatever it was our early ancestors frequented. *Crossword Magic* enables you to have fun learning about words—some of our most valuable commodities. MCT

Crossword Magic by Larry Sherman, L & S Computerware (Sunnyvale, CA). $89.95.

Falcons. By Eric Varsanyi and Thomas Ball. [*With apologies to reputable poets everywhere.*]

My Apple's givin' me an awful fright
Space critters spoilin' for a fight.
There was *Alien Rain* and *Typhoon,* too
Shootin' at me till I'm through;
Space Invaders with tank cars clippin'
Gamma Goblins with blood a drippin'
Space Eggs hatchin' fuzz ball villains—
This ain't no place for younger chilluns.
But the worst of all, Willy-nilly,
Is the *Falcons* game from Piccadilly.
Spacemen floating in the air
Change to birds to attack your lair.
Two sets of these is quite enough,
But what comes next is sterner stuff.
Little dots weave to and fro,
I fire madly as they go;
Each tiny dot grows on the screen
To become a *Falcon* flying machine.
Down they swoop, hell-bent for leather!
No time to question if or whether
You can survive their awful attack,
But if you do, there's no going back.

WHY THE MICROSOFT RAMCARD™ MAKES OUR SOFTCARD™ AN EVEN BETTER IDEA.

Memory — you never seem to have quite enough of it.

But if you're one of the thousands of Apple owners using the SoftCard, there's an economical new way to expand your memory dramatically.

16K ON A PLUG-IN CARD.

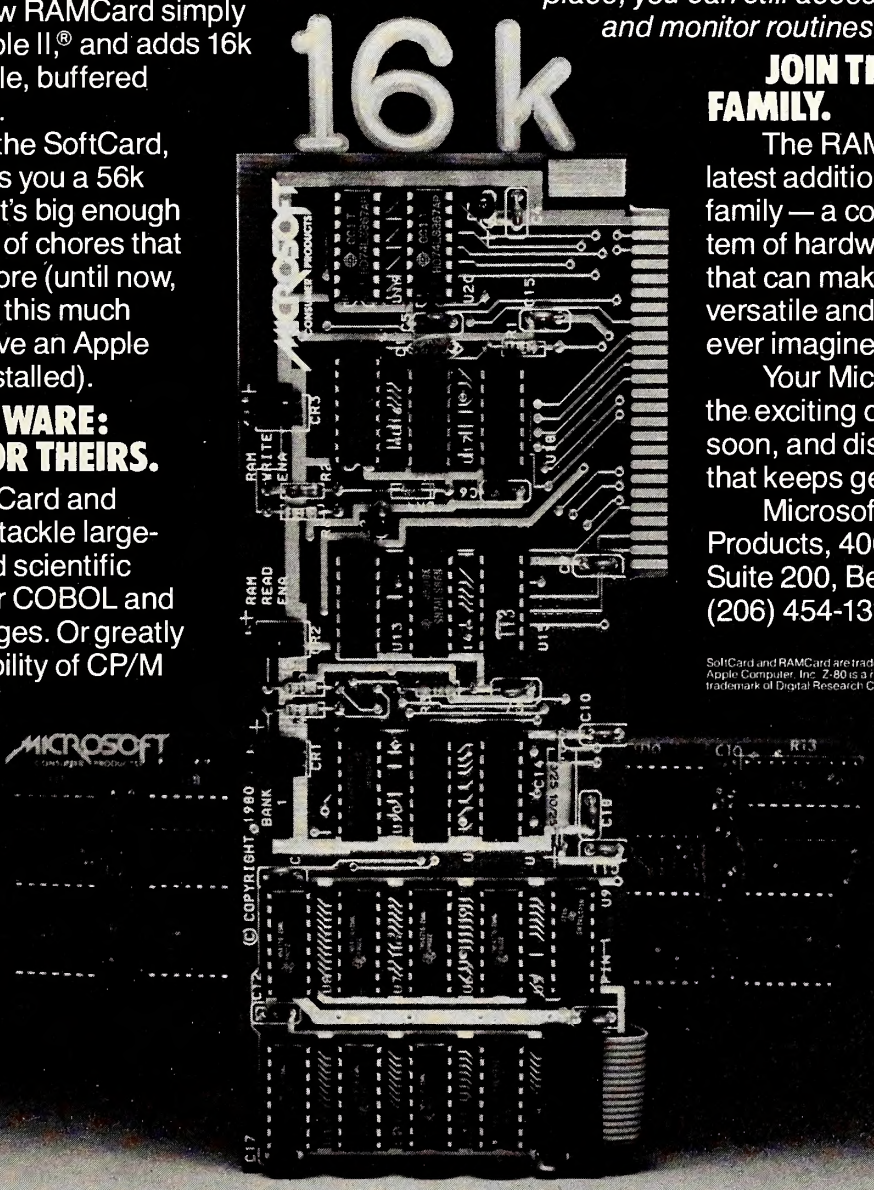
Microsoft's new RAMCard simply plugs into your Apple II,® and adds 16k bytes of dependable, buffered read/write storage.

Together with the SoftCard, the RAMCard gives you a 56k CP/M® system that's big enough to take on all kinds of chores that would never fit before (until now, the only way to get this much memory was to have an Apple Language Card installed).

GREAT SOFTWARE: YOURS, OURS, OR THEIRS.

With the RAMCard and SoftCard, you can tackle large-scale business and scientific computing with our COBOL and FORTRAN languages. Or greatly increase the capability of CP/M applications like the Peachtree Software accounting systems. VisiCalc™ and other Apple software packages can take advantage of RAMCard too.

And RAMCard gives you the extra capacity to develop advanced programs of your own, using the SoftCard and CP/M. *Even with the RAMCard in place, you can still access your ROM BASIC and monitor routines.*



JOIN THE SOFTCARD FAMILY.

The RAMCard is just the latest addition to the SoftCard family — a comprehensive system of hardware and software that can make your Apple more versatile and powerful than you ever imagined.

Your Microsoft dealer has all the exciting details. Visit him soon, and discover a great idea that keeps getting better.

Microsoft Consumer Products, 400 108th Ave. N.E., Suite 200, Bellevue, WA 98004. (206) 454-1315.

MICROSOFT

Three hundred points to kill those devils,
And these were just the easy levels.
A spaceship comes, no, I'm not daft,
And it's defended fore and aft
By more of the guys from the very first round;
Flyin' and divin' they abound.
I've got to say I've not a clue
If it's possible to shoot on through.
But this I know: If you can,
Then I'll concede you're a better man. ART

Falcons, by Eric Varsanyi and Thomas Ball, Piccadilly Software, Summit, NJ. $29.95.

Impressions

☐ **Shuffleboard.** By Howard De St. Germain, David Morock, and Don Hoffman, Innovative Design Software (Las Cruces, NM). You wonder how a computer can play so well, how an armless piece of hardware can put you to shame in a contest of delicate physical skill.

Shuffleboard can be played against the computer or another player. Better that you play with a compassionate friend than the cutthroat computer. The computer is amazingly skillful for its age and ruthless enough to knock any of your high scoring pucks off the target area.

Innovative, who also designed *Pool*, is good at creating graphics for colliding objects. *Shuffleboard* allows you to aim your puck and control the strength of your push, from a very delicate placement shot to a jarring jolt that will send your opponent's puck sailing off the court. You can give your puck a hardy shove and watch it ease up into a graceful slide that accurately reproduces the movement of a real game. Bank your puck off the side walls for those tricky shots; and if you're a good aim, you can bump your opponent's puck off the board while landing your disk to score five big points. The computer even cheers you on with the first four notes of Beethoven's Fifth Symphony when you make a good shot.

If you long to sit poolside on a Mediterranean cruise, lazing in the sun while you indulge in a game of shuffleboard, this game might save you a lot of money. So the sun doesn't come with the game. You can always go out in the back yard and sip tea when you're through playing with your computer.

The program provides options for both an easy and a difficult level. You can play *Tally All* or *Cutthroat*. Choosing the difficult level might mean waiting a long time to win a game.

The computer is a hot shuffleboard player—probably the result of a misspent youth. The game might not be as much fun as a Mediterranean cruise, but how else could you play shuffleboard in rainy weather? DOS 3.2 or 3.3. $29.95.

☐ **Pro Football: A Pointspread Prediction System. College Football: A Pointspread Prediction System. Win at the Races.** All three programs by Ken Perry, Systems Design Lab (Redondo Beach, CA). It's that time of year again. You can pit your Apple against the Las Vegas line or the rest of the office pool with the help of these programs.

Both football programs work the same way. As the season progresses, you enter won-lost and home team data for all the teams you're concerned with. Perry recommends that you collect six weeks' worth of data before asking the computer to predict pointspreads. Thereafter, you just name the contenders and specify the home team, and your Apple will do its calculations and give you a point spread.

Perry claims that his algorithm won more than 73 percent against the Las Vegas line in 1980, playing only upsets—games in which the program predicted one winner and Vegas the other.

In *Win at the Races,* you enter name of track, type of race—sprint or route—and data for each horse. The program has different algorithms for sprint and route races. For sprints, you enter number of days since the horse last raced, number of wins in the last ten starts, number of good races and failures in the last ten starts, number of speed points (defined in the documentation), total number of starts in the horse's career, total winnings, jockey rating, and the last two good speed ratings. For route races, the program looks at post position, jockey rating, total earnings, number of starts, number of wins in last ten starts, and last two good speed ratings. After all data is entered, the program retires for a moment and returns with its predictions. Either DOS. Football programs, $26.95 each; *Win at the Races,* $39.95.

☐ **Wall Street.** By Donald Brown, CE Software (Des Moines, IA). *Wall Street* lets you into the power world of high finance. The fate of the American financial system will be determined by your investment decisions. One to nine players can play this stock market game inspired partially by the noncomputerized game of High Finance. You can buy and sell stock, take out loans, license a secret information service, and scan the pages of a finance journal that keeps you up-to-date with the latest stock information.

You can buy stock in six industries: entertainment, oil, automotive, retail, appliance, and, of course, computer. If you're a computer fan, you know where you should invest. Investment possibilities in the computer industry include Apple, IBM, Tandy, DEC, and CE Software.

After designating the number of players (three to five are best), each player will be asked to select which information service he or she would like to license. The better the information the service provides, the more you have to pay. Messages from the service will be coded and appear randomly on the front page of the finance journal. Don't be looking off into space when your secret information flashes. It might appear for only a second, or it might be plastered as a headline on the front

NEW...FOR APPLE II[1]

FROM PROMETHEUS

VERSAcard

1. Serial Input/Output Interface
2. Parallel Output Interface
3. Precision Clock/Calendar
4. BSR Control

NEW 4 in 1 CARD

ALL ON ONE CARD

with true simultaneous operation . . .at a down-to-earth price!

CHECK THESE FEATURES: True simultaneous operation! VERSAcard's unique hardware selection circuitry makes your Apple "think" that separate cards are installed. This allows VERSAcard interface ports to be compatible with existing software such as APPLE PASCAL, Microsoft Softcard[2], and most other Apple software.

The Serial Input/Output port is RS-232C standard. Features include accurate crystal controlled baud rates, half/full duplex operation, and interrupt capability.

The versatile **Parallel Output Interface** is configured for the Centronics standard and is easily configured for other standards. Firmware is available to utilize the graphic features of popular printers such as the Epson.

Precision Clock/Calendar. All standard real-time clock/calendar functions. One second to 99 years time keeping. Battery back-up supplied.

BSR Control. Add an ultrasonic transducer or the Prometheus Direct Connect Option and VERSAcard provides you with BSR remote control. By utilizing the on-board real-time clock and the Prometheus Time and Event Schedular – you can remotely control various accessories in your home and office.

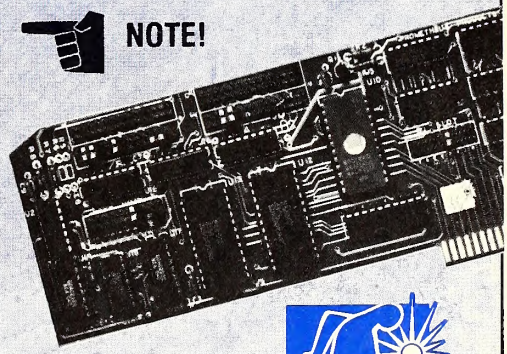
VERSAcard is shipped complete with SOFTWARE for each function contained in an industry standard EPROM. Versatile device drivers permit connection to virtually any standard peripheral. The Prometheus engineering staff is available to help you with any peripheral and special device interface problems. Call us, we'd like to talk with you.

VERSAcard – the only Apple peripheral card you're likely to ever need!

Order: PP-VCD-1 VERSAcard @ $249.00 each.

Compare Key Features	VERSAcard	CPS MultiFunction Card
Parallel, Serial and Realtime Clock Interfaces?	YES	YES
BSR Control Interface?	YES	NO
Compatible w/ Apple Pascal?	YES	NO. Special Pascal disk required.
Compatible with CP/M?	YES	NO. Special CP/M disk required.
Cable Included?	YES. One cable of choice free with each order – through 1981.	NO. Must be purchased separately.
Apple Modification?	NO	YES

NOTE!



Dealer inquiries invited.

All Prometheus boards are completely tested and burned-in prior to shipment. A one-year warranty covers parts and labor.

Registered Trade Marks: (1) Apple Computers, Inc. (2) Microsoft Consumer Products

PROMETHEUS

PROMETHEUS PRODUCTS INCORPORATED

42577 Fremont Blvd. • Fremont, CA 94538 • (415) 490-2370

page of the journal where opponents can take advantage of information you pay for. The code will tell you if a particular stock is going to drop or rise, and the day the event will occur. "In five days there is 20% probability that IBM stock will drop 50 percent." You'll have to decide what to do. Of course, the information is coded, but everybody knows the code. Usually the information applies only to one player's stock.

Each day the finance journal lists headlines announcing major events that affect the stock market. Inside are individual listings of stock price and gain or loss figures for all thirty companies, as well as individual financial positions (holdings, loans, and net worth) of each player. At the end of the week, players must pay fees for interest on loans, information services, and broker transactions. You can go broke in two weeks if you lease a high-priced information service that doesn't pay off.

The object is to become the first millionaire. Everybody starts with ten thousand dollars. A player can also win by remaining solvent while everybody else goes bankrupt. A vicious game.

So you want to be a millionaire. If you don't have time in real life, you can play financier at home and make a mint in your spare time. $24.95.

☐ **Pegasus II.** By Olaf Lubeck, On-Line Systems (Coarsegold, CA). Your space craft flies low over enemy terrain, dropping bombs on enemy structures. Meanwhile, the enemy fires rockets to bring you down.

The beauty of this game is in the variety of controls you have. You can fire missiles at objects directly in front of you or drop bombs on objects below. Hitting the space bar allows you to lag behind, giving you time to aim at objects ahead and making it easier to avoid enemy fire. Or you can stay forward in the thick of the fire where you score more points for your hits.

Pegasus even allows you to create your own terrain, which makes the game far more challenging. You can choose between beginning or intermediate levels for a meander tunnel through which you must maneuver while dodging moving bombs. If you hit the side of the tunnel, it's all over. Likewise, if you hit any cliffs, plateaus, or enemy bases, you'll be blown to smithereens. One blow isn't fatal: you've got five ships. But don't be too careless.

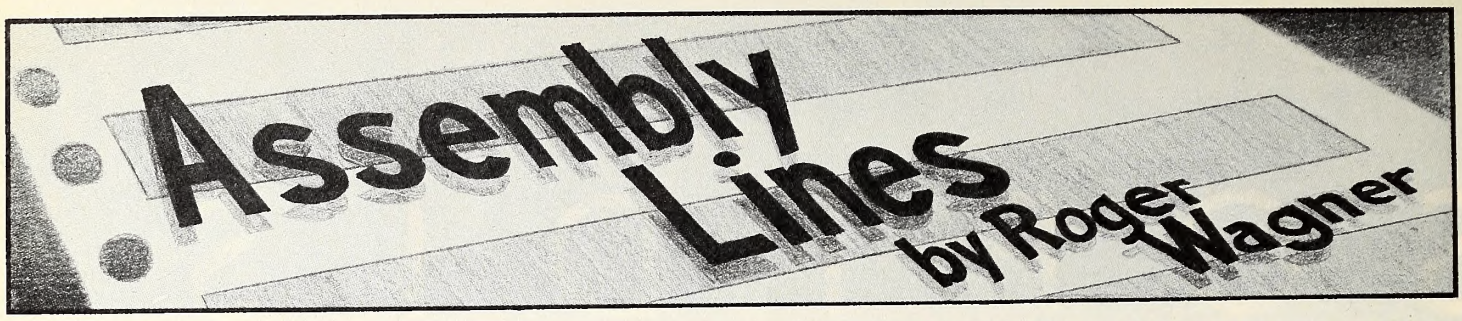
Scooting along over enemy terrain, you'll find three different types of enemy installations: watertowers, fuel tanks, and missiles. The ships' cruising altitude is controlled by a paddle. In the normal terrain it's possible to swoop down quickly after the first mountain peak and fire continuously at everything in front of you. But slow reflexes are likely to cause you to crash into a fuel tank, blowing up three of your ships. If you can't get down in time, you're better off whisking over the enemy structures and dropping bombs. The only drawback is that now you are a sitting duck for the missiles below.

If you master the quick-dive-and-fire approach, you will soon find yourself dealing with the *Pegasus* monsters, who will fire backward while flying away from you. You must dodge their extending lasers while you pick them off. Moving onto terrain again, you will be confronted by a constellation of stars that you cannot hit. If you're good, you can dodge stars while you drop bombs. Next you will run into a field of eagles. Now you have to shoot eagles, dodge missiles and drop bombs faster than you can think.

The eagles are followed by a hovering formation of saucers and—phew! If you make it through all that, you'll need a refueling. Your space station appears, and you have to dock and load up with fuel.

If you tire of the original terrain, you can opt to create your own terrain. A terrain generating program allows you to

GOTO 86



Everyone's Guide to Assembly Language, Part 13

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	

In an earlier issue, I discussed how to access the disk using the RWTS routine. There is another way in which the disk can be read that is more similar to the procedure used in Basic. The advantage of this system is that we need not be concerned about what track and sector we're using, since DOS will handle the files just as it does in a "normal" program. The disadvantage is that we must have the equivalent of Print and Input statements to use in our programs to send and receive the data. So, before going any further, let's digress to input/output routines.

Print Routines. I, personally, have two favorite ways of simulating the Print statement. The first has been described in earlier issues and looks like this:

```
                          1  ****************************
                          2  * DATA-TYPE PRINT ROUTINE *
                          3  * ************************
                          4  *
                          5  *
                          6    OBJ $300
                          7    ORG $300
                          8  *
                          9  COUT EQU $FDED
0300: A2  00              10  *
                          11 ENTRY LDX #$00
0302: BD  0E  03          12 LOOP LDA DATA,X
0305: F0  06              13    BEQ DONE
0307: 20  ED  FD          14    JSR COUT
030A: E8                  15    INX
030B: D0  F5              16    BNE LOOP
                          17 * (ALWAYS UP TO 255 CHRS)
                          18 *
030D: 60                  19 DONE RTS
                          20 *
030E: 84  C1  D4          21 DATA HEX 84
030F: C3                  22 ASC "CATALOG"
                          23 *
0316: 00                  24 EOF BRK
                          25 *
```

This type uses a defined data block to hold the ASCII values for the characters we wish to print. The printing is accomplished by loading the X register with 00 and stepping through the data table until a 00 is encountered. Each byte loaded is put into the accumulator and printed via the JSR to COUT ($FDED). When the 00 is finally reached, the BEQ on line #13 is taken and we return from the routine via the RTS at DONE.

The new item of interest in this listing is the use of the $84 as the first character printed. This will be printed as a control-D and the word Catalog that follows executed as a DOS command.

This is the essence of this month's message, along with the routines. Any DOS command can be executed from machine language exactly the same way it's done from Basic. One need only precede the command with a control-D and terminate the command with a carriage return. (Note: the Read and Write are something of an exception to this but can still be done with only minor compensations).

Because DOS looks at all characters being output, it will see the control-D character and behave accordingly.

Try entering this program and then calling with either a 300G from the Monitor, or a CALL 768 from Basic.

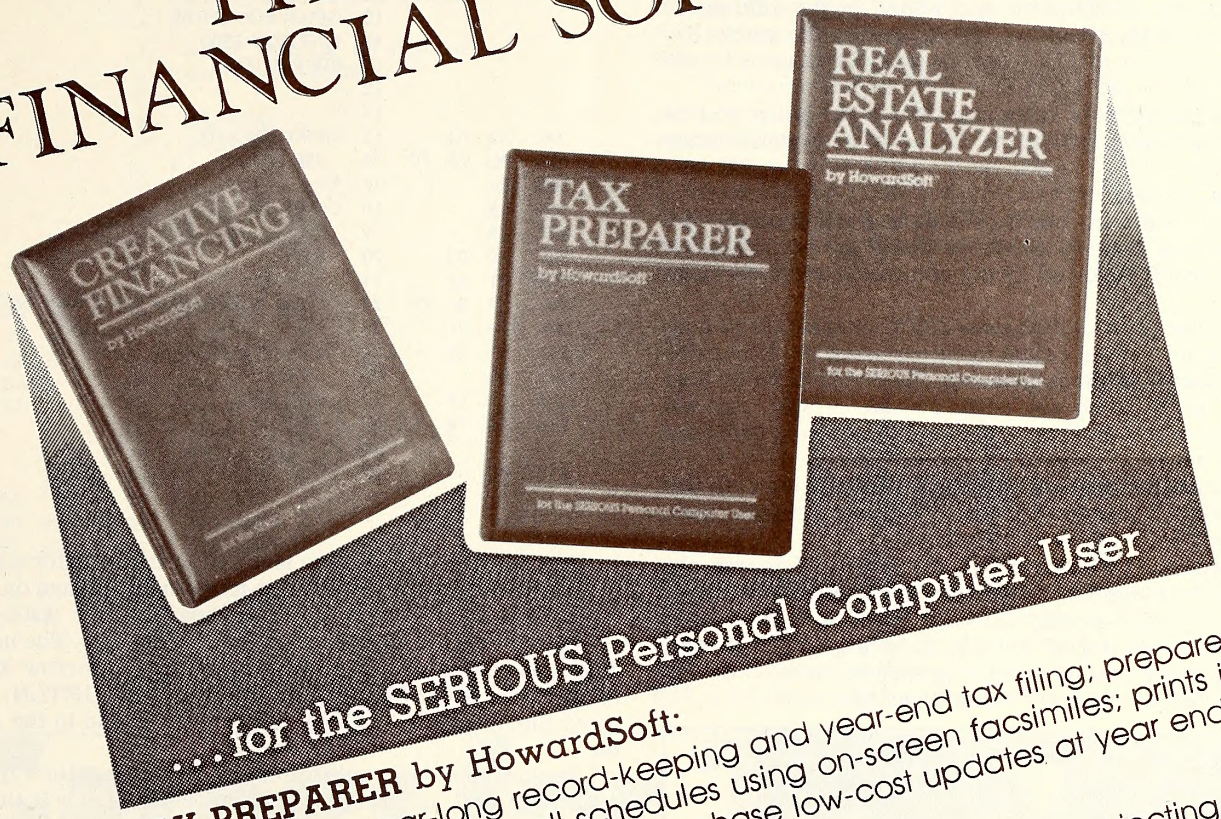
This next print routine is more involved but does offer some advantages. The advantage is that the hex or ASCII data for what you want to print can immediately follow the JSR PRINT statement, which parallels Basic a little more closely, and avoids constructing the various data blocks. The disadvantage is that the overall code is longer for short programs such as this. The general rule of thumb is to use the data-type when you have only to print once or twice during the program, and to use the following type when printing many times.

The logic behind its operation is slightly more complex than the previous routine, but I think you'll find it quite interesting.

Here's the new method:

```
                          1  ************************
                          2  * SPECIAL PRINT ROUTINE *
                          3  ************************
                          4  *
                          5    OBJ $300
                          6    ORG $300
                          7  *
                          8  PTR EQU $46
                          9  EQU $FDED
                          10 *
0300: 20  0D  03          11 ENTRY JSR PRINT
0303: 84                  12 E0 HEX 84
0304: C3  C1  D4          13 ASC "CATALOG"
030B: 00                  14 HEX 00
030C: 60                  15 DONE RTS
                          16 *
030D; 68                  17 PRINT PLA
030E: 85  46              18    STA PTR
0310: 68                  19    PLA
0311: 85  47              20    STA PTR+1
0313: A0  01              21    LDY #$01 ; PTR HOLDS 'E0' — 1 HERE
                          22 *
0315: B1  46              23 P0 LDA (PTR),Y
0317: F0  06              24    BEQ FNSH
0319: 20  ED  FD          25    JSR COUT
031C: C8                  26 *
031D: D0  F6              27    BNE P0 ; (MOST ALWAYS)
                          28 *
031F: 18                  29 FNSH CLC
0320: 98                  30    TYA
0321: 65  46              31    ADC PTR
0323: 85  46              32    STA PTR
0325: A5  47              33    LDA PTR+1
0327: 69  00              34    ADC #$00
0329: 48                  35    PHA
032A: A5  46              36    LDA PTR
032C: 48                  37    PHA
032D: 60                  38 EXIT RTS
                          39 * WILL RTS TO DONE INSTEAD OF
                          40 * E0 !
                          41 *
```

THE BEST IN FINANCIAL SOFTWARE



. . . for the SERIOUS Personal Computer User

TAX PREPARER by HowardSoft:
A package for year-long record-keeping and year-end tax filing; prepares returns for several forms & all schedules using on-screen facsimiles; prints in IRS format for filing; buy now and purchase low-cost updates at year end.

REAL ESTATE ANALYZER by HowardSoft:
A tool for objectively comparing alternative investments and projecting future results; professional in both analyses of cash flow & R-O-I and client-oriented report printouts; allows what-if studies for changes in economy, loans, rents, taxes, & more .

CREATIVE FINANCING by HowardSoft:
A flexible package for evaluating real-life loans & investments; provides cash flow & R-O-I projections as well as payment tables and objective decisions; can be used for yield to maturity of discounted bonds, present value of future income stream, buy vs. lease decisions, APR of loans, & much more .

NOW AT YOUR LOCAL DEALER

Howard Software Services
. . . for the SERIOUS Personal Computer User
6713 Vista del Mar | La Jolla, CA 92037 | (714) 454-5079

This one is rather interesting in that it uses the stack to determine where to start reading the data. You'll recall that when a JSR is done, the return address minus one is put on the stack. Upon entry to the Print routine, we use this fact to put that address in PTR, PTR+1. By loading the Y register with #$01 and indexing PTR to fetch the data, we can scan through the string to be printed until we encounter 00, which indicates the end of the string.

When the end is reached, the BEQ FNSH will be taken. In that section, the Y register (the length of the string printed) is transferred to the accumulator and added to the address in PTR, PTR+1, and the result pushed back onto the stack. Remember that the old return address was E0−1 until it was pulled off.

Now when the RTS is encountered, the program will be fooled into returning to DONE instead of E0 as it would otherwise have done.

To summarize, then:

1) Any DOS command can be executed from machine language just as it is done in Basic by doing the equivalent of Printing a control-D followed by the command and a carriage return.

2) A data-type print routine uses ASCII characters in a labeled block, which is then called by name using the X register in a direct indexed addressing mode. The string to be printed should have the high bit set (ASCII value + $80), and the string must be terminated by a zero (at least when using the routine given here).

3) A JSR to a special print routine can also be done. In this case the ASCII data should immediately follow the JSR, again have the high bit set, and end in a 00.

Input Routines. The other side of the coin is, of course, the Input routine. You might be surprised by the number of times I get calls from people saying, "If only the input in such-and-such program would accept quotes, commas, etc." The solution is actually quite simple and is presented here.

In its simplest form, the routine looks like this:

```
                          1  ****************************
                          2  *  INPUT ROUTINE FOR BINARY *
                          3  ****************************
                          4  *
                          5  * STORES STRING AT PTR LOC
                          6  *
                          7    OBJ $300
                          8    ORG $300
                          9  *
                         10  GETLN EQU $FD75
                         11  BUFF EQU $200
                         12  PTR EQU $46
                         13  *
                         14  *
0300: A2 00              15  ENTRY LDX #$00
0302: 20 75 FD           16    JSR GETLN
0305: 8A                 17  *
0306: A8                 18  CLEAR TXA
0307: C8                 19    TAY ; T-REG = LEN NOW
0308: A9 00              20    INY
030A: 91 46              21    LDA #$00
030C: B9 00 02           22  C2 LDA BUFF,Y ; PUT END-OF-STRING MARKER.
030F: 29 7F              23     AND #$7F
0311: 91 46              24     STA (PTR),Y
0313: 88                 25     DEY
0314: C0 FF              26     CPY #$FF
0316: D0 F4              27     BNE C2
                         28  *
0318: 60                 29  DONE RTS
```

The heart of this routine is a call to the Monitor's GETLN routine, which gets a line of text from the keyboard or current input device and puts it in the keyboard buffer ($200−2FF).

This saves our having to write one ourselves. The beauty of this is also that all the escape and left/right arrow keys are recognized. When the routine returns from GETLN, the entered line is sitting at $200+. The length is held in the X register.

At this point we could, presumably, just return from our routine as well; but as it happens, all the data now in the buffer has the high bit set—that is, #$80 has been added to the ASCII value of each character. Because Applesoft in particular, and many other routines in general, don't expect this, the high bit should be cleared before returning. Also $200+ will hold only one string at a time, so there should be some provision for putting the string to some final destination.

Both are accomplished in the Clear section of this routine. First the length of the string is transferred via the TXA, TAY to the Y register. My preference is then to mark the end of the string by bumping Y by one and storing a 00 as a terminator. That step is optional.

Next, C2 begins a loop that loads each character into the buffer, does an AND with #$7F, and then stores the result at a location pointed to by PTR, PTR+1 plus the Y register offset.

The AND #$7F has the effect of clearing the high bit by forcing bit 7 to a 0.

The Y register is then decremented and the loop repeated until the DEY forces Y to an $FF. This will indicate that the last value was $00, and we have thus completed scanning the buffer.

This routine will work fine as long as you're willing to manage the string entirely yourself once it gets to the PTR, PTR+1 location.

As noble as it might be to write programs entirely in machine language, I usually prefer to write in both Applesoft and machine. This is because unless speed is required, Applesoft does offer some advantages in terms of program clarity and ease of modification. After all, if there were no advantages to Applesoft, why would somebody have written it in the first place?

So, to that end, here are two new listings, the first in Applesoft:

```
10  IN$ = "X"
20  PRINT "ENTER THE STRING: ";
30  CALL 768: IN$ = MID$(IN$,1)
40  IF IN$ = "END" THEN END
50  PRINT IN$: PRINT: GOTO 20
```

and the second in assembly language:

```
                        1  ****************************
                        2  * INPUT ROUTINE FOR FP BASIC *
                        3  ****************************
                        4  *
                        5  * IN$='"' MUST BE 1ST VARIABLE
                        6  * DEFINED IN PROGRAM!
                        7  *
                        8    OBJ $300
                        9    ORG $300
                       10  *
                       11  GETLN EQU $FD75
                       12  VARTAB EQU $69
                       13  BUFF EQU $200
                       14  *
                       15  *
0300: A2 00            16  ENTRY LDX #$00
0302: 20 75 FD         17  GETLN
0305: A0 02            18    LDY #$02
0307: 8A               19    TXA
0308: 91 69            20    STA (VARTAB),Y
                       21  * STORE 'X-REG = LEN OF IN$'
                       22  * IN LEN BYTE OF IN$
                       23  *
030A: C8               24    INY
030B: A9 00            25    LDA #$00
030D: 91 69            26    STA (VARTAB),Y
030F: C8               27    INY
0310: A9 02            28    LDA #$02
0312: 91 69            29    STA (VARTAB),X
                       30  * SET LOCATION PTR OF IN$ TO
                       31  * $200 (INPUT BUFFER)
                       32  *
0314: 8A               33  CLEAR TXA
0315: A8               34    TAY ; Y-REG = LEN NOW
0316: B9 00 02         35  C2 LDA BUFF,Y
0319: 29 7F            36    AND #$7F
031B: 99 00 02         37    STA BUFF,Y
031E: 88               38    DEY
031F: C0 FF            39    CPY #$FF
0321: D0 F3            40    BNE C2
                       41  *
0323: 60               42  DONE RTS
```

The important difference to notice here is that IN$ has been defined as the first variable in the Applesoft program, and that the machine language routine uses this fact to transfer the string to Applesoft.

The way this is done begins at XFER. When an Applesoft string variable is stored, the name, length and location of the string are put in a table, whose beginning is pointed to by locations $69, 6A (VARTAB, VARTAB+1).

Since IN$ was the first variable defined, we know that its name and pointer will start at wherever VARTAB points. The name is held in positions 00 and 01, the length in 02, and the location in 03 and 04.

	+4	loc + 1
	+3	loc
	+2	length
LOMEM:	+1	N$
		1

—LOMEM: ($69,6A)

By loading the Y register with #$02, we can store the length of the entered string in the proper place. The location of IN$ is then set to $200 by putting the appropriate bytes into positions 3 and 4. Now Applesoft is temporarily fooled into thinking that IN$ is at $200—right where our input string is held!

The routine finishes by clearing the high bit, as before, and then returning with the RTS.

When the return is done, line 30 of the Applesoft program immediately assigns IN$ to itself in such a way as to force Applesoft to move IN$ from where it was in the input buffer to a new location up in its usual variable storage area. The net result can be obtained in various other ways besides the MID$ statement, but the way shown is the least intrusive in terms of affecting other variables. (You could use A$=IN$:IN$=A$, but then you'd need a second variable in your program—no problem, just more names to keep track of.)

Try assembling this routine and the Applesoft program. Make sure the input routine is loaded at $300 before running the Applesoft program. Note that you can enter commas, quotes, control-C's, etc. Only END or pressing reset should be able to exit this routine.

Next month, we'll put all this together into a mini-data base that stores and retrieves a number of strings with a normal disk file.

The Confessional. Alas, I've erred. The July and August issues of *Assembly Lines both* contained errors that I would love to say were merely typographical; but, unfortunately, I couldn't convince the editor and typesetters to take the blame for me.

In the July issue, I discussed the ADC and SBC commands. Please note that ADC *does* clear the carry if the result is $FF or less. SBC behaves similarly, in that if the result of the subtraction requires a borrow, the carry is cleared (borrow set, as in $50−$80); otherwise the carry is set (borrow clear as in $40−$10).

Please note that the CLC or SBC still needs to be done before an addition or subtraction operation to assure an accurate result.

The August issue had twice as many errors. Again the carry was involved. (Sigh. . . .)

On page 60, I mentioned that the carry should be cleared before calling RWTS so that an accurate check for an error could be done when it returned. At the time I was under the impression that RWTS did not specifically clear the carry if no error occurred. It does. This can be seen by examining the end of the RWTS routine itself, or demonstrated by experiment. From my own experimentation at an earlier time, I somehow got the impression that the carry was not specifically cleared by RWTS after a no-error call.

In retrospect, I think this probably came about through another bug that I've since discovered. Some of you may have experienced a beep from the Monitor when calling the RWTS utility given in the August article. This seems to be intermittent; the "beep" doesn't occur if you have done a legitimate DOS command, such as Catalog, prior to calling the routine.

I have isolated the cause to be a result of the contents of location $48. If this location holds a $00, no bell is given when you call a routine. However, if $48 holds a nonzero number, everything still works okay, but the Monitor talks to you every time you call the routine. I suspect it was this effect that made me think RWTS wasn't doing its job with the carry. C'est la guerre. . . .

The second error was in regard to which track and sector to read to alter DOS for the disk volume modification. In the first example it instructed the user to read track 1, sector D by typing in:

```
06: 01 0D 01 60 00 10
```

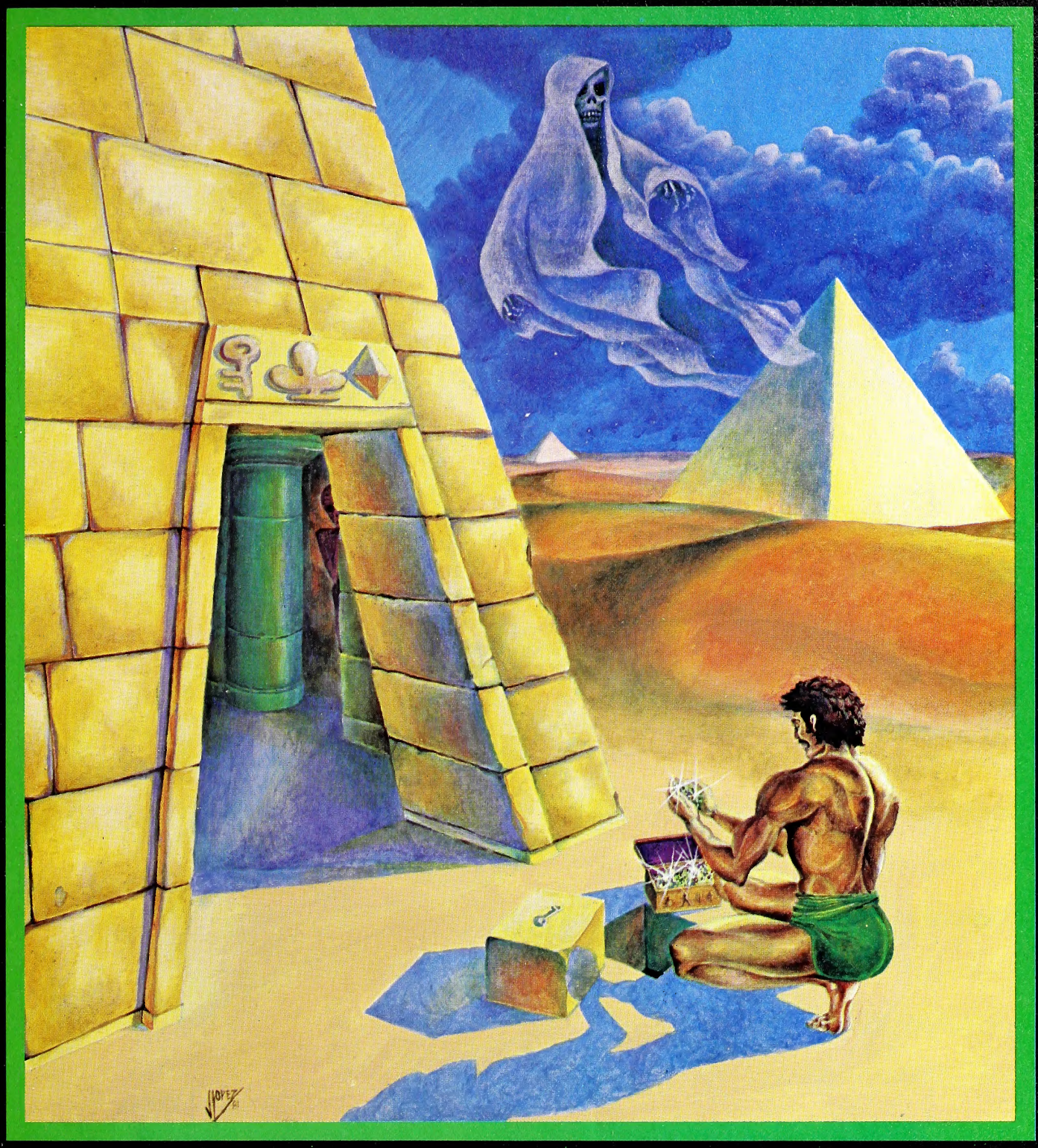
The line should have read:

```
06: 02 02 01 60 00 10
```

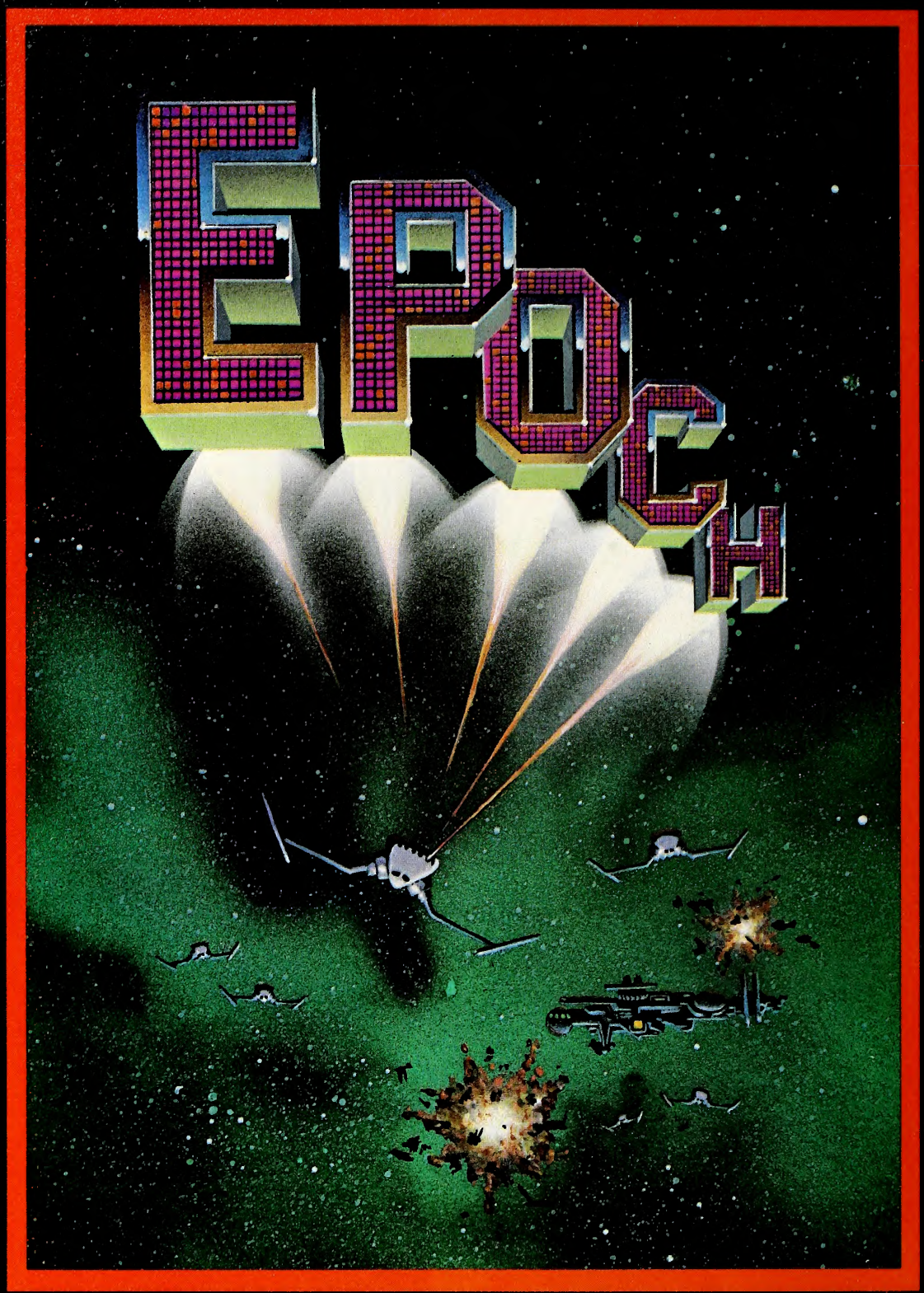
This would have read in track 2, sector 2, which is where the data we're interested in actually resides.

My thanks to the many readers who responded to this test of their attention (and perseverance) and wrote me such kind letters advising me of the errors.

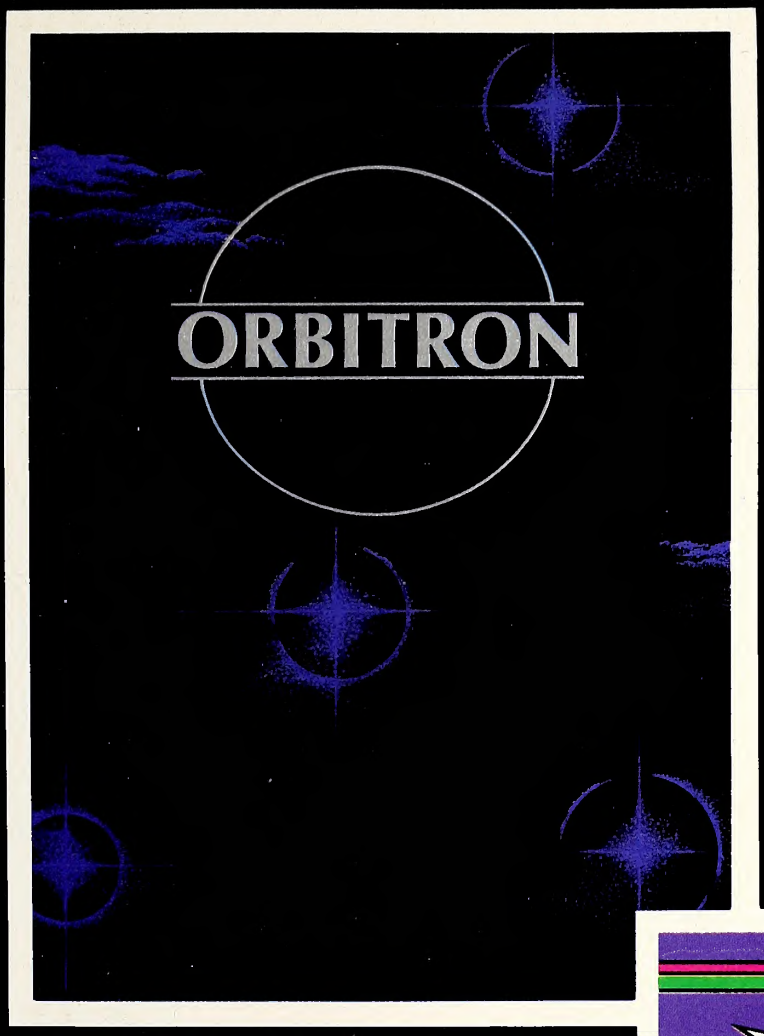
My apologies for any confusion this may have caused. Better luck next time? ■



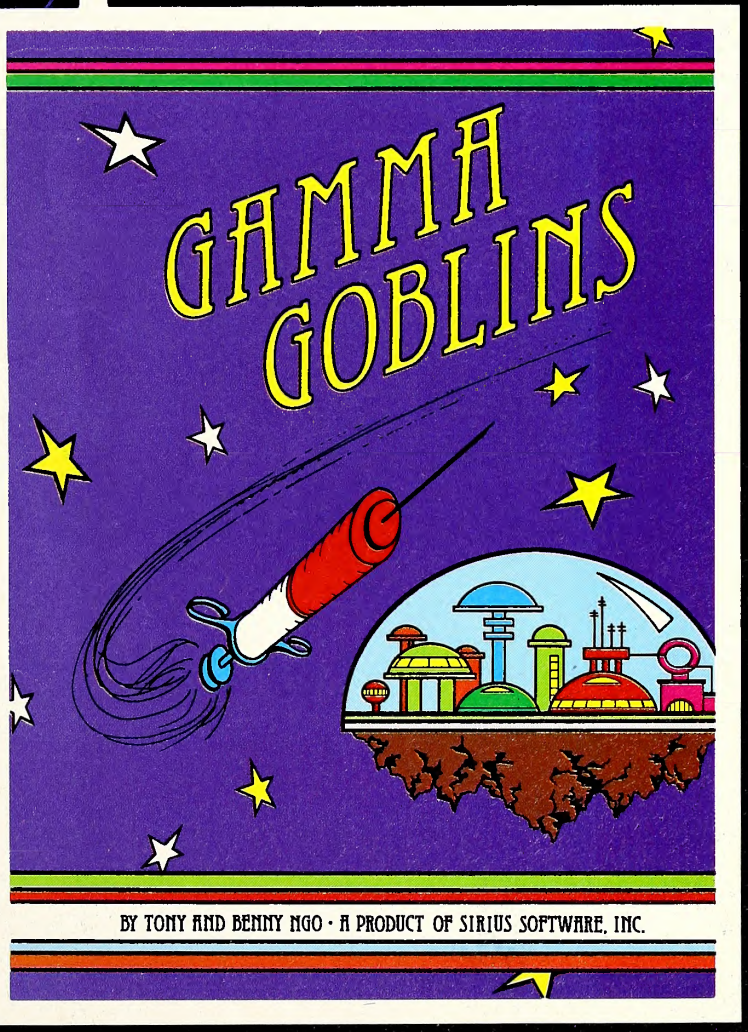
Play Copts & Robbers
In The Tombs Of Ancient Egypt



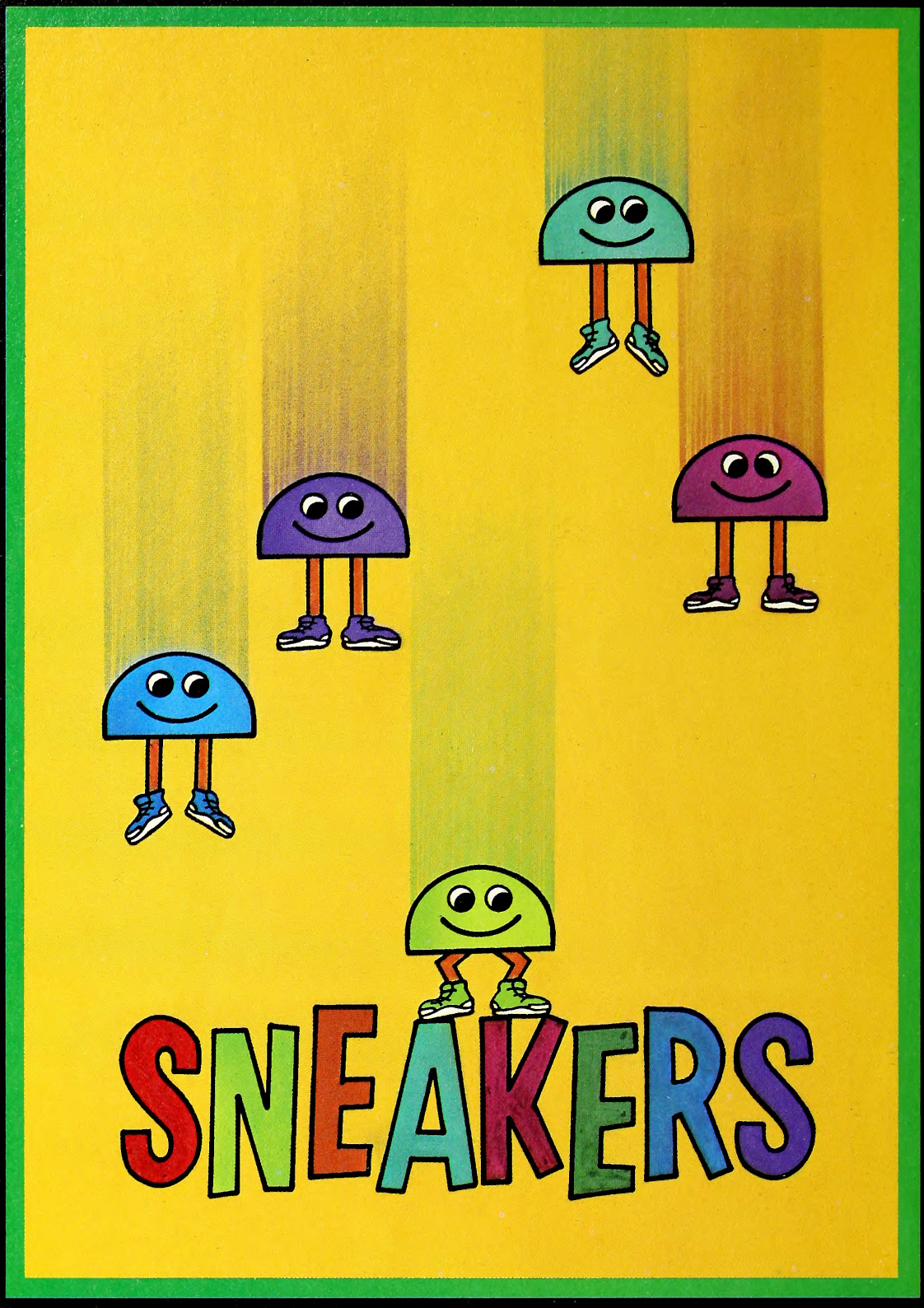
Where your secret weapon is the fourth dimension...



In the center of an orbiting space station you are protected only by a revolving force shield. Enemy forces are advancing from all directions and begin to place killer satellites in orbit around your station. And then, look out for the meteors!

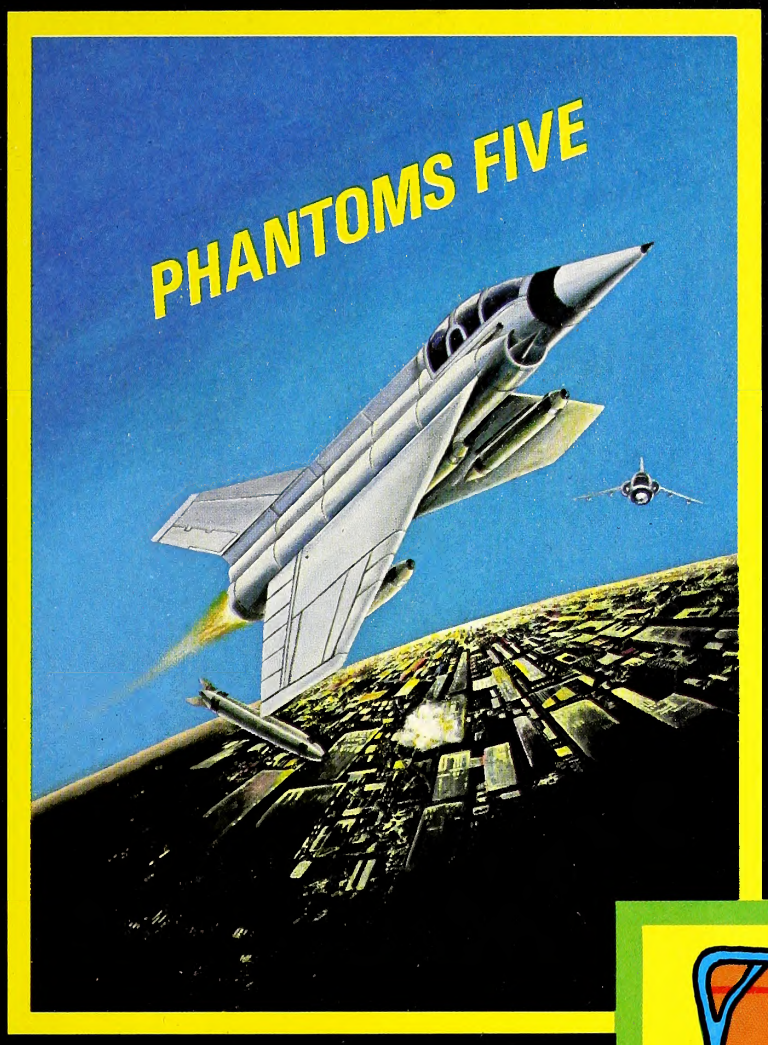


A "bloody" good game for the true-blue game freak. Your mission in this exploratory operation is to deliver whole blood to Hemophilia, a city in the sky, and return to Anemia Base before the Gamma Goblins overcome you. A real heart stopper!

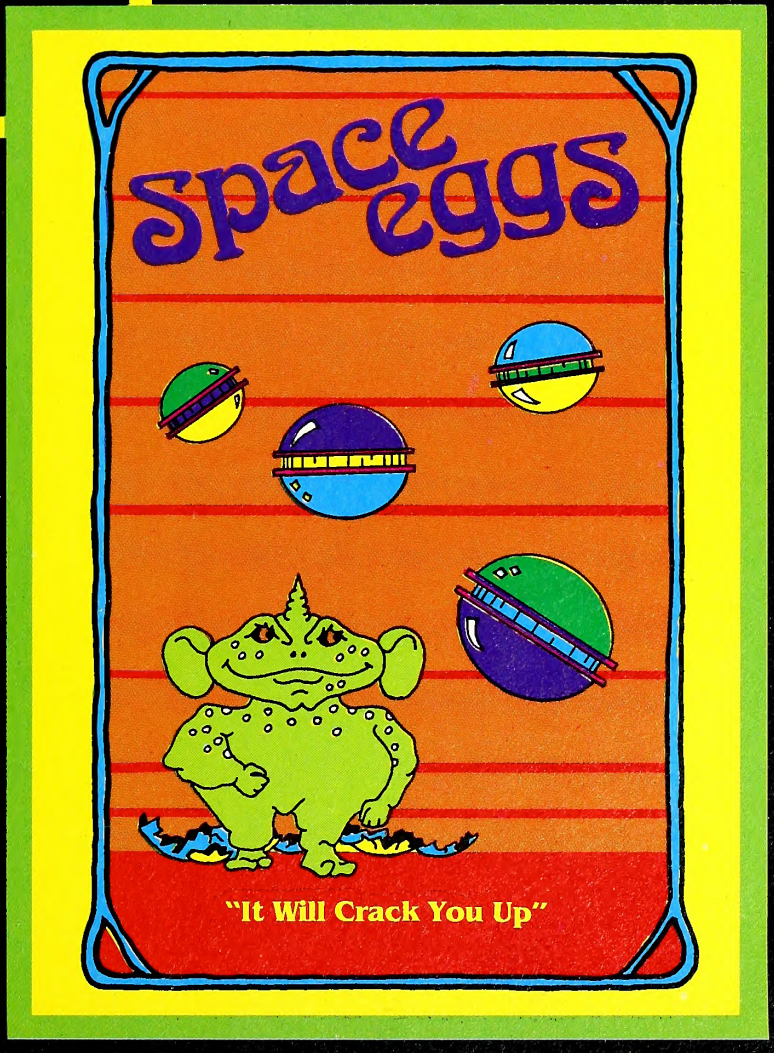


What say we go out and stomp a few???

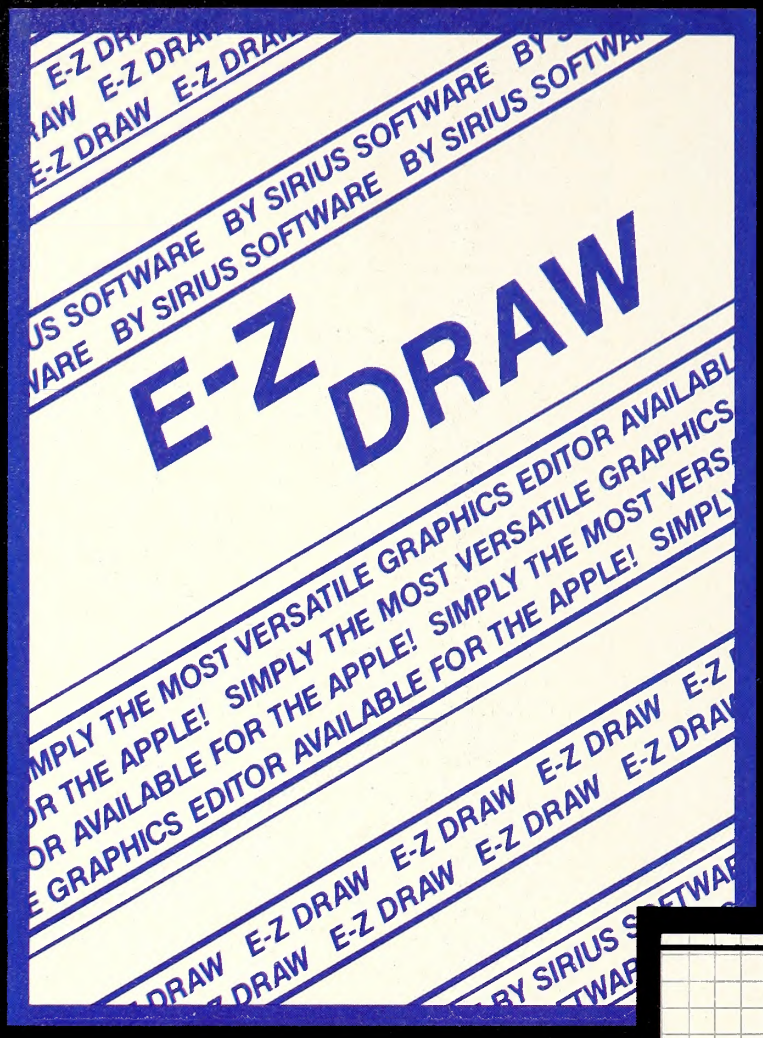
Endless Excitement Stomping Sneakers And A Swarm Of Other Creatures



Phantoms Five simulates a fighter-bomber mission in real time, three dimensional color graphics. While you try to make your bombing run, you have to avoid being hit by anti-aircraft fire and fight off enemy aircraft as well.



Hatch some fun with the Spiders, Wolves, Lips, and Fuzzballs. Space Eggs will crack you up! Each package includes a multi-color T-shirt iron-on that says "I FRIED THE SPACE EGGS."



This is the graphics editing package we based our business on. Includes the Higher Text Character Generator by Ron & Darrell Aldrich and over 20 original and imaginative type styles.

D2D0073

A Product Of Sirius Software, Inc.

Pascal Graphics Editor

PGE

The professional graphics editing package for use within the Pascal environment.

GORGON is here...

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

BOTH BARRELS

A two game pack featuring "High Noon" and "Duck Hunt" You'll love the bad guy that falls off the roof and the dogs fighting over the ducks. Fun for the young and the young at heart.
Copyright 1980 By Sirius Software, Inc.

STAR CRUISER

Save yourself from the swooping aliens! This is a fast action arcade style game that can be played from ages three and up, but beware, the difficulty increases with each new wave of aliens.
Copyright 1980 By Sirius Software, Inc.

CYBER STRIKE

Interstellar challenge for the dedicated arcade gamer. You are in command of a light transport ship equipped with Hyperspace Drive, Antimatter Torpedoes, Local and Galactic Sensors, Meteor Shields, and an Instrument Panel which continually tabulates all information vital to your mission. You alone can prevent the clone take over of the allied settlement bases. WARNING . . . this game requires practice to play successfully.
Copyright 1980 By Sirius Software, Inc.

AUTOBAHN

Hair raising excitement at 120, 160, and 200 kilometers per hour! Drive through heavy traffic, oil slicks, narrow roads, and dark tunnels (with headlights). Watch out for the fire trucks! Only on the Autobahn can you drive this fast.
Copyright 1981 By Sirius Software, Inc.

PULSAR II

A unique two game series that provides scoring options for separate or combination game play. To destroy the "Pulsar" is no easy task. It is surrounded by spinning shields that send out orbs of energy aimed directly at you. "The Wormwall" places you in one of the strangest mazes ever created. The walls do not connect. Openings only occur temporarily as moving colored segments in the walls cross. In addition, there are munching mouthers in each level of the maze ready to gobble you up should you misjudge the time and location an opening will occur.
Copyright 1981 By Sirius Software, Inc.

Contact Your Local Computer Dealer For More Information • Dealer Inquiries Invited

Sirius Software, Inc.

2011 Arden Way #2, Sacramento, California 95825

PROGRAMMING: Copts & Robbers was programmed by Alan Merrell and Eric Knopp. Epoch was programmed by Larry Miller. Orbitron was programmed by Eric Knopp. Gamma Goblins was programmed by Tony and Benny Ngo. E-Z Draw was programmed by Nasir Gebelli and Jerry W. Jewell. Pascal Graphics Editor was programmed by Ernie Brock. Sneakers was programmed by Mark Turmell. Gorgon, Phantoms Five, Space Eggs, Both Barrels, Star Cruiser, Cyber Strike, Autobahn, and Pulsar II were programmed by Nasir.

SYSTEM REQUIREMENTS: All software mentioned in this advertisement require an Apple II or II+ with 48K with the following exceptions: E-Z Draw requires a 48K Apple with Applesoft in ROM (or a 64K Apple II or II+) Pascal Graphics Editor requires an Apple II or II+ with Language System.

NEWSPEAK

☐ Leonardo Da Vinci was a man ahead of his time. An inventor, engineer, experimental scientist, as well as artist, Da Vinci was drawing pictures of flying machines and designing armored combat vehicles when he was also creating some of the greatest artwork of all time. No doubt he would have found a computer useful if it had been around in the sixteenth century.

Da Vinci didn't have the benefit of modern technology, but he would probably approve of the work of a couple of scientists today who are using a computer to uncover some of the hidden secrets of his own painting, the Mona Lisa.

Dr. John Asmus, manager of the laser laboratory at Maxwell laboratory in San Diego, California, and Dr. Carlo Pedretti, professor of art history at the University of California at Los Angeles, have teamed up to apply technological advances to the arts. With the help of a computer, they've proven that the Mona Lisa in the Louvre looks very different from Da Vinci's original. It's still the same painting, but a yellow varnish, which was applied long ago by art restorationists to preserve it, has significantly altered the original colors over the years and has covered up some important details.

Asmus said he and Pedretti were interested in what the Mona Lisa looked like before the varnish dulled the colors, so they employed methods NASA uses to touch up photos of Jupiter and Saturn. They first had a high-resolution, large format photograph taken of the Louvre Mona Lisa. The photograph was flown to the Jet Propulsion Laboratory in Pasadena for image processing on their computers. The photo was divided into five million squares, and the intensity of the colors was measured in each region, producing a magnetic tape with all the digital information needed by the computer. The computer then gave them a simulated picture of what the Mona Lisa would look like if all the varnish were peeled away.

The sky in the painting turned from brown to blue in the cleaned-up version, and the color of Mona Lisa's skin turned from brown to alabaster. Pedretti is also interested in the veil Mona Lisa is wearing on her head and over her shoulders. There are details underneath the veil that are beginning to show up as the dense areas of the painting are lightened.

The project probably never would have been attempted without a computer, which provided a rapid and convenient way to take off the varnish without having to touch the original picture, says Asmus. He and Pedretti plan to do more sophisticated filtering to bring up the intensity of the dark areas in hopes of discovering further secrets of the Mona Lisa.

Asked why the Mona Lisa was so special in comparison to some of Da Vinci's other works and why it caused so much hoopla among art lovers, Asmus deferred to the opinion of his colleague, Carlo Pedretti, a prominent Da Vinci scholar. According to Asmus, Pedretti has said that "the Mona Lisa is not his favorite painting, and that some of Da Vinci's other works are far superior."

☐ Students can get into the colleges of their choice now—even the expensive ones—with the help of the computerized National Scholarship Research Service. Daniel Cassidy of San Rafael, California, who started the service two years ago, employs fourteen Apple computers that specialize in matching students with little-known, off-beat scholarships.

For example, the $500 Kenneth Gunter Fellowship is waiting for a student at the University of Arizona; applicant must have a good academic record and be able to rope cattle. There are also funds available for left-handed people at Juniata College in Pennsylvania, for sons and daughters of fishermen at Tufts, and for students of Polish ancestry from the Kosciuszko Foundation of New York City.

The service lists 50,000 private grants offered by corporations, trust and memorial funds, religious groups, and unions, with awards ranging from $300 to $20,000 for four years in college. For a $35 fee, Cassidy sends students a brief questionnaire from which his computer will come up with thirty to fifty private sources of money each student can apply for and stand a good chance of getting.

Cassidy, who financed his own college education with scholarships and grants, says there's approximately a billion dollars in financial resources available for students from the private sector; last year, about $140 million dollars in grants were never claimed. But with the government's proposed cuts in federal student loans and scholarships ahead, Cassidy emphasized, students can't afford to let these private-sector sources of funds go untapped.

Most of the grants are easy to get, and many do not require financial need or good grades. As a matter of fact, the United States Office of Education offers a grant for students who "must be charac-

terized as academic risks for college education."

☐ **Apple Inc.** It almost seems as if they've defied the laws of nature; they most certainly have defied the laws of business. Apple Computer recently released their third quarter profit and sales improvement report, showing a rise in net income to $11.9 million or $.21 per share, compared to $2.7 million or $.06 per share in the same quarter last year. This is a 334 percent profit increase on a 179 percent sales gain for the third quarter of fiscal 1981 over the comparable 1980 period.

"Gratifying," is how Mike Markkula, Apple's president and chief executive officer, describes the increased sales volume and the overall momentum of his company. In spite of the negative pressures of high interest rates and a lackluster economy, demand is remaining strong as Apple moves into the last quarter of fiscal 1981, notes a pleased Markkula.

The stock figures are also defying economic law, according to a recent article in *Apple Times*, the in-house journal for Apple employees, which reports that the secondary stock offering in early June triggered a six-dollar rise in the value of Apple stock. VP-general counsel Al Eisenstat explained that approximately three million shares of stock were absorbed into the market immediately, and the price continued to rise after the sale was completed. The trading price of Apple stock rose from $27 a few weeks before the offering to $33 after the sale. The sale did not bring capital into Apple but provided an opportunity for a number of employees to sell stock acquired under stock option programs.

☐ When IBM finally unveiled its entry in the micro market in August, *Newsweek* (August 24) headlined its announcement, "IBM wants a bite of the Apple." Xerox also reserved a seat on the bandwagon recently, but Mike Markkula doesn't seem worried.

Apple anticipates far more vigorous competition coming up from proven data processing companies, Markkula says. "We feel their presence will stimulate worldwide demand for the products that Apple originated and in which it specializes."

Some of the latest figures from Markkula show more than two hundred fifty thousand Apples in use—an installed base that "drives an enormous amount of applications software and contributes to the growth of our industry," he says.

Apple research and development expenditures in the current quarter increased to $5.5 million from $2.2 million last year. Markkula believes programs related to advanced microcomputer hardware and software will also reinforce Apple's position in this expanding field. The executive says the company is firmly committed to the Apple III, which he calls "an excellent system whose market will develop more rapidly as additional software, now in the process of completion, becomes available."

☐ In the meantime, Apple continues to grow. Production started in early July at a new 125,000-square-foot assembly plant in Singapore. Logic boards, analog cards, and disk interfaces for both the Ireland and United States assembly plants are now being turned out. By 1982, Apple Singapore will be producing Apple II Pluses, disk drives, and printers.

Here in the United States, Apple opened a sales office last May in New York City to meet the demand for Apple systems in New York and northern New Jersey; a month later, the southeast sales region opened the doors of an Atlanta office. And not to forget Sunnyvale, California, in the valley where it all started in the first place (ho ho ho): Plans are also underway to open two new buildings in nearby Cupertino—Bandley V and Bandley VI.

☐ No question about it—Apple is growing fast. But are they growing too fast for their own good? Not in the opinion of the Peninsula Chapter of the Stanford Business School Alumni Association. Citing "superior management of rapid growth" over the past three years, they have named Apple Computer the Entrepreneurial Company of 1981.

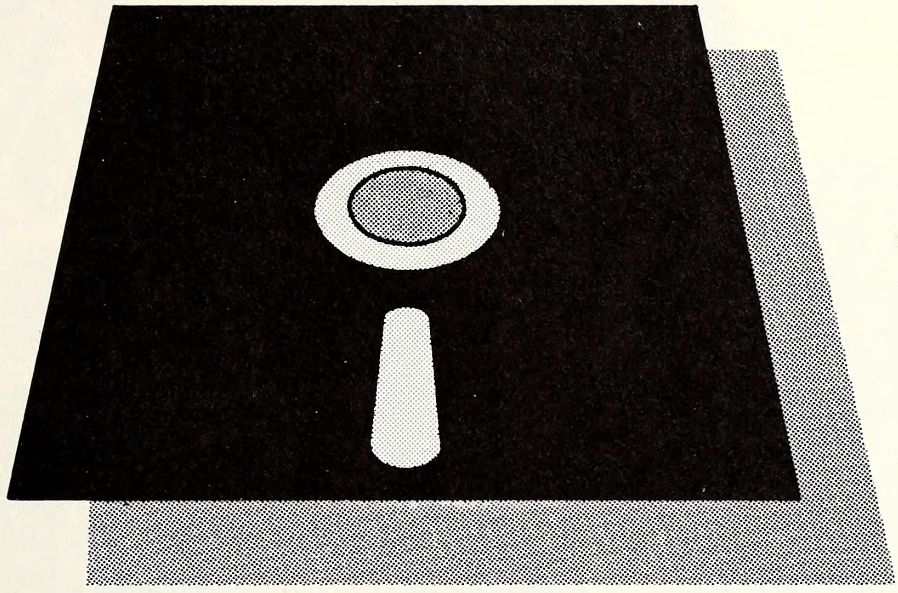
Some other growth figures from the company to ponder: For the first nine months of fiscal 1981, Apple earned $28.5 million or 51 cents per share, compared with $8.2 million or 17 cents per share last year. Sales for the nine months increased to $237.1 million from $75.7 million in fiscal 1980. Apple's profit margin before taxes rose to 26 percent of net sales in the third quarter of fiscal 1981, up from 23 percent in the second. This change, explains Markkula, reflects lower prices for semiconductor components and generally improved Apple manufacturing efficiencies and reduced emphasis on pass-through sales of non-Apple peripheral products.

Stockholders' equity in Apple amounted to $164.1 million at the end of the third quarter, compared with $19.8 million a year earlier. Total assets were $222.3 million, compared to $41.7 million at the end of the fiscal 1980 third quarter.

☐ **Selective Computers.** Computers aren't going to escape the draft, either. Selective Service officials in Washington say that, if the draft is revived, a single national computer will go into action. The computer will perform most of the duties formerly carried out by local boards, keeping all the records of potential draftees who register. Those to be drafted will be selected by random lottery by the computer. The Selective Service believes draft notices can be sent out in two days if Washington makes an emergency decision under the new system. ❑■

$600

All You Ever Need To Spend



TITLE: "THE PROGRAM WRITER/REPORTER*"

Enables ANYONE to write complete, running, debugged BASIC LANGUAGE Programs in 35 to 40 minutes with NO PRIOR PROGRAMMING KNOWLEDGE OR ABILITY.

IF you are one of the many who bought a microcomputer in the belief that with just a little studying you could write your own programs, you now know that you can't.

IF you, as a businesman, thought you could have stock software modified at a reasonable cost with reasonable results, you know that's not possible either.

IF you are a hobbyist getting tired of the untold hours it takes to write a program, only to find it takes more hours to debug than to write.

IF you are a skilled programmer you don't have to be reminded of the repetitious time spent on each new application.

IF you have left your micro-computer sitting somewhere gathering dust . . . meet THE PROGRAM WRITER/REPORTER* .

THE PROGRAM WRITER/REPORTER* is not just another data base generator.

THE PROGRAM WRITER/REPORTER*, at your direction, makes complete running programs that are thoroughly documented, easy to modify at any time by YOU!

THE PROGRAM WRITER/REPORTER* cuts programming time up to 90% for a skilled programmer.

THE PROGRAM WRITER/REPORTER* will make anyone a skilled programmer in 30 to 35 minutes!

THE PROGRAM WRITER/REPORTER* does the work! You can answer the simple direct questions and THE PROGRAM WRITER/REPORTER* CREATES . . . AND ALL IN BASIC LANGUAGE.

Q. After THE PROGRAM WRITER/REPORTER* has produced a program, can it be modified?

A. Yes, the resulting program is modular, fully documented and readily accessible for alterations or deletions.

Q. Does the program created use so much disc space that there is very little space left for the record storage?

A. No, the code produced is extremely compact despite complete documentation. If requested THE PROGRAM WRITER/REPORTER* will even "pack" or compress information. You may even delete the "remarks" making it even more space efficient.

Q. Must I be expert or even conversant with Basic Language?

A. No, all questions to and answers from the operator require no computer language knowledge, simple every day English will do.

Q. What about math ability?

A. If you can count your fingers and toes, you'll have no problems.

Q. Will the programs which I produce with THE PROGRAM WRITER/REPORTER* be bulky, slow or amateurish?

A. No, the resulting programs will be sophisticated and extremely fast operating. For example, should you create a mailing list or inventory program, the time for any record to be retrieved and displayed from a full disc would take a maximum of 1 second.

Q. Must the programs produced conform to a pre-determined format and file length?

A. No, you determine format and file size to fit your requirements.

Q. Can I develop my own business programs?

A. yes.

Q. What are the limitations? What programs can I produce with THE PROGRAM WRITER/REPORTER* ?

A. Your own ingenuity and hardware limitations. 100's of different programs.

TECHNICAL ASPECTS

The Reporter Package - makes reports your way.

Record access by a hashing algorithm guaranteeing fast record retrieval.

Duplicate keys permitted.

Record deletion automatically supported.

Record access and file maintenance is user transparent.

Minimal disc overhead since there is no special assembly language routine called. No "Basic" overhead.

Programs produced can be transported between 6800, 6502, 8080, Z80, 8085, 8086 and Z8000 based systems.

Can be used with Micro-Soft Basic and CP/M systems.

Complete file maintenance including up-date of any record in any field, delete and add new records even with duplicate key.

VANLOVES

800-255-5119

Vital Information Inc. 7899 Mastin Dr. Overland Park KS 66204


Gamemaster
The Mansion of the Future Just a Modem Away

"Enter the gamemaster's mansion and step into the future. . . ." The doorkeeper's words as he welcomes you to this remarkable building are not fluff. Architecturally, the sprawling, elaborate Victorian house smacks of decadent, long-ago grandeur; but everything else about it speaks of the future.

The first sign of the future is the address, not of the company, but of the house itself. You won't find this mansion on any street, boulevard, or lane. You can't reach it by foot or car. It resides in the mind of a computer, and you go there by way of your modem. Your key is a membership in GameMaster.

As soon as you connect with the mansion, a doorkeeper checks your account number and password, then ushers you in. He announces your arrival on the mansion's public address system, and any visitors now in the house see the announcement on their monitors. If you like, you may look at the guest book, which tells you all the members who are in the house when you get there. You're also invited to send a message over the PA if you wish. This is your chance to say, "I'm looking for someone to play board games. Meet me in the parlor if you're interested."

Checking the Premises. Your entry accomplished, you're on your own in the foyer. You're free to wander through the house, up and down hallways and stairs, or to portal to any room you choose—a process resembling beaming up in "Startrek."

In each room, you may choose a room description. The descriptions bring personality and color, teasing your mind's eye with images of warm hearths, musty books, high ceilings, and oaken splendor. In the final testing stages is a driver that will enable Apple owners to see the house in animated hi-res.

The mansion has thirty-nine rooms and six stories. An entire floor is devoted to game rooms. The rooms on the other floors offer services, information, and other activities in which you can participate.

The library holds books that describe all the rooms, and on the library balcony you'll find rules to all the games. There's a classroom where you can take vocabulary practice in any of three languages, do math exercises, play a geographical word search, or check your spelling.

You can look at star charts in the observatory—a broad-scoped room that also offers you a monthly horoscope reading that's less detailed than Sydney Omarr, and a biorhythm that's probably accurate.

In the kitchen, you'll find recipes to peruse and use, and you can add some to the collection, if you like. Next door, the greenhouse offers tips on raising happy, healthy potted plants.

Visit the darkroom for advice about photography, and leave questions or suggestions of your own for others.

The main piece of furniture in the bedroom is—not a bed—but an alarm clock, through which you can arrange to have the gamemaster signal you when a member arrives, wanting to play the game you've been waiting for, or wake you when you want to get up.

With the appropriate driver, you can listen to music and compose some of your own in the conservatory. The gallery offers similar pleasures in the realm of art and graphics.

Getting Down to the Business of Fun. But the backbone of GameMaster is, of course, the games. Eight rooms house games, divided by category. There's the arcade for quick-point games; club room for cards; board room for backgammon, chess, and the like; state room for games of politics, diplomacy, and strategy; war room for war games; and the locker room for a super baseball game and football. Then there's the time room, for games of the future—a special and fascinating category—and the engine room, a sleeper among the game rooms that offers one of the mansion's most popular multiplayer games—*Twelve Wheeler*—a trucking game in which the goal is to profit from transporting goods from one city to another.

BY MARGOT COMSTOCK TOMMERVIK

Prizes, usually in the form of system user credits, are given for high scores over a period of time in some games, and extra credits are often hidden in the house.

Each game room has an alcove to which you repair to find opponents and from which you enter the playing area. Members can converse in the alcoves.

But no alcove precedes the cold stone steps leading down into the chamber. Here, below the game rooms, deep below the mansion, in twisted, dank hallways, lie the catacombs, caverns, and gamemaster's den. Each of these is a separate and unique adventure game, although the catacombs and caverns tie in with each other; to gain entrance to the caverns, you must find a key in the catacombs.

Hi-res graphics drivers are available for both catacombs and caverns. The catacombs are a giant maze of convoluted corridors—a great place to learn mapping. The caverns are not as twisty and turny, but they're packed with wonderfully vicious monsters who guard finely-detailed, hi-res treasure. The gamemaster reports at least one member who has spent sixty hours in the caverns—all via modem.

A Future Where Apples Run Errands. Many of the services provided enable you to avoid running errands or doing tedious chores. (Unfortunately, the module that provides for a maid and butler to step through your monitor and clean your house isn't running yet.)

Other services in operation, in the works, or in negotiation include a car pool bulletin board in the garage, complete with up-to-the-second local traffic reports; do-it-yourself hotel and Ticketron-type reservation services; instant weather news; and general late news with thorough coverage of the computer world.

Some rooms are much less popular than others, and some have fallen prey to governmental interference. The banquet room, for example, was intended for exchanging dining-out information. But when GameMaster asked the local restaurant association for a base restaurant list—which the GM staff felt was essential to getting the room started properly—the request was refused, so the room may be abandoned for the present.

Making Friends by Modem. Perhaps the most unique rooms in the mansion are those in which members can visit each other. The parlor is open territory. Anyone may enter the parlor anytime and, if others are present, they may converse. A typeahead buffer makes it possible for one person to be heard while others are typing in their comments, which will be heard in turn. This works out to be quite conversational, although talking on the offbeat takes some getting used to.

The three conference rooms require permission to enter when conferences are in session. Members may reserve conference rooms for actual private meetings, or they may announce a topic and invite others to join in. Unlike the parlor conversation, dialogue in the conference rooms is private.

Popular also is the mailroom—and it's real. Regular members get a private numbered mailbox, complete with right-left-right combination, with membership.

The People Who Make It Happen. "I've been a game player forever," says chief gamemaster and founder Harlow Stevens. Even as a child, it wasn't just play he enjoyed, but the strategy, intricacy, and infinite variety of gaming.

Stevens studied electronics in high school, and it isn't surprising that this interest, combined with his love of gaming, led him to work in a computer store after that. It was at a college fraternity reunion that the idea for GameMaster was conceived.

Stevens, Bob Kniskern, who was then a television repairman, and Paul Martin had all become fans of microcomputers. At the reunion, they agreed that they all wanted to make games for the com-

puter, but they were competitors as well as gamers, and they enjoyed interaction with people as well as with computers. So they weren't satisfied with the play format possible on the computer. Good games wouldn't work for two; players would have to turn away during the other players' turns.

"So we had to make a gaming network," says Stevens. With several monitors involved and players at least a building—if not a continent—apart, players can take their turns simultaneously and secretly, with only the outcomes shown.

"The rooms and the house came from needing a medium for the games."

Thus was GameMaster born.

Willingness To Work Overcame Setbacks. After months of development, Stevens, Kniskern, and Martin were ready to unveil their brainchild. In the very week they opened for business, another announcement excited the personal computer world: it was called The Source. It offered many of the services GameMaster planned, without the fun. But The Source predominated, took the thunder, accepted the kudos and sales, and promptly proved unable to keep up with demand. (The Source, too, has since overcome these problems.) Disgruntled Source customers wrote the service off because they couldn't get through to use it; the phone was always busy.

But GameMaster had already been lost in the crowd.

It wasn't until a writer for *Interface Age* happened to hear of GameMaster and wrote it up for the magazine that business began to pick up and Stevens could justify devoting full time to the service. And full time to Stevens means just that: "I spend thirteen to eighteen hours a day, seven days a week, here," he says.

"Here" is a modest office in the Chicago suburb of Evanston, Illinois, housing the 272K Alpha Micro on which the mansion runs. By its side sits a happy Apple running a separate GameMaster ABBS. The Apple performs two other important functions. Around the clock, it's the mansion's best advertisement; 15 percent of the GameMaster members hear of the mansion through chance calls to the bulletin board.

The Apple's more important function—although it's seldom called upon to perform it—is to back up the Alpha. Should the GameMaster system go down when Stevens isn't nearby, GM staff in Fort Wayne, Indiana, can telephone the Apple and have it reset the system.

Back Home in Indiana. And Fort Wayne is precisely where most of the GameMaster staff hangs out—most notably cofounder Paul Martin. Martin, an optimistic, outgoing person who draws out the sunshine in others ("You could talk to a dead man, and he'd talk back," his dad once told him), is an Apple owner and enthusiast. He and the Fort Wayne staff members still work only part time on GameMaster, doing development and programming on Apples that simulate the Alpha Micro.

Although he helped set up GameMaster as a game network, Martin doesn't see games as the main purpose of the system.

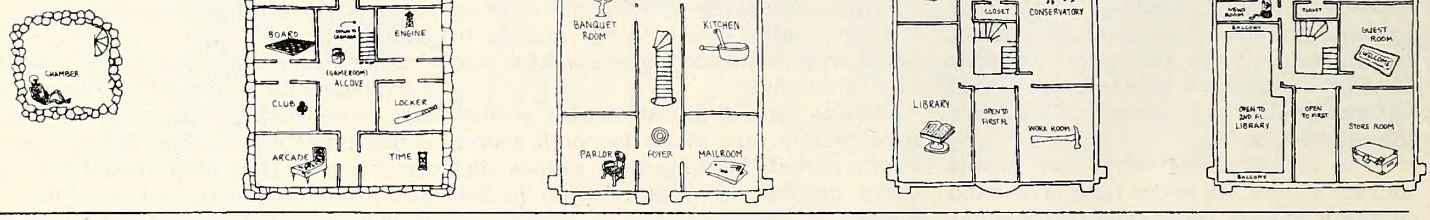
"The mansion isn't a house; it's a philosophy. It's intended to appeal to—and it does—people who understand where the world is going and are helping it get there." Martin would like to see the network become a medium for the exchange of ideas and developing programs. "These are the people who can take the full power of the micro and use it to improve the world."

Stevens adds to these thoughts: "One of our major functions is making the computer humanistic and easy to use. A key to this is that we don't take ourselves too seriously, and we have fun."

Certainly the GameMaster membership has fun. Fifteen to twenty members visit the mansion each day on the eight phone lines. The favorite rooms of the visitors are the games, the mailroom, and, happily, the parlor.

Among GameMaster members, the youngest is thirteen, the eldest in her sixties. Right now there are more members in their teens and twenties than in any other age range, but the thirties and forties crowd is catching up.

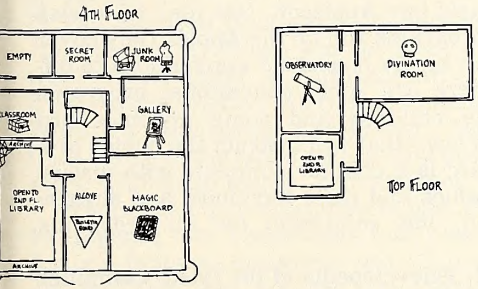
Apples Abound. And, although GameMaster will work with any microcom-



puter with a compatible modem, 70 percent of the members have Apples. Besides the fact that the mansion was designed with the Apple in mind, Stevens accounts for the profusion of Apples in a couple of ways. People who scrimp and save to buy a computer often end up with TRS-80s or Atari 400s because of the low prices, he says, and, once they've got the computer, they've spent all they can. And those who can afford peripherals usually don't choose a modem. "If TRS-80 had a modem that went with it as nicely as the Micromodem goes with Apples," Stevens suggests, "more would get one."

Stevens expects an influx of new members from the sales of the new IBM and the Osborne microcomputers. Those two are designed for business, and software for entertainment is nil. Owners and users will turn to GameMaster for games.

With systems like Sprint springing up and cutting the cost of long distance telephoning, more people are able to join GameMaster from outside the Chicago area. Even so, GameMaster's own plans for a network that serves local needs are spurring Stevens, Kniskern, and Martin toward opening local GameMasters in other cities. The most convenient first step in that direction is to set up a second system in Fort Wayne because of the immediate accessibility of staff; that one should open in the next couple of months.



By the middle of 1982, the company hopes to have built mansions in six or seven midwest cities. Whether to franchise, lease, or merely hire staff for the new locations is still a question.

Imagination and a New Face Move GM Forward. Meanwhile, GameMaster is not sitting still. The hi-res drivers are less than a month old, and a music driver for the conservatory is brand-new. More hi-res will be forthcoming from the newest face at GameMaster, talented graphics artist John Grosjean. In addition, the messenger driver for the bedroom will soon enable members to sign up for games, leave a schedule of times when their systems will be up, and have the big clock set off alarms in their Apples when competitors are ready to play. The same driver will make it possible to write letters off-line and drop them in the system later.

Game lover Stevens expects this facility to offset a sort of problem. "The single most frustrating aspect of this is getting people to play with other people. People on GameMaster act just the way they might at a party: extroverts will join in; introverts will not." But Stevens doesn't think it has to be that way, so he's instituted a number of programs to encourage everyone to participate.

Among these is the happy hour; from four to six each afternoon, rates are halved in the parlor. Tournaments are another kind of mixer, such as a backgammon tournament. And, what is probably most effective of all, Harlow Stevens is almost always there himself for members to play games with or visit with.

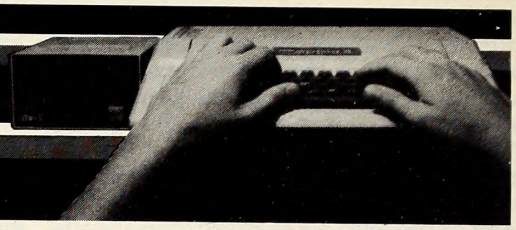
That the ideas and innovations are working and that people are beginning to think of the gamemaster's mansion as a comfortable, fun place is evident in a new phenomenon. Sometimes members don't have time to play a game, says Stevens, but "they'll stop in just for a few minutes to chat."

Isn't that what makes a house a home? ◨

How You Can Join GameMaster. *For the $65 full membership in GameMaster, you get 360 credits (each credit is a minute of house time); a mailbox for five months; and a GameMaster notebook with maps, descriptions, and game rules. A la carte membership costs $10; then time, mailbox, and documentation are extra. If you spend more than $65 as an a la carte member, you are automatically made a full member with all attendant rights and privileges.*

MARKETALK

Impressions



from page 64

create mountains and valleys of varying slope by using a paddle to designate terrain height. You generate twenty-one frames of terrain, which are then saved to the disk for later use. Make terrain as difficult as you want. High mountains and low valleys give more protection to enemy installations. Not only that—now you can't avoid the deadly eagles and saucers and have to maneuver as well as shoot, drop, and dodge. If you're a closet jet fighter pilot, Pegasus will get you off the ground. $29.95.

☐ **Word Handler.** By Leonard Elekman, Silicon Valley Software (Redwood City, CA). Convenient for editors, a dream come true for writers, and useful for anybody who wants an easy-to-use word processing program. One of the most appealing features of the program is that it starts its user writing immediately. Silicon Valley Software has come out with a manual so simple that it promises "to have you capable of doing useful work with the *Word Handler* in ten minutes." The manual is written in a straightforward style, with the information of first importance taking up just a few pages in step-by-step simplified text. People impatient with manuals might take as long as eleven minutes to wade through it.

Word Handler has some nice features: punching a few keys will change the text into bold face, underline, or produce superscript; hitting control N (for normal) will return it to regular text. Single words, individual lines, or entire pages can be deleted almost instantly with other simple commands. Text can be copied from anywhere in the document, saved internally until needed, and then placed at the position of the cursor by making two more commands. Blocks of text may be moved from one place to another by using a combined form of the copy command. *Word Handler* can also be used for printing newsletters or making a very professional-looking book. It can be set up to allow folded paper printout by making mirror image margins (two-sided printing).

The program uses a hi-res character display with sixty-six characters on a line. Unfortunately, the letters M and W on two different video monitor screens came out as little more than a white blob; the M is really bad. You can distinguish between these two letters, however; so depending on how fussy you are, it might not make much of a difference; and, of course, it prints out clearly. *Word Handler* is a writer's dream—almost. DOS 3.2 or 3.3. $249.00.

☐ **Universal Boot Initializer.** S & H Software (Manvel, ND). If your Apple has a Language System or ROMCard, you must boot your system master or your Basics disk to use the alternative Basic language. No big deal, you say; but it is a bit of a bore to wait for it to load. S & H Software has the solution.

A disk initialized with the Universal Boot Initializer boots apparently as fast as any disk that's running only a simple Hello program in the process. But, in that quick boot, it loads the second Basic.

So if there's an Integer game you love playing on your Plus (or vice versa), you can init a disk with *UBI* and transfer the game onto it; you'll never have to boot your master to run the Integer game again. If the program is 3.2 as well, and you have 3.3, never fear. *UBI* gives you the choice of DOSs.

With *UBI*, you can initialize disks in DOS 3.3 or 3.2, in Integer or Applesoft, and with either or both languages ready to load instantly on the disk.

Nor need you worry about the quality of the DOS with which your disks are being initialized. S & H has licensed DOS 3.2, DOS 3.3, IntBasic and FPBasic, and the update programs directly from Apple, so you're getting the real thing.

The secret of the speed is S & H's own addition to these standard Apple programs: Quick-Boot. It does what it says, and, if you've room on a disk for it, it's well worth putting there. And, if you do nothing else with it, you'll want to keep a *UBI* initted disk by your Apple to use in place of your master when you need your alternative language. $40.

☐ **Apple Pascal: A Hands-On Approach.** By Arthur Luehrmann and Herbert Peckham, McGraw-Hill (New York, NY). Aside from the manuals that come with the language system—which are less helpful to the Pascal beginner than one might wish them to be—this is the first Pascal text that deals specifically with that language's implementation on the Apple. It's a bright, witty, admirably lucid text that teaches not only Pascal and the Apple Pascal operating system, but programming as well. As its subtitle suggests, the book requires attendance at an Apple. Luehrmann and Peckham start the reader off with simple procedures that do interesting things, moving quickly from elementary Writeln statements to simple graphics and sound generation routines. An attitude of experimentation is encouraged throughout. This is a refreshing text. $14.95.

☐ **32 Basic Programs for the Apple Computer.** By Tom Rugg and Phil Feldman, dilithium Press (Beaverton, OR). If you like to learn from other people's programs, but you're tired of translating Brand-X Basic into Applesoft, this may be a book for you. All thirty-two programs run on the Apple. They cover a range of themes, from handy household applications to differential equations. There are some educational programs suitable for school-age children and some graphics programs—of high res and low—that will instruct the novice programmer. Each program is a chapter complete with discussion, summary of variables and their functions, and suggestions for customization. 16K minimum for all programs. $17.95.

☐ **The Basic Handbook: Encyclopedia of the Basic Computer Language.** By David A. Lien, Compusoft Publishing (San Diego, CA). This is essentially a dictionary and thesaurus of Basic—a handy tool to have if you need to wrestle with program listings in foreign dialects. Basic-speaking computers from four continents are covered in a format that includes definitions, synonyms, illustrative test programs, alternate spellings, variations in spelling, and cross references to related commands. $19.95.

☐ **Bez Wars.** By John Besnard, Bez (Irvine, CA). Yup. It's another Bez game. This one is a chaotic free-for-all where two players fire lasers from mobile missile launchers at opposite ends of the screen. A warlike blend of football and *Space Invaders* seems to have inspired *Bez Wars*. You need two to play this game. A paddle moves your rocket launcher across the bottom of the screen while your opponent moves across the top. Each player has a small squad of what we shall call Bez soldiers, since football players don't have to defend themselves against rockets.

With the two armies of Bez soldiers squared off against each other, players try to knock one another off. Ideally you want to hit your opponent's rocket launcher, which is protected by Bez soldiers who march back and forth across the screen. Hitting the soldiers wipes them out and gives you small points.

To keep the game spacy, every now and then a flying

saucer hovers across the fifty-yard line, annihilating any Bez soldiers in its way. It's a race to see who will shoot the saucer for bonus points.

It's hard to tell if it's better to fire continuously at random or aim carefully. A wild flurry of rockets will kill your own Bez men but is more likely to land a fatal blow on your opponent's rocket launcher.

Skill doesn't seem to make much difference with *Bez Wars.* It's a perfect game for a giddy mood when a chaotic little war will help. You'll laugh trying to figure out how you won or lost. Children especially will enjoy this game. $26.95.

☐ **The Cube.** By Patricia Shanahan, Midkemia Press (San Diego, CA). Make no mistake, the cube is complicated. You start out with a large cube composed of twenty-seven smaller blocks. Each of the six faces of the large cube is a different color—purple, orange, blue, black, white or green. The colors are actually on the face of the smaller blocks. You can rotate any one face of the large cube in a clockwise direction, which will mix up the color pattern of the blocks, so that with a few moves your large cube will look like a patchwork quilt. The object, then, is to restore your cube to order, with any one of its six faces the same color.

If you've seen Rubik's Cube, you'll know this game. The Rubik's Cube itself could take months to solve; in fact, when a paperback book appeared with a formula for solving Rubik's secret, people rushed to snatch the book from the bestseller racks.

The computer version requires a stronger imagination, not only because you cannot physically manipulate the blocks, but because any move you make rotates the position of nine blocks. Have fun pulling your hair out. The cube is for devotees of hard-core brain teasers.

Who knows? Patricia Shanahan may live to see someone write a program to solve her puzzle, if she doesn't beat them to it. $19.95.

☐ **The Wurst of Huntington Computing.** By Fred Huntington (Corcoran, CA). When was the last time you took a pickyness test? How about a compatibility test with your best friend/worst enemy? All this and much more is possible in the *Wurst of Huntington Computing.* The Wurst supplies just that—some of the worst jokes you'll ever see on a disk, but also some whimsical satire—provided you can laugh at yourself.

There are eight categories to pick from: openness test, unintelligence test, chauvinism test, observation test, nighttime fortune fantasy, pickyness test, Chinese horoscope, and the compatibility test. You are then instructed to pick a number for the program you want and leave the room before it comes up. If you don't make it in time, you may have to take the pickyness test. Here you rate yourself on a series of items such as, "When I butter toast it must be buttered all the way to the edges evenly," and "Do you feel it is important to dust the tops of books at least twice a year in your library?"

If your party needs a good conversation starter, or you'd like to analyze your marriage or test your powers of observation, the *Wurst of Huntington Computing* might be for you. At its very best it'll provide some good laughs; at its very wurst you can always leave the room. $19.95.

☐ **Quick-Text-Editor.** By Seaport Software (San Diego, CA). Since *Quick-Text-Editor* is an editing program only and not a word processor, it's well worth its $24.95 price. But if you try to make it do more than it's capable of doing, you'll probably wish you had forked out an extra hundred bucks for an elaborate word-processing program. *Quick-Text-Editor* purports to be an editing program only, and it does some functions rather well. A delete line command allows you to remove lines of text from your current file in two different ways: specified can be used to delete a known range such as lines 12 to 32; the global option allows you to step through the entire file, starting at any line number, and delete or retain each line—one at a time. Other functions include fast search and replace and configuration for either forty-column or eighty-column display. *QTE* has most of the standard functions needed to handle most text writing and editing needs. $24.95.

☐ **Galaxy Gates,** by Eric Popejoy, Magna Soft (Los Angeles, CA). *Galaxy Gates* is a nifty entry for a new company—Magna Soft. One of an arcade series, this one involves your ship trying to blast through three concentric rings to kill an enemy at the center of the circles.

Using a paddle to steer your ship and either paddle buttons or the keyboard to fire, this is a fast moving, well-executed game that's faithful to the genre.

Adding to the involvement are enemy sorties out from the rings as well as a force field that won't permit you to sit in next to the circles and fire away.

Dexterity and wit, rather than brute force, are needed to stay alive.

A nice touch is the recording in memory of the top ten scores achieved at a sitting. Alas, they're not written to disk, so you'll never be able to prove to Uncle Josh in Joplin how good you've gotten. $24.95.

☐ **Planet Protector,** by Eric Popejoy, Magna Soft, Los Angeles, CA. *Planet Protector* is another in a series of arcade games from Magna Soft.

In this one, the player takes control of three gun emplacements and attempts to shoot missiles out of the sky to protect his six cities. This version has the same player constraint as a previous version, in that your gun emplacements have limited ammunition—so you'd best use it wisely.

Play is with a joystick for aiming and the f-g-h keys on the keyboard for triggering your weapons.

As with other Magna Soft product, it's a day late, being at least the third similar program to hit the Apple market. The quality of programming, animation, and graphics deserves to be in a game that's first on the market. $34.95.

☐ **Cross Clues.** Science Research Associates (Chicago, IL). In *Cross Clues,* two players, umpired by a computer, compete to uncover an interlocking matrix of hidden words. Requiring skill, intellectual ability, and luck, *Cross Clues* claims fifty different puzzles (of which *Softalk*'s review disk contained three) and three time allowance alternatives, so youngsters and parents can play together. In the fast mode, thinking and acting pay off. A fun twist: if the game thinks you're too good, it cuts your time allowance in half! For teens and adults. Applesoft. $29.95.

Alien Lander. By Alick Dziabczenko. The object of this simulation is to bring your ship to a soft landing from twenty thousand meters, with enough fuel remaining to get off the ground and back up to that altitude. The trick isn't to steer away from mountain peaks, as on some other landing simulations, but to control thrust correctly, given a limited amount of fuel. You can vary the problem with three parameters: gravity, thrust, and atmosphere. A gravity setting of one with zero atmosphere would simulate a lunar landing; six and six would represent terrestrial conditions. If you choose to have an atmosphere, you have to make sure you don't penetrate it too quickly, or your ship will overheat and explode. The higher you set the thrust the more control you have; at any setting your engines can fire for a maximum of ninety seconds. You control the engines with either the paddles or the number keys.

The program offers a rather unimpressive hi-res display of stars as you begin your descent; a three-dimensional, mountainous landscape rises to meet you as you approach the planet. Beneath this window you see a digital readout of your altitude, speed, elapsed flight time, and fuel supply. At any time, you can toggle from starscape or landscape to a printout from the onboard computer, which offers status information and advice.

To achieve a fully successful landing, you have to get your speed down to under thirty kilometers per hour on touchdown. To accomplish that efficiently, without wasting fuel that you will need for the return flight, requires split-second accuracy with the paddle or keyboard. $24.95. ◨

SEND YOUR APPLE ON A CRUISE

And maybe it'll let you come along.

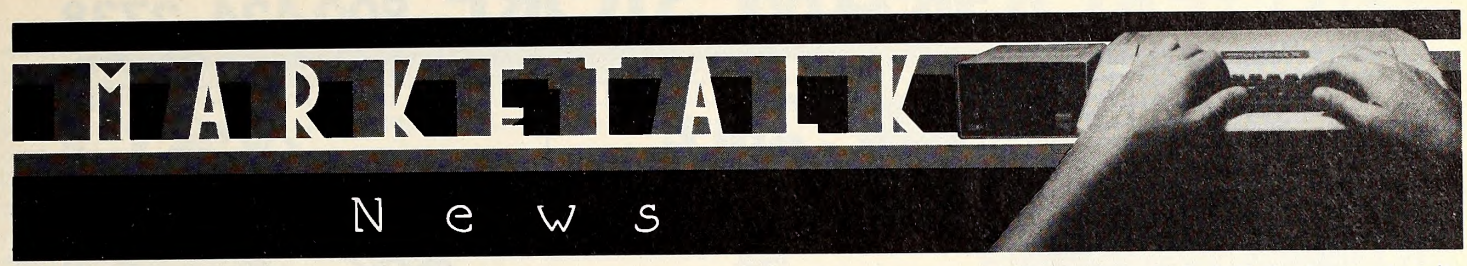
On June 5, 1982, a luxurious cruise ship will depart from Vancouver, Canada, with several Apples on board. For seven days, these privileged Apples will be learning such arcane arts as assembly language from Roger Wagner and graphics from Ken Williams, as well as compiling Applesoft programs into assembly code with Dennis Goodrow.

Besides coming back more intelligent, the Apples will be treated to the usual shipboard conveniences by a professional staff dedicated to providing everything a microprocessor might desire. The Apples will enjoy some of the most dramatic scenery north of Silicon Gulch, stopping in Ketchikan, Juneau, and Skagway, some of the last frontier communities. They'll cruise past Glacier Bay—giving the Apples a look at a true hi-res graphic.

Amazingly, in keeping with Softalk's policies, the Apples will be taking the tour free. Apples craving the companionship of their owners may inquire about the cost of human participation by writing to:

Softalk Cruise
Attention: Donna Siebert
11021 Magnolia Boulevard
North Hollywood, CA 91601

This tour is exclusively arranged for Softalk readers by Valencia Plaza Travel Agency of Newhall, California. Roger Wagner, Ken Williams, and Dennis Goodrow are committed to giving seminars on this cruise unless extraordinary business contingencies arise.



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

☐ According to recent neurolinguistic research, kids learn to spell and do arithmetic by visualizing the letter or number combinations involved. Such findings, which can help us improve learning processes, have been applied in two forthcoming programs from **Behavioral Engineering** (Scotts Valley, CA) called *Spelling Strategy* and *Math Strategy*. Both teach the use of visualization to improve verbal and mathematical skills. Color TV or color monitor highly recommended. Each under $50 and available soon through **Apple Computer**'s Special Delivery Software.

☐ **Broderbund Software** (San Rafael, CA) calls its new game *Genetic Drift* a combination of scathing social commentary and nerve-racking drama, with finger-mutilating action. Also new from Broderbund is *Space Quarks*, with dancing monsters in an animated arcade research expedition. $29.95 each.

☐ If you're a self-employed professional or run a small business at home, *Easy Ledger* by **Highlands Computer Service** (Renton, WA) offers the simplest form of accounting—single entry bookkeeping under either expense or income. *Easy Ledger* keeps track of expense items under any one of ninety-nine user selected headings and keeps a year-to-date, monthly, and type running total. Supports 80-column or 132-column printouts and one or two disk drives; entirely menu driven. $55. Highlands also has a new advanced adventure called *Goblins*, in which you venture into Goblin country to secure treasures without being killed by said goblins—not to mention ogres, elves, or ghosts. You'll navigate submarines, climb mountains, explore swamps, and duck flying boomerangs. Requires quick mind, bold heart. $27.50.

☐ People with diabetes and others who require similar diets may now find help in following the food exchange diets of the American Diabetes Association with *Diet Manager*, a program for planning all meals and snacks to fit user's diet needs. 32K. $24.95; by mail, $25.95 from: Daniel Tobias, 7 Broadview Road, Poughkeepsie, NY 12603.

☐ *Copts and Robbers*, another fast-action game from **Sirius Software**, entombs you in an Egyptian pyramid. The only way out is to find four jewels and a statue and take them to the vault room. Beginners' maze and two advanced ones contain treasure-guarding mummies and a ghost who likes to take things and hide them. $34.95.

☐ *CONST* from **Computing Interface** (Redondo Beach, CA) is a program for those involved in construction estimating. It asks you questions concerning sizes and materials; makes allowances for door and window openings; and calculates quantities needed and exact sizes of structural members. These figures can then be used with the latest material prices to give exact construction costs. Program can also help make design changes or material substitutions once the original program output has been studied. Author: David Lovejoy. $75.

☐ **Rosen Grandon Associates** (Tampa, FL) has released an enhanced statistical system for the Apple II: *A-STAT 79.6*. In addition to mainframe-type data processing techniques and multiple file handling, *A-STAT 79.6* interfaces to the *Apple Plot* hi-res graphics program as well as the *File Cabinet* data base program and can read *VisiCalc* files. Other features include: simplified command language, compatibility with older versions, added capabilities in the bivariate tables program, added residual analysis, file sorting, and statistical report writing. New version, $145; upgrade of older model, $25.

☐ An expansion module is now available for your *Temple of Apshai* game from **Automated Simulations** (Mountain View, CA); it's called *Upper Reaches of Apshai*. *Hellfire Warrior* has a module, too, called *The Keys of Acheron*. Add new adventures, monsters, and treasures; generate a new character or take along a veteran from previous adventures. Modules are not stand-alone games. $19.95.

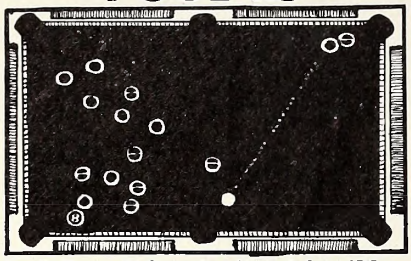
☐ It's you against the medfly in *Sterl* from **Creative Computing** (Morris Plains, NJ), a simulation game in which two pest control methods—pesticide and release of sterile male flies—are used to attempt total eradication. *Sterl* is one of four games in the *Ecology Simulations I* package. $24.95 for all four.

☐ *Architect's Business Manager* from **Concept Group** (El Paso, TX) is a comprehensive systems package that provides management information for architectural firms with up to 250 employees and as many as 134 active clients. Subsystems include job cost, payroll, general ledger, and accounts payable. $2,000. Also useful for architects or engineers is Concept Group's *Word Star*, a document-oriented professional word processor that allows you to make your own glossary of architectural terms and check your spelling automatically. Both require eighty-column board and SoftCard. $495. And, finally Concept Group helps architects stay abreast of technology with the *Masterspec 2* Basic and SLV disk library. To become

THE PROGRAM STORE

CALL TOLL FREE 800 424-2738

POOL 1.5



By Hoffman, St Germain & Morock from IDS
The pressure is on: if you can just get enough english on the ball to bank it into the corner pocket . . . In POOL 1.5, you can! A remarkable action-simulation of the real thing, this program allows full control of your "cue-stick" for aim (265 directions) and control (all types of english). Play four different types of pool at your choice of table speed, and even get an "instant replay" of any shot, in slow motion! Hi-Res color graphics are used throughout this real-time game.

48K Disk...$34.95

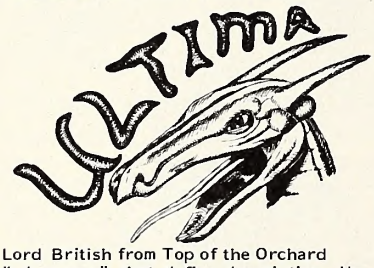
DEATH-MAZE 5000

From Med Systems
A new breed of adventuring! Venture through a graphically represented 3-D maze, with halls that could dead end -- or recede to infinity. Step through the doors or drop into the pits. Will you encounter monsters and mayhem, or will you be treated to useful objects and information? Will you ever get out alive?

You may never find your way out of Deathmaze 5000, but you'll keep trying!

32K APPLE II, 16K TRS-80...$12.95

ULTIMA



By Lord British from Top of the Orchard
An "adventure" that defies description. Unlike the text-type adventures, ULTIMA allows you to wander through towns, countries and continents using Hi-Res graphics and set-commands. And unlike the usual adventures, your journey spans not only space but time, as well. In ULTIMA, you start in the dungeons-and-dragons era but may progress through the space age and beyond! Whether you are "into" adventures, fantasy role-playing, or arcade games, you won't want to miss playing ULTIMA!

Supplied on two disks, one protected (only one disk drive required). 48K...$39.95

APPLE®

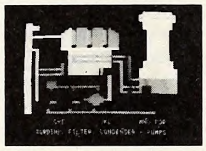
For Apple II & Apple II Plus
Applesoft, 48K Disk
unless otherwise noted

HI-RES SOCCER

By Jay Sullivan from On-Line
Here's a game that let's you really get your kicks! A real-time Soccer game that offers the excitement and challenge of the real thing. You manipulate your eight fully-animated characters against a friend or the computer. Move the ball down field to scoring position, but don't get tackled or your opponent may score against you.

Hi-Res graphics give the kicking, throwing, and corner kicks an absorbing realism, aided by the game clock, scoreboard, and sound effects. Three skill levels allow challenge for beginner through expert.

48K Disk...$29.95



THREE MILE ISLAND

(SPECIAL EDITION)

By Richard Orban from Muse
New machine language version lets you decide whether or not nuclear technology is too complex to handle. The comprehensive documentation describes in detail the operation of the pressurized reactor illustrated in Hi-Res graphics. You must supply electricity -- profitably -- or lose your license to operate. But sloppy operation or pushing too hard may cause a radiation leak . . . or worse!

48K Disk...$39.95

SABOTAGE

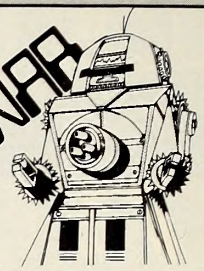
From On-Line Systems
As commander of an anti-aircraft base, your mission is to clear the skies of enemy planes and helicopters. The opposing forces have other plans. While keeping you busy with a firestorm of bombs, they are dropping paratroopers to sabotage your base!

Quick, machine language animation in Hi-Res color graphics. You can fire conventional or controlled weapons one shot at a time or in rapid sequence. And with auto skill-level escalation, the better you get, the tougher "they" get!

48K Disk...$24.95

THE PROGRAM STORE
Franchises Available

ROBOTWAR



From Muse Software
The thinking person's fast action game. A contradiction? No, RobotWar combines your forethought, programming, and logic to create and condition a robot that will take part in a fast, futuristic gladiator battle!

RobotWar will provide fun and challenge while honing your programming skills. Program your robot in special "Battle Language," debug it on the cybernetic "test bench," and finally watch as your efforts win or lose against up to four competitors on the battle field.

48K Disk...$39.95

ORBITRON

By Eric Knopp from Sirius
Your space station is in a stable orbit, high-energy force fields functioning normally. But what's this? One-by-one, killer satellites begin orbiting your station, preparing to take out your rotating force field -- and you! Fight them off with your weaponry, but don't lose track of the fast-moving meteors. They may be on a collision course.

Sound and fast graphics make ORBITRON a treat; the seven levels of difficulty and bonus point scoring add to the challenge.

48K Disk...$29.95

By Bill Budge from Stoneware
Hi-Res, arcade-type lunar lander offering great fun and a real challenge. As you bring your LEM down, you control it through 360 degrees of rotation. Move horizontally and the moonscape "scrolls" aside beneath you. You pick your site carefully with the aid of the closeup view and try to set your craft down gently, because if you come in too "hot" you're in for a spectacular crash!

48K Disk...$24.95

SANTA PARAVIA

From Instant Software
A classic simulation of government, based in the year 1400 AD. You can play by yourself or compete with up to 6 players as you assume the role of rulers of neighboring Italian city-states. Most of the factors that go into government -- economics, politics, social issues, etc. -- are simulated in this program.

48K disk...$19.95

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.

Prices Subject to Change

THE PROGRAM STORE
4200 Wisconsin Avenue NW, Dept. OA10 Box 9609
Washington, D.C. 20016

a subscriber and receive each quarterly update, you must be licensed by the American Institute of Architects. Fee for first year is $550; renewal each year is $175. Fifty disks included. For more details, contact George Staten, AIA president at Concept Group.

☐ **WIDL Video** (Chicago, IL), publisher of Apple directories, has just released its second edition of the *Apple II Resource Directory*, a where-to-find-it book of hardware, boards, peripherals, and accessories. Product information, photos, prices, and descriptions are included, as well as lists of Apple reference manuals, publications, user groups, and a special section of computer bulletin boards across the United States accessible from your Apple. $5.95.

☐ Electrical interference can damage your Apple or disrupt program operation. The *Model ISO-3 Super Isolator* from **ESP (Electronic Specialists**, Natick, MA) is built to control interference caused by nearby power lines, lightning, heavy machinery—even your own disk or printer. Designed for a 1,875-watt load. Each socket can handle a 1,000-watt load. $94.95.

☐ **American Avicultural Art and Science** (St. Louis, MO) has a set of two programs for detail engineers, physics students, and scientific field engineers: *Moment of Inertia*, with fifty-six formulas for twenty-two bodies of mass, as well as ability to calculate dimension, choice of mass, or inertia on selected axes; and *Element of Triangle*, which contains three major triangles and twenty-three formulas. Also calculates sides, angles, altitude, area, and radius of inscribed circle simultaneously to find force and directions. Both programs on single disks, using touch-key selection input system. Instruction includes more than one hundred commonly used industrial materials. 36K with DOS 3.3 or 3.2. Two-program set, $40.

☐ Russ Walter's "Computer Blitz Weekends" will be offered this month, October 24 through 25, and again November 14 through 15 at **Creative Computing** in Morris Plains, New Jersey. For anyone interested in learning about computers, the workshops go from 9 a.m. to 7 p.m. Saturday and Sunday and cover these topics: beginning, intermediate, and advanced programming; equipment; computer industry overview; microcomputer applications; purchasing; computer languages; and careers in programming. Tape recorders welcome, as well as follow-up phone calls after the seminar. In preparation for the blitz, participants can purchase Walter's four-volume *Secret Guide to Computers* ($14.80), available from the Computer Education Center in Morris Plains. Take your lunch and learn through the lunch hour. Course fee $68, or $36 for either day. For more details, write to Barbara Garris at the Computer Education Center, or phone (201) 540-0445.

☐ *Lesson-Tutorgraphs* are classroom oriented software packages that present programmed lessons with full-color hi-res illustrations and animation. They allow students to review any text page independently, test themselves, and review missed test questions. First in the series of packages, now available from **Teach Yourself by Computer (TYC) Software** (Geneseo, NY), are *Weather Fronts* and *Shore Features* for high school or college students. *Weather Fronts* provides an introduction to weather front characteristics, while *Shore Features* acquaints students of geography, geology, or oceanography with seashore formations. Each lesson comes with teacher/student user manual. Both $24.95.

☐ **Andent** (Waukegan, IL), researchers in anthromedical technology, robotics, and applied cybernetics, offers *Prescription Form Writer* for busy doctors. Program prepares multiple preprinted prescription blanks, enters any drug, dosage, quantity, and instruction. Print out any number of forms. $20. Also *Dental Insurance Form Writer* prepares Universal ADA Insurance Claim Forms. Allows rapid billing and claim submittal. Master form can be created for each patient/family and saved for later use. Enter treatments, add dates, print insurance billing. Print as many copies as needed; each diskette will accommodate more than one hundred families. User definable; up to ten practitioners. DOS 3.2 or 3.3, eighty-column printer. $100.

☐ A new report for owners of retail computer stores from **Irv Brechner Enterprises** (Livingston, NJ). *Computer Store Advertising: A Complete Guidebook* offers advice on every aspect of advertising. Written in easy-to-understand language, it covers a variety of topics. $25.

☐ **McGraw-Hill** (New York, NY) will hold two different electronic communication workshops this November. A two-week course titled "Basic Data Communications" will be held at the Los Angeles Marriott Hotel, Los Angeles, November 9 through 20, 1981. Course will provide engineers, technicians, operations staff and marketing/administrative personnel with a background in data processing with a detailed understanding of all aspects of basic data communications.

☐ Another learning opportunity sponsored by **DataCommunications** magazine will be a two-day seminar titled, "The X.25 Packet Network Protocol." Topics will include concepts and terminology; physical level; link level; packet assembly/disassembly for nonpacket mode terminals; and upper layer protocols. Seminar will be held November 16 and 17, 1981, at the Adam's Mark Hotel in Houston, Texas.

☐ **Soft CTRL Systems** (West Milford, NJ) introduces *Utility ROM*. Program has five different functions including catalog directory sort, data disk create, program recovery, automatic line numbering, and list control. $39.95.

☐ Small computer systems won't be so vulnerable with *The Voltage Surge and Transient Suppressor* from **Cuesta Systems** (San Luis Obispo, CA). Electronically removes or greatly reduces sudden voltage changes that could affect the performance, or produce catastrophic failure of, sensitive electronic equipment. Plugs into an AC line power receptacle on the same fifteen-amp breaker circuit as the electronic equipment being protected. Solid-state semiconductors clip all overvoltage surges beyond 132 VAC, and a passive filter network snubs high frequency transients that might occur anywhere over the full input voltage waveform. A two-amp internal fuse provides safety overload protection. $29.95.

☐ *Accounting Plus II* from **Systems Plus** (Palo Alto, CA) is a general accounting package that offers general ledger, accounts receivable, accounts payable, and inventory with purchasing. Totally integrated and menu driven with user prompting for easy use. Supports two or three floppy drives or hard disk; reset protected to prevent loss of data, maintains up to 300 customers, 300 vendors, and 500 charts of accounts, on-line updating of data files. Inventory-purchasing module supports 1,000 items per volume, multiple volume by multiple disk, four different costing methods, maintains master parts list, prints purchase orders, and maintains on order quantities. Requires two or three disk drives, or Corvus. Integrated package of four available for $1,395. Choice of any three modules including general ledger, $995. Single modules, $425.

☐ **Brain Box** (New York, NY) announces the release of two hundred educational programs on thirty titles. The Brain Box *Courseware Kit* accompanies each title, permanently houses the disk or cassettes, and contains a glossary of commonly used computer terms, an overview of microcomputer hardware and functions, a primer for Brain Box programs, and a pronunciation guide to the Brain Box adaptation of the phonetic alphabet. Each lesson combines text with animated graphics plus sound effects. Available on disk and cassette, each lesson is 16K. Discount prices for multiple copies. Demonstration disk with a variety of lessons available for $30, which will be credited against the purchase price of any two or more titles.

☐ *SMART* (the Securities Market Analysis, Reporting, and Transaction system) from **Software Resources** (Cambridge, MA) is an integrated system designed for institutional money managers and investors. Provides capabilities for retrieving, storing, graphing, and analyzing securities and economic data. Portfolio module provides capabilities for maintaining and reporting data and transactions for securities portfolios. Users can automatically access data from one or more remote data

bases and prepare graphic displays in a variety of formats. Graphing options include open-high-low-close charts, volume histograms, line charts, and point plots. Two disk drives recommended. $2,150.

☐ Software to suit your appetite from **SoftTouch** (Costa Mesa, CA). *Recipe Handler* organizes and prints a shopping list of your favorite recipes. Recipe expansion of serving size adjusts each ingredient for quantity of people to be served. DOS 3.2 or 3.3. $39.95. Dealer inquiries invited.

☐ *World and State Capitals* from **American Avicultural Art & Science** (St. Louis, MO) consists of one hundred world nation capital test and a fifty U.S. state capital test. Designed to eliminate teacher supervision and encourage student spelling accuracy as well as knowledge of the United States and the world. Score results are retrievable for teacher's use; teacher can produce own test. Intended for a wide range of grades with four choices of questions, or for family use. $25.

☐ **Scholastic Inc.** (New York, NY) is offering nearly two hundred programs for the 1981-1982 school year for all grades and most subject areas including computer awareness. Detailed descriptions of all programs are included in the new 1981-1982 *Scholastic Microcomputer Software Catalog*, which is available by writing to company.

☐ *Compu-Law*, a law office management system for the Apple II from **Decisionmakers** (San Jose, CA), helps with critical calendar dates, accounts information, billing, and case load scheduling. Use with any eighty-column word processor. $2,500.

☐ **Creative Software Development** (West Valley City, UT) announces the *Time Machine II*, a real-time clock peripheral. You can select combinations of six date and seven time formats using software commands. Routines are interrupt driven and do not affect other Apple II functions. Basic system, with applications disk software, $135; applications disks #2 and #3, each $15.

☐ **C & H Video** (Hummelstown, PA), offers *the Menu*, a home menu planner/recipe retrieval system. Stores up to 399 recipes, changeable any time; up to 42 meals planned and written; shopping list printouts with ingredient quantities based on number of persons fed. Printer required. $29.95.

☐ "Speech Coding," a seminar on various synthesized voice techniques, will be presented by **DataCommunications** magazine in San Francisco on December 2-4, 1981, at the Jack Tar Hotel. Course will focus on basic elements of digital speech processing. Other topics include analog speech signals and their processing; analog/digital and digital/analog conversion of speech signals; special forms of Fournier Transform; and future voice applications. Registration fee is $650.

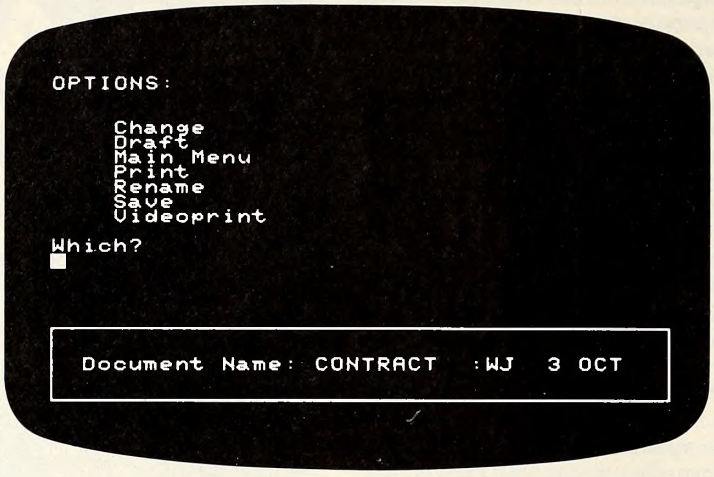
☐ The *Multiple Choice* from **CompuServe** (Columbus, OH) offers multiple choice educational tests, including an analogy quiz like those taken by graduate school applicants, a trivia test, and a personality profile test. Service provided at $5 per hour weekday evenings, weekends, and holidays. Weekday daytime access, too. Requires a modem and membership in CompuServe.

☐ Through the CompuServe Information Service, **Archer Commodities** (Chicago, IL) provides Apple II owners with current market reports, commentary, and futures trading information, as well as charts, newsletters, and quotation equipment. Fee of $5 per hour weekday evenings, weekends, and holidays. Weekday daytime access also available.

☐ Home data retrieval from your office computer twenty-four hours a day is possible with the *Auto-Cat* from **Novation, Inc.** (Tarzana, CA). It's an automatic answer direct-connect modem designed to communicate at 300 baud over dial-up telephone lines using a standard modular jack. Has three data modes—automatic answer, manual answer, and manual originate. $249.

☐ Apple II owners can eliminate the need for expensive printed stationery, business cards, invoices, mailing labels, etc., with the *Business Papers Kit* from **Solutions Softworks** (Roselle, IL), which allows them to print their own individualized stationery and business forms. $39.95.

zardax™



A Word Processor for the real world

When you buy a personal computer for your office, you don't want to take a college degree to use it. Computer Solutions has Word Processing software so good that it will make you want to buy an Apple II.

We call it **ZARDAX** — a wonderful writing tool for business use. Powerful and versatile, yet easy to use and natural.

You enter your text in free-form, inserting carriage returns only when you want definite paragraph breaks. Then watch while ZARDAX prints up your copy, breaking it into lines where appropriate. Change margin, pagewidth and pagelength with simple commands and print up a new copy in a different format.

With ZARDAX, printing 100 copies is easy. Inserting different specific information into each copy is easy. You can be in control of the format of your words and sentences.

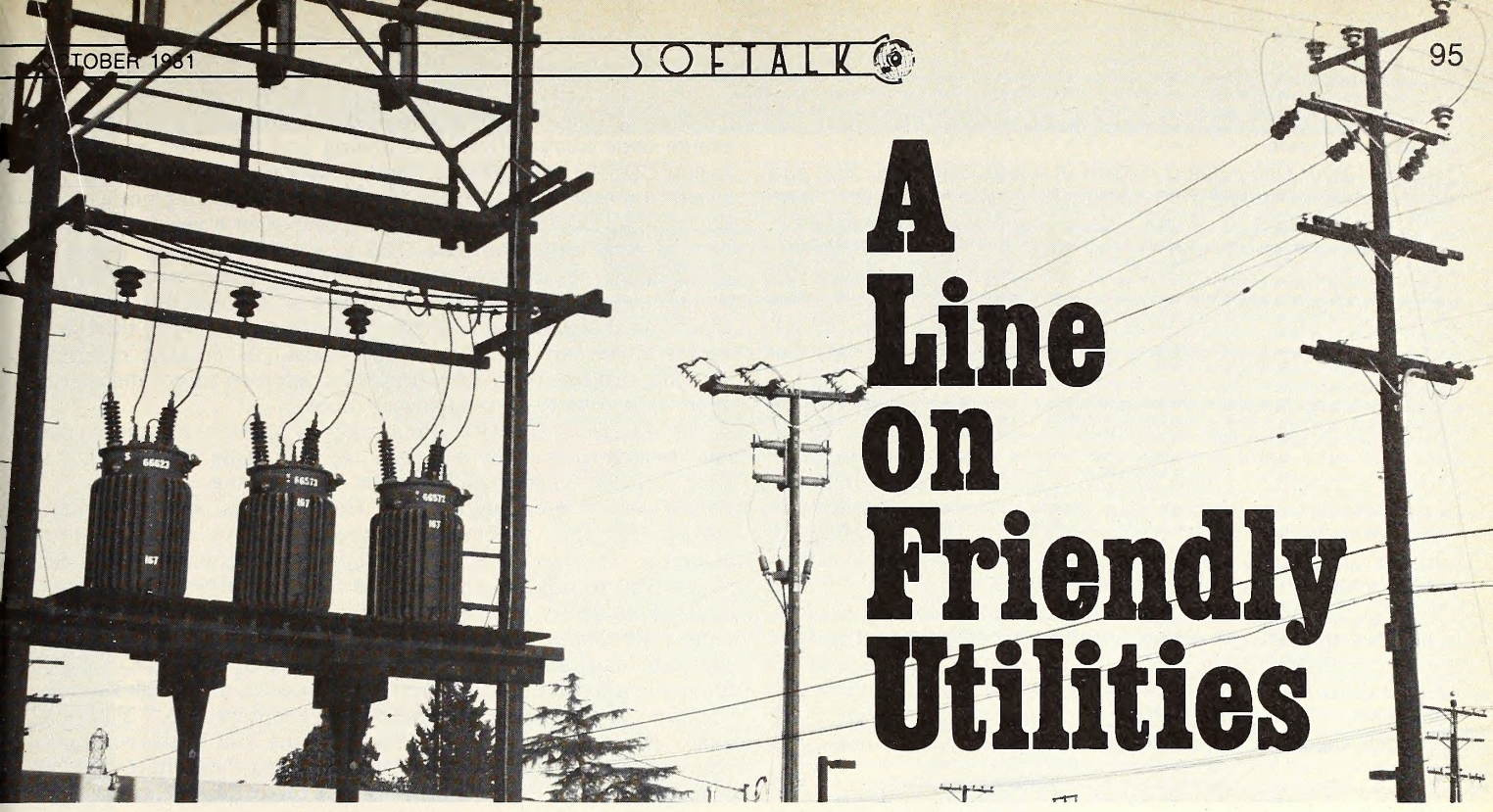
Fast recording of your documents on inexpensive magnetic diskettes. Retrieve your documents from diskette so easily, make some changes, and record the altered version. Then ZARDAX will print your new improved version. Never again will you retype a whole document just because you wanted a few changes.

requires a 48K, 16-sector disk-based Apple II Plus

Standard paragraphs recorded on diskette can be combined in different ways to make new documents. There are all kinds of possibilities. ZARDAX supports a wide variety of printers. From the best (and most expensive) letter-quality printers to the cheapest dot-matrix printers. ZARDAX's best feature is its price — only $295 at your Apple dealer.

If your dealer doesn't carry ZARDAX, have him mail us on The Source CL1791. Or have him call one of our distributors. In the USA, Action-Research Northwest, 11442 Marine View Drive SW, Seattle, WA 98146. In the UK, Rocon Ltd, Radley Road Industrial Estate, Abingdon Oxon. Or dealers may write us at Computer Solutions, P.O. Box 397, Mount Gravatt. 4122. Australia.

Computer Solutions



A Line on Friendly Utilities

BY THERON FULLER

Apple users are blessed with a vast selection of utility software. But the boon becomes a curse when you're ready to buy one and have to choose among them. This article describes some utilities that do their jobs well in a friendly way, whether you're a novice Apple user or a veteran programmer.

To be good, a utility program must meet certain criteria:

1. It must save you time and effort. The utility should help you use the Apple to do things the Apple does better than you and leave you free to do those things you do better than the Apple.

2. The functions of the utility must be easy to understand and easy to operate. You shouldn't have to be a computer expert to use a utility.

3. It must be a good buy. That is, the purchase price should be reasonable considering the value of the functions the program provides.

Here are some utilities that meet these criteria:

DOS Toolkit, Apple Computer Inc. A remarkable utility package, *DOS Toolkit* consists of two smaller toolkits: the *Apple II Assembler/Editor System* and the *Applesoft Toolkit.*

The *Apple II Assembler/Editor System* is a powerful 6502 relocating assembler with which you can edit and assemble very large machine language programs. Even if you're not an assembly language jockey, you can use the editor to write Basic programs in an easy-to-read structured format or to edit Basic programs with convenient text-manipulation features.

The *Applesoft Toolkit* is a programmer's delight. It contains the *Applesoft Programmer's Assistant,* the *High-Resolution Character Generator, Animatrix*—a character-set editor—and a *Relocating Loader* for machine language programs.

The *Applesoft Programmer's Assistant* helps you write and change Applesoft programs. It numbers lines automatically as you enter programs. It allows you to renumber and merge programs easily. It tells you the length of the program in memory, and it compresses a program in final form by deleting remarks. It also displays any control characters in a file, allows you to type special characters from the keyboard, and cross-references the variables in a program by line number. What's really nice is that all these features are available with just a couple of keystrokes.

Animatrix and the *High-Resolution Character Generator* simplify graphics; the former lets the user define graphic character sets, and the latter makes it easy to display them on the hi-res screen. The features of these utility programs make it easy to enhance graphic displays by adding text to them, and to add interest to screen printouts by using different character sets. These subroutines also can be used to develop animated graphics. The *High-Resolution Character Generator* can produce the same character set as the hardware character generator in your Apple—and many others besides: foreign alphabets and decorative alphabets, both with upper and lower case, and graphics alphabets for animation. Apple furnishes fifteen different character sets on the *DOS Toolkit* disk, and you can use the *Animatrix* program to generate a character set of your own design.

The *Relocating Loader* lets the user write machine language programs that can run on Apples with various amounts of memory by making machine-language subroutines relocatable.

The *Applesoft Toolkit* documentation includes three programs on the disk, *Skylab, Maxwell,* and *Ribbit,* that demonstrate the use of the *High Resolution Character Generator* and the *Relocating Loader.* In addition to providing a good example of how to use the *High-Resolution Character Generator* for animation, *Ribbit* is fun to play.

The *DOS Toolkit* is an excellent utility package that deserves a place in every serious programmer's library.

Super Disk Copy II and **Multi-Disk Catalog III,** Sensible Software. As your software library gets larger, so does the task of organizing and keeping track of all the files you are accumulating. Then there's the problem of converting files from the DOS 3.2 thirteen-sector format to the DOS 3.3 sixteen-sector format. A person with a single drive must have the skill of a one-armed paperhanger to use Apple's *Muffin* program to transfer all his/her files from DOS 3.2 disks to DOS 3.3 disks, since *Muffin* will transfer only one file at a time. *Super Disk Copy III* and *Multi-Disk Catalog II* greatly reduce the drudgery associated with organizing and maintaining your disk library.

Super Disk Copy III lives up to its name. It is an easy-to-use, menu driven program that allows transfer of all types of files under DOS 3.3, 3.2, and even 3.1, if your library goes back that far. You can copy individual files (Applesoft, Integer, binary, and text), copy an entire disk, or copy DOS. The program can easily be configured for any combination of DOS 3.2

and 3.3 source and target disks, and can be used with either one disk drive or two.

Super Disk Copy has a wealth of useful features. You can init copy disks, delete files, undelete files (a lifesaver), and lock or unlock files. You can replace control characters in file names, fix file sizes, alphabetize disk directories, and rearrange file sectors for improved access times. The program will display a map of sectors in use and tell you the number of free sectors on a disk. You can use *Super Disk Copy* to free up DOS and unused directory sectors for up to 13K of additional disk storage.

Multi-Disk Catalog III is a very fast machine language data base program for keeping track of the contents of your disk library. It very quickly reads and stores the file names, types and sizes, number of free sectors remaining, and actual volume number from each of your disks. It permits the identification of each disk with both a name and a disk identification number. Both sides of a disk can be given the same identification number for easy reference. *Multi-Disk Catalog* permits you to group together games, utilities, and other types of related files through the use of two-character field. A flip DOS feature allows the program to read information from DOS 3.3 and 3.2 disks, which greatly simplifies the task of cataloguing your disks. Fast, powerful sort and search features allow you to query the data base for specific information about your library.

Super Disk Copy III and *Multi-Disk Catalog III* take most of the pain out of organizing and maintaining the files in your disk library. With these utilities, library maintenance is a task instead of a chore.

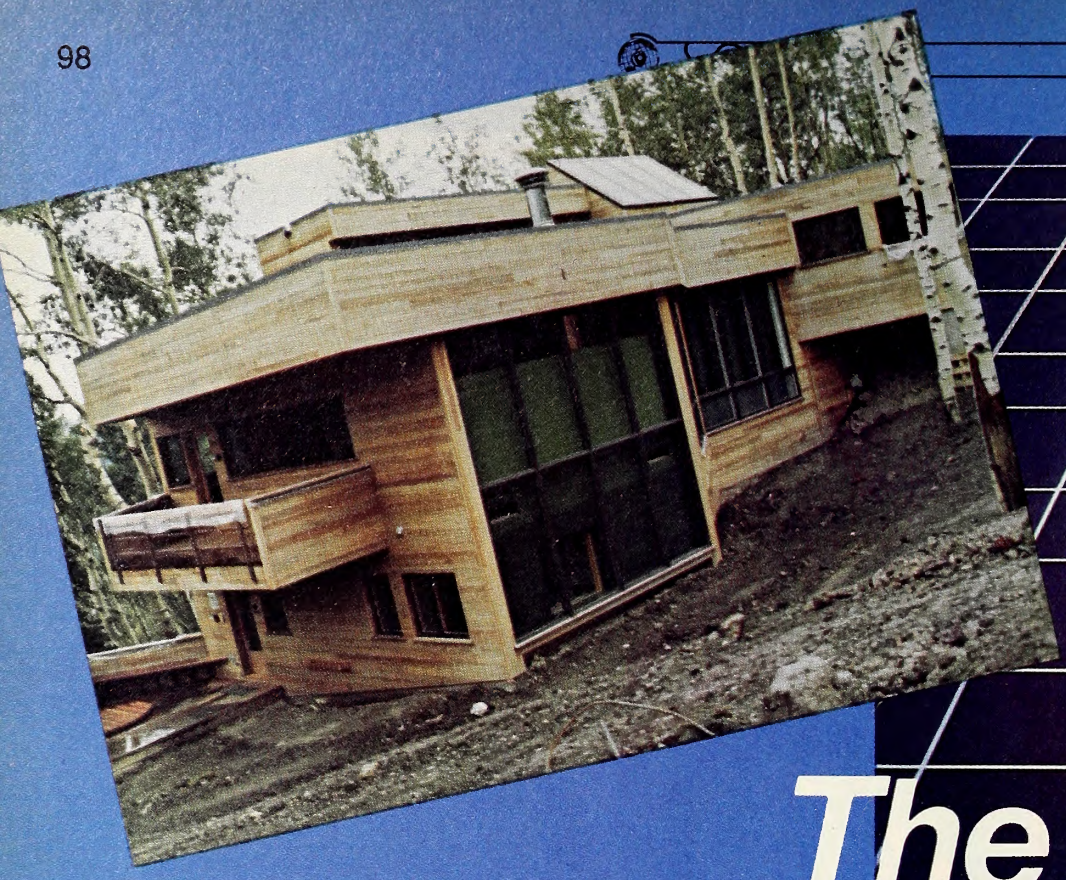
Double DOS by Leighton Paul and **DOS Plus** by Sensible Software. *Double DOS* and *DOS Plus* provide the software solution to the schizophrenic problem of having to live with both DOS 3.3 and 3.2 disks. Each of these utilities installs machine language code above Himem, allowing you to switch easily between DOS 3.3 and DOS 3.2. *Double DOS* and *DOS Plus* remain ready for action after most Basic commands, including FP, INT, RUN, LOAD, NEW, CLEAR, and after a reset. The process of switching from one DOS to another does not affect a program in memory and can even be done under program control if desired. This feature provides a lot of flexibility for such things as transferring individual programs between thirteen-sector disks and sixteen-sector disks. In certain circumstances, programs on copy-protected thirteen-sector disks can write text files to sixteen-sector disks.

Double DOS and *DOS Plus* make it possible for the Apple user to stay unhassled in spite of the two DOSs. They save the user a lot of rebooting—and a lot of swearing.

DOS Boss by the Beagle Bros. Beagle Bros. software has a certain flair and a lot of style. (One Beagle Bros. menu grumps at you to "hurry up" if you don't make a choice.) *DOS Boss* allows you to impart some of this flair and style to your own disks. This utility lets you tailor the DOS on your disks to suit your own tastes. You can shorten DOS commands such as catalog, load, and save to "C/," "L/," and "S/." *DOS Boss* allows you to change DOS error messages from "disk full" to "burp!" or "syntax error" to "I don't understand." You can customize your catalogs by changing the dull "disk volume" header to "sexy software" or "Frank's disk," or whatever else suits your fancy. Sector numbers and file codes can be altered or eliminated. One neat routine allows you to organize listings by file types.

All this foolishness—and practicality—is made easy by the DOS documentation. They also throw in their Apple Command Chart and the *Apple II Tip Book* for good measure.

Who says utilities have to be serious to be usable? ◨



The Solar

Designing Solar Homes with the Computer

BY CRAIG STINSON

This house is an example of passive solar architecture, which means that it provides a substantial part of its occupants' heating needs by the nonmechanical collection and distribution of radiant energy from the sun. Home of Dr. Patrick Moore and Patricia Moore in Snowmass Village, Colorado, the house was designed by the firm of David Finholm & Associates. Assisting in the design were an Apple computer and a set of programs by Solarsoft, Inc.

A passive solar house differs from an active solar house in that it relies as much as possible on so-called natural means of heat distribution—radiation, conductance, and convection—while avoiding such power-consumptive devices as pumps and fans. Buildings that are primarily passive in design but make moderate use of mechanical devices for heat distribution are sometimes called hybrid solar buildings.

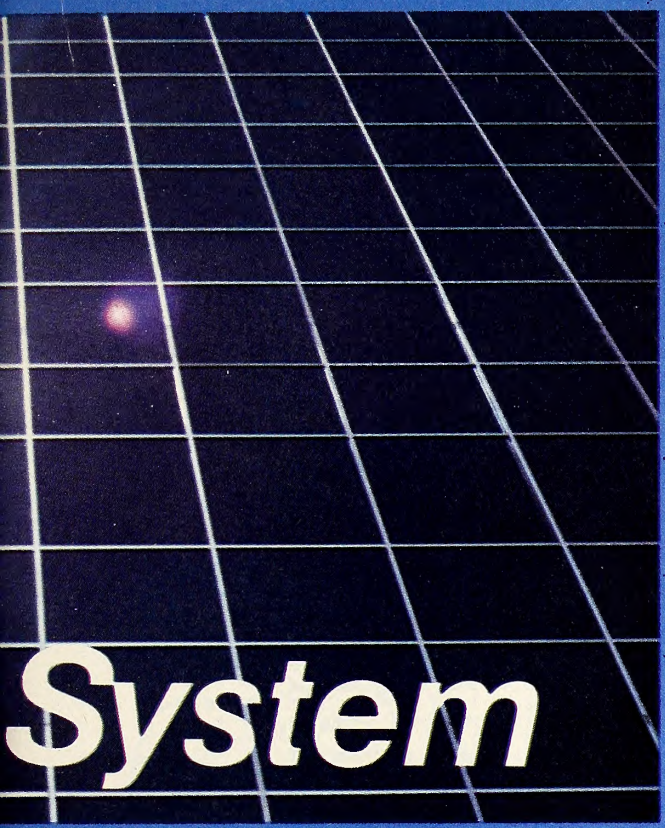
Like acoustics, solar architecture is an inexact science, mostly because there are so many variables that affect the way a building performs. Quite apart from climate, the architect must also consider other factors such as the amount of solar radiation that will be absorbed directly through glazing and indirectly via ground reflectance, the thermal conductance and capacitance of building materials and insulation, the amount of heat that will be generated within the building by its occupants and their activities—to mention only a few.

Nothing New Under the Sun. Fulfilling the thermal requirements of a building through passive solar means is not a new goal; passive solar concepts have been incorporated into building design in many places by many peoples, particularly in areas like the southwestern United States, where there's abundant clear sunshine.

But a passive solar *technology*—a coherent, quantified set of principles to guide architects—is a comparatively new development. Much of the impetus for this technology has arisen over the past half decade or so from research done at Los Alamos Scientific Laboratories in New Mexico. The results of the work at Los Alamos have been published in the two-volume *Passive Solar Design Handbook* (available from the National Technical Information Service, Springfield, VA, as DOE/CS-0127/1 and DOE/CS/0127/2). The first volume is primarily descriptive and introductory, the second quantitative.

The methodology established by the researchers at Los Alamos and the data base published in volume two of the handbook have been incorporated into three programs for the Apple, written by Matt Crosby and William Ashton and published by Solarsoft. The programs, named *Sunpas, Sunop,* and *T-Swing,* make up the computerized equivalent of a solar consulting service for architects, providing them with the best intelligence currently available on the subject of passive solar design. Solarsoft promises to update the programs from time to time, keeping them cognizant of ongoing research.

The three programs perform the following functions: *Sunpas* calculates the fraction of a building's heating requirements that can be expected to accrue from a given solar design in a given climate and the amount of auxiliary (nonsolar) heat expected to be required. *Sunop* takes data from *Sunpas* or from user input and performs calculations for economic optimization. *T-Swing* is a simulation program that calculates diurnal temperature fluctuations in specified thermal systems. *T-Swing* also contains a module called *Solgain* that



System by Apple

calculates such things as the angles of incident solar radiation at specified latitudes upon windows of specified orientations with overhangs of various dimensions, and so on. *Solgain* also calculates the amount of heat (in British Thermal Units) that can be expected to fall upon such specified windows on a clear day around the twenty-first of each month of the year.

All the Solarsoft programs provide a wealth of detailed information about how a given solar design can be expected to function, based upon correlations developed at Los Alamos. *Sunop, T-Swing,* and *Solgain* generate, in addition to numerical reports, a variety of graphic displays (see figures 1-5).

It Don't Mean a Thing If It's Got Too Much Swing. One of the important things about these programs is that they work together as an integrated unit. A good solar design is a sort of complex juggling act. The object is by no means simply to capture as much sunlight in a house as possible. In most climates a building that could supply sufficient solar energy to meet all its heating needs would be both economically unsound and problematical from the standpoint of heat distribution.

One of the major economic considerations is the phenomenon of diminishing returns. Additional solar glazing collects additional heat at an ever-decreasing rate, so that at some point the incremental cost of a solar BTU exceeds that of a BTU provided by nonsolar methods. The distribution problem arises from the fact that solar heat, once captured, needs to be stored and transported in some way to be useful. Adding solar glazing without a compensatory increase in storage mass would probably lead to unmanageable overheating in south rooms without necessarily providing enough heat to the north side of the house.

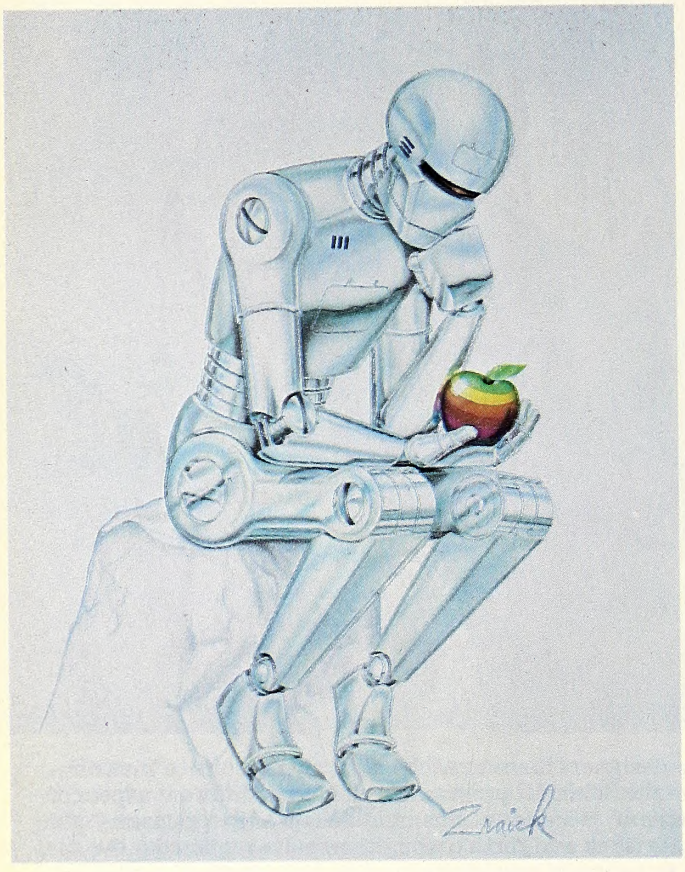
The designer, therefore, has a lot of tradeoffs to measure. Each of the Solarsoft programs considers a different aspect of the problem. Because of this, and because the programs can pass data back and forth among themselves, allowing the designer to perform sensitivity studies on individual variables, the Solarsoft programs can do much to facilitate the balancing process.

In the case of the Moore house, the Solarsoft programs were used for final design consulting. The firm of David Finholm & Associates had been working in the field of passive solar design for several years, so the residence was already well thought out from a solar point of view. The Solarsoft programs were used for three purposes: to confirm that the amount of thermal storage mass was appropriate for the building's 340 square feet of solar glazing; to determine whether any fans would be needed to prevent excessive temperature swings in the south side of the house and move heat into the lower north rooms; and to study the economic feasibility of a system of movable insulation.

The exploded drawing of the Moore house (page 101) shows the location of iits principal solar design features. The western, southern, southeastern, and eastern walls provide a total of 340 square feet of double-paned solar glazing. The greatest solar energy gains will occur through the south walls of the two western bedrooms and the library.

Where Sunlight Runs into a Wall. Directly behind the glazing of those two bedrooms is 290 square feet of Trombe wall. A Trombe wall is a wall of heavy masonry material that lies directly in the path of sunlight entering the building. The wall heats up during the day and gradually radiates its warmth to

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.

the rest of the room. Vents are provided at the top and bottom of the wall to create a convection loop, drawing cool air in from the room through the lower vent and returning warm air through the top. The vents can be closed at night to prevent the reversal of this circulation.

The southern face of the library is lined with 79 square feet of water column walls. These function in much the same way as the Trombe walls, except that the heat storage medium is water encased in metal. Water has certain advantages over masonry as a storage medium; it has a higher specific heat, which means it will hold more heat longer than masonry, and because of its fluid nature it will distribute its heat evenly within itself.

On the roof of the building are 68 square feet of clerestory solar windows and a solar water heater.

Figure 1 shows a portion of the analysis performed by *Sunpas*. The table headed Climate Responsive Takeoffs includes

CLIMATE RESPONSIVE TAKEOFFS

	SURFACE	TYPE	W	H	AREA	U	U*A	AZ	TI	AB	RMI	O1	O2
1	E WALL 1	NS	22	7.7	132	.038	5	-	-	-	-	-	-
2	WINDOWS	DG	10.6	3.5	37	.56	20	98	90	1	9	0	0
3	N WALL 1	NS	23.3	7.7	159	.038	6	-	-	-	-	-	-
4	WINDOWS	NS	6.6	3	20	.56	11	-	-	-	-	-	-
5	E WALL	NS	5.4	21	113	.038	4	-	-	-	-	-	-
6	N WALL	NS	21.5	21	452	.038	17	-	-	-	-	-	-
7	E WALL	NS	6.4	20.5	122	.038	4	-	-	-	-	-	-
8	WINDOW2	DG	2.5	3.6	9	.56	5	98	90	1	9	0	0
9	N WALL	NS	15	20.5	290	.038	11	-	-	-	-	-	-
10	WINDOW1	NS	5	3.6	18	.56	10	-	-	-	-	-	-
11	NW WALL	NS	20	20.5	357	.038	13	-	-	-	-	-	-
12	WINDOW1	NS	8.8	1.5	13	.56	7	-	-	-	-	-	-
13	WINDOW2	NS	12	3.3	40	.56	22	-	-	-	-	-	-
14	W WALL	NS	28	20.5	446	.038	16	-	-	-	-	-	-
15	DOOR1	NS	3	6.7	20	.15	3	-	-	-	-	-	-
16	WINDOW1	DG	10	2.8	28	.56	15	82	90	1	9	0	0
17	DOOR2	NS	3	6.7	20	.15	3	-	-	-	-	-	-
18	WINDOW2	DG	17.1	3.5	60	.56	33	0	90	1	9	72	24
19	S WALL	NS	16.5	3.9	64	.038	2	-	-	-	-	-	-
20	S WALL	NS	16.5	2.2	36	.038	1	-	-	-	-	-	-
21	TROMBE	TWI	16.5	14.3	185	.226	41	8	90	.9	9	14	0
22	WINDOW1	DGI	4.5	3.5	16	.39	6	8	90	1	9	0	0
23	WINDOW2	DGI	16	2.2	35	.39	13	8	90	1	9	14	0
24	SE WALL	NS	10	6.5	38	.081	3	-	-	-	-	-	-
25	WINDOW1	DG	7.2	3.7	27	.56	14	43	90	1	9	0	0
26	SE WALL	NS	10	2.2	22	.038	0	-	-	-	-	-	-
27	TROMBE	TWI	10	10.5	105	.226	23	43	90	.9	9	14	0
28	S WALL	NS	16	16	117	.038	4	-	-	-	-	-	-
29	WATER	WWI	7	11.3	79	.56	44	8	90	.9	9	14	0
30	WINDOW	DGI	5.3	11.3	60	.56	33	8	90	1	9	0	0
31	S WALL1	NS	11.5	9.5	109	.038	4	-	-	-	-	-	-
32	BUFFER WL	NS	28	8	224	.026	5	-	-	-	-	-	-
33	ROOF	NS	36	45.5	1638	.024	39	-	-	-	-	-	-
34	SKYWALLS	NS	54	5	202	.038	7	-	-	-	-	-	-
35	GLASS	DGI	18.9	3.6	68	.56	38	8	90	1	9	12	0

TOTAL CLIMATE RESPONSIVE HEAT LOSS = 496 BTU/HR-DEGREE

INFILTRATION AND PERIMETER LOSSES

INFILTRATION:	VOLUME	35000
	AIR CHANGES	.5
PERIMETER	EXPOSED FEET	222
	F-VALUE	.17
INFILTRATIVE LOSS	235	BTU/HR-DEGREE
PERIMETER LOSS	38	BTU/HR-DEGREE

FIGURE 1.

data input by the architect as well as some calculated by the program. In the column marked Type, the abbreviations are as follows: NS stands for a nonsolar surface—one that will not acquire solar energy; DG stands for direct gain and includes all windows except those facing north; TW and WW stand for Trombe wall and water wall, respectively; and the addition of the letter I to an abbreviation indicates the presence of movable insulation.

The U-value in the sixth and seventh columns of the table is a measure of thermal conductance of a surface, expressed in BTUs per hour per square foot per degree Fahrenheit of temperature difference between the inside and outside of the building. The rightmost six columns apply to the solar surfaces only and measure the azimuth (the compass orientation of a surface, with south as zero), the tilt of the surface in degrees above the horizontal, the absorptivity of a surface, the R-

value of movable insulation, and the dimensions and position of overhangs above windows.

The table in figure 2 is a sort of bottom line of the *Sunpas* analysis. The figures under SLR are the calculated monthly solar-load ratios—the ratios of solar energy gained to heat lost. SSF stands for solar savings fraction and represents the portion of the house's heating requirements that can be expected to accrue from the solar design features. The last column shows the amount of auxiliary heat that will have to be supplied to maintain the specified temperature range—in this case 65 to 75 degrees—within the house.

CALCULATED AUXILIARY HEATING REQUIREMENT

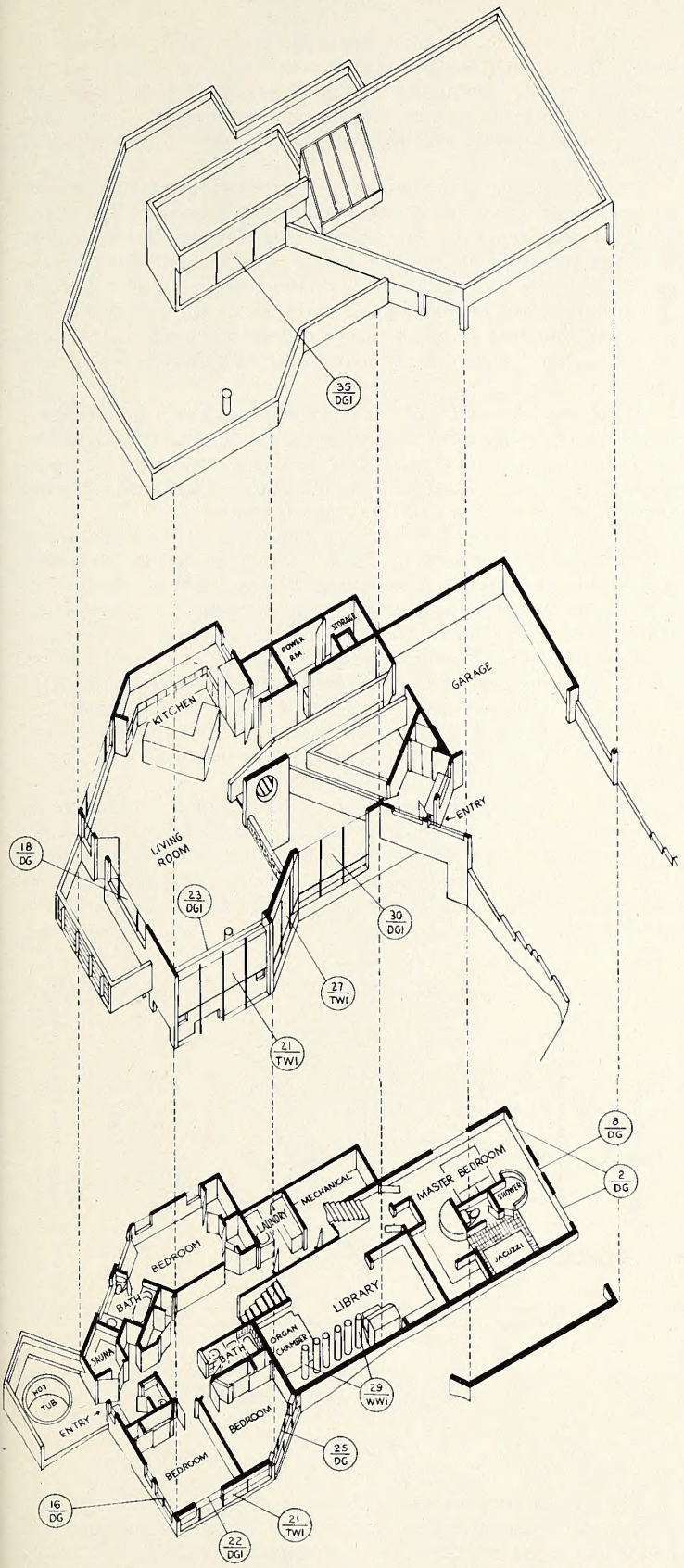
MONTH	SLR	SSF	AUX
JAN	1.1	.38	9.6
FEB	1.6	.55	5.5
MAR	2	.67	3.6
APR	2.2	.71	2
MAY	3.6	.88	.5
JUN	10.4	.98	0
JUL	62.6	.98	0
AUG	30.4	.98	0
SEP	7.6	.98	.1
OCT	3	.83	1
NOV	1.4	.5	5.3
DEC	1.2	.4	8.9

YEARLY AUXILIARY 36.5 MILLION BTU
YEARLY SSF: 57%

MOORE RESIDENCE

FIGURE 2.

Two *Sunpas* analyses were run on the Moore house, one with the presumption of movable insulation, the other without. The program calculated solar savings fractions of 57 percent in the model with movable insulation and 33 percent in the model without.



IN THIS EXPLODED AXONOMETRIC DRAWING OF THE MOORE RESIDENCE, YOU ARE LOOKING DOWN FROM THE SOUTHWEST, APPROXIMATELY FROM THE ANGLE OF THE MIDAFTERNOON SUN. THE NUMBERS IN THE CIRCLES REFER TO THE BUILDING'S SOLAR SURFACES, AS LISTED IN FIGURE 1. THE LETTERS BELOW THE NUMBERS DENOTE THE TYPES OF SOLAR SURFACES. DG = DIRECT GAIN; DGI = DIRECT GAIN WITH MOVABLE INSULATION; TWI = TROMBE WALL WITH MOVABLE INSULATION; WWI = WATER WALL WITH MOVABLE INSULATION.

```
ECONOMIC VARIABLES
------------------

PROJECT : MOORE
LOCATION: ASPEN

PERIOD OF ANALYSIS (PT)..... 8
DOWN PAYMENT FRACTION (DPR)..... .2
MORTGAGE TERM (T)..... 30
ANNUAL INTEREST RATE (I)..... .16
ANNUAL DISCOUNT RATE (R)..... .13
ANNUAL INFLATION RATE (O)..... .1
FED, STATE, LOCAL TAX RATE(FS)... .3
TAX CREDIT RATE (CT)..... .3
MAXIMUM TAX CREDIT (MA)..... 3000
PROPERTY TAX RATE % OF TC (TX)... 1E-03
ANN.INFLAT.PROPERTY VALUE (O1)... .15
PASS.SYSTEM YR.EXP.% OF TC (OMI). 5E-03
ANN.INFLATION OPERATING EXP.(O2). .1
COST OF BACKUP FUEL ADJ.COP (PO). 6.41
FUEL ESCALATION RATE (E)..... .2
ASSESSMENT VALUATION FACTOR (V).. .6
BACKUP SYSTEM CAPITAL COST (BSC). 2000
BACKUP SYSTEM YEARLY COST (OB)... 1E-03
FIXED COST (FC)..... 0
SOLAR COLLECTOR AREA (A)..... 0
VARIABLE COST (VC)..... 15
CONSERVATION COST (CC)..... 3500
BUILDING LOSS COEFFICIENT (L).... 11603
SOLAR SAVINGS FRACTION (SSF)..... 0
ANNUAL HEATING DEGREE DAYS (DD).. 8426
AUX. FUEL PRICE (FP)..... 6.41
```

FIGURE 3.

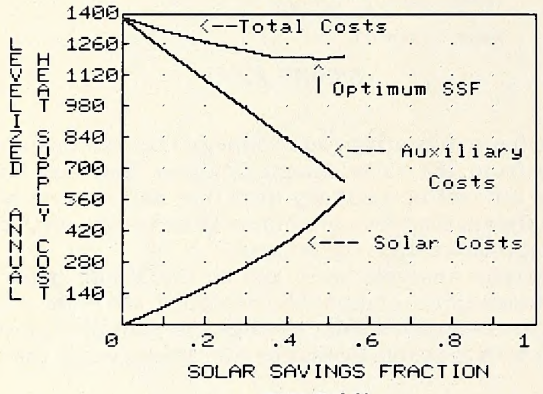


FIGURE 4.

These results were then fed to *Sunop* to determine, among other things, whether the movable insulation's costs could be justified by the savings in auxiliary energy it would provide. Figure 3 lists underlying economic assumptions that were made for the *Sunop* run. Figure 4 shows part of the results of that run.

The ascending and descending curves in figure 4 represent the solar and auxiliary costs, respectively, as functions of the solar savings fraction. The third curve represents total costs. Because the slope of the solar cost line increases with increasing SSF, while the auxiliary costs decrease at a steady rate, the total cost line reaches a minimum at an SSF of just over 50 percent. Had this graph been extended to the right, the total cost line would show a marked increase at SSF values above 60 percent.

Wide Variety of Light Shed by Sunop. *Sunop* generates a great deal of other information as well, including a table of life-cycle costs, an annual cash flow analysis that takes into consideration local tax credits for solar designs, and a chart of net present value versus solar savings fraction.

Finally, the Moore house was analyzed by *T-Swing* to see what kinds of temperature fluctuations could be expected, given the architect's provisional design, and to determine whether a fan would be needed to move the heat from the south side to the lower north rooms. For comparison the building was simulated using anticipated temperature conditions in March and January, and with three different models. The first model included neither a fan nor the water wall columns located in the library. The second model included the water wall but no fan. The third included both water wall and a variable-speed, thermostatically controlled fan.

Figure 5 is a graphic overlap of the first and third models. The lines marked 1A and 1B represent the twenty-four-hour temperature curves for a typical March day for the southern rooms, with and without water wall and fan, respectively. The 14A and 14B lines provide the same information for the lower north rooms.

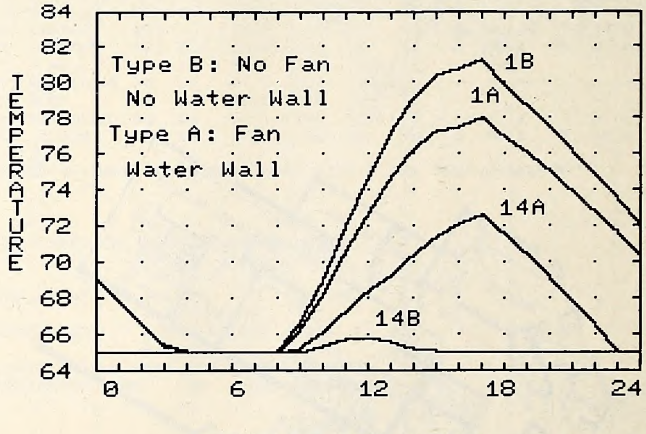


FIGURE 5.

The graph shows clearly that the presence of the fan and water wall moderates the temperature variations between one end of the house and the other and helps to avoid overheating in the southern rooms. The lines marked 1 and 14, incidentally, represent the extreme ends of a complex fourteen-node thermal system that takes into account all the important components of the house's thermal storage system.

The Solarsoft analyses on the Moore house resulted in recommendations for the installation of a temperature-activated system of movable insulation along the thermal walls of the house and the inclusion of a variable speed fan to carry warm air into the lower north side. ∎

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 MUDGE RANCH ROAD
COARSEGOLD, CA 93614
209-683-6858

VISA, MASTERCHARGE, CHECK OR COD

THE BASIC Solution

By Wm. V. R. Smith

The response to the subroutine for turning DOS on and off has been strong. Several readers' letters explained how the routines had solved problems they had been working with. Such letters are very helpful in channeling the direction of future Basic Solutions. If you have found a programming problem, please send it to *Softalk*. Not all your questions can be answered, but those that might be of interest to other readers will be explored.

This month's Basic Solution routine searches through a string variable S$ for the occurrence of another string held in the array SS$(). SS$() may have as many strings as you wish to search for. The total number of strings to search by is held in the variable SN. A sample program is provided to demonstrate how to set up the required variables.

The subroutine returns to the calling program with the variable FL set to the location in SS$( ) that was found. FL will return as 0 if no occurrence was found.

This routine is quite useful to search through a mailing list for certain zip codes, last names, cities, and so on.

Your ideas and subroutines are always welcome. 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.

```
1000  REM  ********************
1001  REM  *
1002  REM  * SEARCH STRING
1003  REM  * VAR S$ FOR
1004  REM  * SS$(1-SN)
1005  REM  * FL>0 - FOUND
1006  REM  * FL=0 NOT FOUND
1007  REM  *
1008  REM  * WM V R SMITH
1009  REM  ********************
1010  L = LEN (S$)
1020  FOR X = 1 TO L
1030  FOR Y = 1 TO SN
1040  SL = LEN (SS$(Y))
1050  A$ = MID$ (S$,X,SL)
1060  IF A$ = SS$(Y) THEN FL = Y:Y = SN + 100: NEXT Y:X = L + 100: NEXT X: RETURN
1070  NEXT Y
1080  NEXT X
1090  FL = 0: RETURN


  10  PRINT : PRINT "----------": PRINT
  20  SS$(1) = "APPLE"
  21  SS$(2) = "COMPUTER"
  22  SS$(3) = "MONITOR"
  23  SS$(4) = "DISK"
  24  SS$(5) = "PRINTER"
  25  SN = 5
  30  PRINT "ENTER A PHRASE WITH ONE OF THESE WORDS"
  31  PRINT
  35  FOR X = 1 TO SN
  40  PRINT X;" - ";SS$(X)
  45  NEXT X
  50  PRINT : INPUT "THANK YOU ";S$
  55  GOSUB 1000
  60  IF FL = 0 THEN 80
  65  PRINT : PRINT "YOU ENTERED WORD # ";FL
  70  GOTO 10
  80  PRINT : PRINT "YOU DID NOT ENTER A PROPER WORD"
  90  GOTO 10
  99  END
```

ON-LINE systems

PRESENTS

HI-RES ADVENTURE

MISSION: ASTEROID
HI-RES ADVENTURE # 0

MISSION: ASTEROID is an introduction to the HI-RES ADVENTURE family of games. This adventure is slightly easier and a little shorter than our other HI-RES ADVENTURE games. MISSION: ASTEROID is designed to acquaint beginning Adventure players to the wonderful world of Hi-Res Adventure.

In this adventure you find that an Asteroid is about to hit the Eorth and destroy it. It is your mission, as an Astronaut, to rocket ta the Asteroid and blow it up before it reaches Earth.

OVER A HUNDRED HI-RES PICTURES. (Looks greot on b/w and color televisions).

FULL 21-COLOR!! HI-RES GRAPHICS. (Each roam a work of art).

YOUR GAME MAY BE SAVED FOR LATER CONTINUANCE.

RUNS ON BOTH 48K APPLE-II AND APPLE-II PLUS.

THIS EXCITING GRAPHIC GAME WILL CHALLANGE YOUR IMAGINATION AND TEST YOUR CREATIVITY EVERY STEP OF THE WAY!

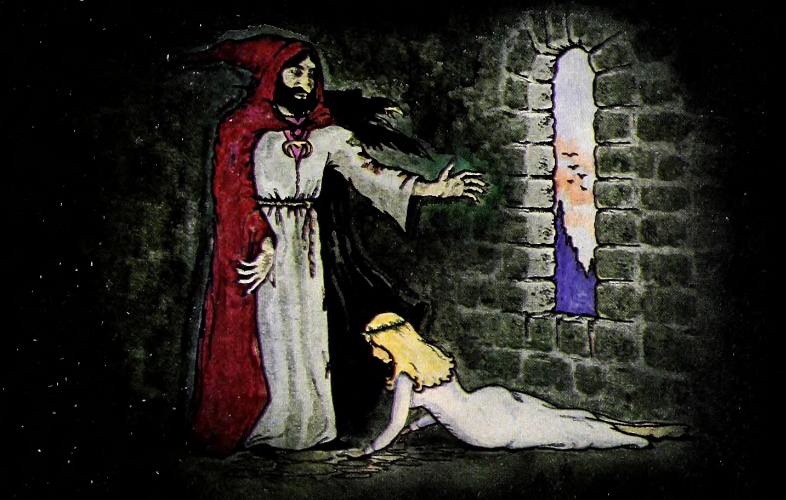
$19.95 Disk

MYSTERY HOUSE
HI-RES ADVENTURE # 1

Through the use of over o hundred Hi-Res pictures you ploy ond see your odventure. You communicote with HI-RES ADVENTURE in ploin English (it understands over 300 words.) All rooms of this spooky old house oppear in full Hi-Res Grophics complete with objects you can get, corry, throw, drop or ?

In this particular HI-RES ADVENTURE gome, you are tronsported to the front yard of a large, old victorian house. When you enter the house, you ore pulled into the mystery, murder and intrigue and can not leave until you solve the puzzles. Your friends ore being murdered one by one. You must find out why, ond who the killer is. Be careful, because the killer may find you! As you explore the house there are puzzles to be solved ond hazards to overcome. The secret passoge woy may leád you to the answer.

$24.95 Disk



THE WIZARD AND THE PRINCESS
HI-RES ADVENTURE # 2

Only ON-LINE SYSTEMS could deliver a HI-RES ADVENTURE game on such an epic scale. In this adventure yau find you must do battle against on evil wizard in order to save the life of the princess. To find the wizard and his castle you must first cross deserts, oceans, mountains, travel to an island and encounter many strange beasts. You will be forced to learn magic, navigate at seo and dig for treosure. This game should provide months of adventure.

HUNDREDS OF HI-RES PICTURES. (Looks great on b/w and color televisions).

FULL 21-COLOR!! HI-RES GRAPHICS. (Each room a work of art).

YOUR GAME MAY BE SAVED FOR LATER CONTINUANCE.

RUNS ON BOTH 48K APPLE II AND APPLE II PLUS.

BY FAR THE MOST AMBITIOUS GRAPHIC GAME EVER WRITTEN FOR THE APPLE!!

$32.95 Disk

All of these Hi-Res Adventures are available naw at your local computer store. They will run on any 48K Apple II or II Plus with a disk drive. To order direct send the purchase price plus one dollar for shipping to:

ON-LINE SYSTEMS - 36575 Mudge Ranch Rd. - Coarsegold, CA 93614 - 209-683-6858

Visa, Master Chorge, C.O.D. or Checks Accepted.

DB MASTER™ WOULD LIKE TO THANK THESE BIG NAMES FOR OUR GOOD NAME!

IBM WESTINGHOUSE BROADCASTING **AT&T**
DOW CHEMICAL **NASA** PROCTOR & GAMBLE
EF HUTTON ALLIED CHEMICAL **HONEYWELL**
APPLE COMPUTER **TRW** CHASE MANHATTAN
MERRILL LYNCH E I DU PONT **NEW YORK LIFE**
ITT-CANNON **PRICE WATERHOUSE** U S ARMY
UNIV OF CALIFORNIA G E **UNIV OF MANITOBA**
XEROX **LAWRENCE LIVERMORE LAB** SONY
OWENS CORNING SHEARSON LOEB RHOADES
CABLE NEWS NETWORK **HUGHES RESEARCH**
ARTHUR YOUNG & CO MOUNTAIN COMPUTER
JOHNS HOPKINS **MINOLTA** SHERATON CORP
McDONNELL DOUGLAS BANK OF LOUISVILLE

Impressed?

Thanks to customers like these, DB MASTER is the second fastest selling business software package for the Apple II.

Even more impressive are the comments from some of our customers. As Mr. M. Robert McElwain, Senior VP of the Bank of Louisville states, "After having purchased more than 400 software packages, I'm still trying to establish a list of top ten that I wholeheartedly endorse. I don't have ten, but I'M ONE CLOSER—DB MASTER has been added to the list. It's a real treat to acquire those which are truly outstanding. DB MASTER is quality software which I highly recommend."

Equally impressive is the range of features built into DB MASTER. As Mr. McElwain continues, "I could comment on the screen formatting, short forms, security, auto date . . . but where do I stop? With over 100 Apples, we think we recognize good software when we use it."

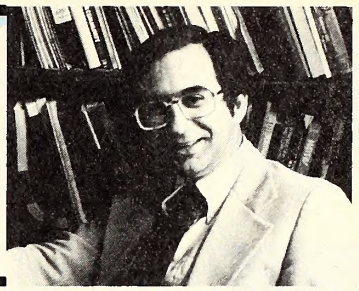
Our special thanks to Mr. McElwain. And to all our equally impressed customers.

As they all know, in today's highly competitive marketplace, a good name is hard to come by.

STONEWARE™
MICROCOMPUTER PRODUCTS
50 Belvedere Street, San Rafael, CA 94901 (415) 454-6500

Mind Your Business

BY PETER OLIVIERI



For several issues, we've been focusing on data base management systems. With this column we'll conclude, temporarily, our examination of these packages by discussing *DB Master* and the PFS series. Altogether, we've evaluated five packages: *Data Factory, CCA Data Management System, Datadex, DB Master*, and *PFS*. Next time we concentrate on this type of utility, we'll cover *Super Kram, Request, Thinker,* and the *Data Reporter,* among others.

Remember that opinions in any review reflect the biases and preferences of the reviewer. For example, certain features of a package may be more important to you than to me; thus, you should weight them more heavily than the review does in reaching your purchasing decision. Evaluations are intended only as a guide.

The PFS Series. The PFS software series is designed to be a personal information management system. The system was developed by the Software Publishing Corporation (Palo Alto, CA).

The first package, *PFS,* requires a 48K, sixteen-sector, disk-based Apple II system. With this program, you are able to create files, enter data in those files, copy files, search through files, print files, and remove records from files.

The most striking feature about *PFS* is how easy it is to use. The manual is extremely well done, helpfully illustrated, and completely understandable. Even if the program itself weren't so friendly, documentation like this would go a long way toward making it so; since it is, the guide is gravy.

The *PFS* system is operated on the principle that information is kept in forms in files. With *PFS,* you are the one who designs and manages these forms. Each form may have up to thirty-two pages, each page containing about twenty lines of data. You can fit about one thousand pages on one data disk. While you can use as many disks as you wish for a file, searching through multiple disks for information can become quite cumbersome.

The PFS menu offers you options to create a file, add a form (enter data), copy a file, search through a file, print a file, and remove a record.

To create a file, you must give it a name and then design the form that will be used to enter your data into that file. A file name may have only eight characters in it, which you may find a bit short.

Designing a form is remarkably easy. You first move a cursor to wherever on the screen you would like to place your field. Then you simply type the field name followed by a colon and move on to the next place you want to put a field. The length of each field is determined by the distance to the next field heading. There is virtually no limit.

When you're ready to enter data, and select that option, a blank form of your design appears. Right arrow moves the cursor among field positions; you merely type the data. When you press control-C, all the data currently on your form is saved on your disk and another blank form appears.

You can, if you wish, add attachments to any record at a later time. However, these attachments apply only to the individual record, not one in the file. You cannot define fields on an attachment; they are really for additional comments. You can search on these comments when you're looking for a particular record.

Copying a file using *PFS* requires that you have a system with two disk drives. While this is an obvious restriction, you could copy a disk using the Apple Pascal copy features.

Searching through the file for certain records is quite easily accomplished. When this option is selected, a blank form is displayed on the screen. You then enter a variety of search criteria in any or all of the fields of your form. Records that match all the criteria you've specified will then be displayed. You can search using a full-item match, a partial-item match, or relational operators with numeric data (< > =).

One important feature, surprisingly glossed over in the manual, is the ability to update a record. The oversight may have occurred because the process is so simple. When you have retrieved a particular record through searching, you can retype a field value or even enter it for the first time. After making updates this way, pressing control-C saves the updated record on disk.

The print options with *PFS* are limited. A companion product, *PFS: Report,* is a more powerful report generator designed to be used with *PFS;* the cost of both *PFS* packages combined is consistent with the cost of other DBMS packages.

Within *PFS* alone, the print option allows you to select which records to be printed just as if you were defining search criteria. Then you may choose to have field names printed or not, and you may specify number of lines per record. You're then asked to identify which fields of the record you wish to have printed. However, you may not change the order of the fields, and your only formal choices are to print a field on the same line as a previous field or on a new line. You may also print individual records from the search mode, but here the entire record is printed without options.

Some of the more obvious applications for *PFS* include phone messages, recipes, patient records, customer lists, coin collections, and so on.

The PFS: Report. It's best to have two disk drives to take full advantage of the features of this software. *PFS Report* is a table consisting of up to nine vertical columns. Each row of the report contains information from one record. The rows can be sorted numerically or alphabetically, and the program can perform calculations on numeric information. *PFS* can total, average, or count the number of items in a column. Subtotals are also possible. You may include up to three *derived* columns in your report—derived from data currently in the file. However, if you have only one disk drive, you cannot sort; you may only have seven columns in your report; and you cannot produce subtotals.

The *PFS: Report* menu allows you to print a report, predefine a report, or set new headings on a report. The records to be selected for printing are chosen by using the same techniques as in a *PFS* search. You can then number each field and specify the order of the columns in the final report. In addition, you can indicate which columns are to be totaled, averaged, or counted.

PFS: Report allows three of your final report columns to be derived from other data in your file. Columns can be added together, multiplied by a constant and so on. When using derived columns, *PFS: Report* slows down quite a bit.

Other features of this package allow you to predefine (and save) up to eight report formats and set new report headings if you wish. The *PFS* series is nicely done, easy to use, and a valuable addition to one's software library. Further compatible modules are promised.

DB Master. *DB Master* is a comprehensive data base management system designed to provide the user with a sophisticated tool for processing data.

The system requires 48K and DOS version 3.3. Of course, a printer significantly increases the usefulness of the system,

DBMS

	BACKUP:	EQUIPMENT:	USER GUIDE:	EASE OF USE:	CREATING A FILE:	ENTERING DATA:	REPORT FEATURE:	PRINTING CAPABILITY:	SORTING:	SEARCHING:	STRENGTHS:	WEAKNESSES:	COMMENTS:
The Data Factory Micro Lab 3218 Skokie Valley Road Highland Park, IL 60035	Provided with purchase	APPLE II or, Apple II Plus 48K, DOS 3.3	Good to very good	Very good	Very good	Excellent	Good	Good	Good to very good	Good to very good	(1) You can list the files on your disk. (2) Amount of disk space left is displayed. (3) You can get file statistics. (4) You can select another file whenever you wish. (5) Transfer option allows you to relocate records into a new data base (a very nice feature). (6) Construct and Append permits the adding of more fields at a later date.	Field names are limited to five characters. There can be lots of disk swapping (particularly with a one-disk system). The guide could be improved.	Very good product, friendly. In addition, Micro Lab has other products compatible with it.
DB Master Stoneware Microcomputer Products 50 Belvedere Street San Rafael, CA 94901	Send for free backup	Apple II Plus, 48K, DOS 3.3	Fair to good	Very good	Good	Excellent	Excellent	Excellent	Very good	Excellent	(1) Comprehensiveness of the system. (2) Password protection features. (3) The Dynamic Prompting. (4) Extensive report creation options. (5) Capability of up to one hundred fields.	The user guide could be much improved. It takes time to learn to use the features of this package. Field names reduce potential level of field.	Excellent data base management system. It would be particularly appropriate for the small business owner.
PFS and PFS: Report Software Publishing Corporation P.O. BOX 50575 Palo Alto, CA 94303	Send $15 with registration	APPLE II or, Apple II Plus 48K, DOS 3.3	Excellent	Excellent	Excellent	Excellent	Good to very good	Good to very good	Fair	Very good	(1) User's guide is exceptionally well done; includes actual screen images. (2) Very easy to learn how to use. (3) Well thought-out series.	Multiple disk files can be cumbersome. Once designed, you cannot add fields to a form. File names are limited to eight characters.	This is an excellent package, particularly suited for the home, though not limited to use there. It is easy to use and clearly designed for the new user. It may be best suited to applications with reasonably short forms, although forms that include lengthy text fields can find no comparable system.
Datadex Information Unlimited Software 281 Arlington Avenue Berkeley, CA 94707	Shipped when warranty card received	APPLE II Plus, 48K, DOS 3.3	Very good to excellent	Excellent	Excellent	Excellent	Good to very good	Good to very good	Very good	Very good to excellent	(1) Easy to design forms—actual screen images are shown in the user's guide. (2) You can add your own programs to the package's utility file list. (3) Search option has a very nice browse feature. (4) You can create subfiles from your master file. (5) New fields can be inserted.	You are limited to how many fields you can fit on a screen. There are no additional pages or form attachments.	It is impressively easy to learn to use Datadex. Especially likable are some of the special features (browsing, guessing, translating files, list of peripherals). Best suited to files without a great many fields.
CCA Data Management System Personal Software 1330 Bordeaux Drive Sunnyvale, CA 94086	You can make a copy yourself	APPLE II, 32K or 48K, DOS 3.2 or 3.3	Very good to excellent	Good	Very good	Very good	Good	Good	Very good	Good	(1) Error messages referred to on the screen are well documented in the manual. (2) You can easily calculate how much free space is available. (3) Interfaces with VisiCalc and contains instructions on how to use the data with other programs.	The screen format for creating files could be better. You can have only twenty-four fields in each record.	This is a good system with good documentation.

and dual disk drives are a definite advantage. The version being reviewed is version 3, the latest *DB Master* update.

The user's guide supplied with *DB Master* is quite comprehensive. The material itself is well-written, but it requires careful reading. A manual designed with more illustrations and a more pleasing typeface (important to many users) might have been preferable, but, given the completeness of the material, these shortcomings can be overlooked.

DB Master is an excellent DBMS despite some of the criticism you're about to read.

Files created with *DB Master* may have up to three levels of password protection—a useful feature in many business settings. When you use this package, a feature called dynamic prompting lists (in abbreviated form) at the bottom of the screen the information you need to know to proceed.

DB Master fields can be numeric, alphanumeric (one to thirty characters), or computed. You may have up to ten computed fields in each record. The report generator allows up to twenty-four fields in each report. All fields you design may contain a default value that will automatically be entered as data if you wish.

Creating a file is not as simple as it is with some of the other DBMSs. A *DB Master* screen form layout sheet is provided to facilitate the creation of your form before you begin using the program. You put your field names and lengths on a sheet in the exact position that you wish them to appear on the screen. When you build your form on the screen, you must respond to several questions printed at the bottom of the screen: the name of a field; the field type; where on the screen a field should be placed (this is a horizontal and vertical location; you use your layout sheets to indicate these grid locations; the length of the field; and whether there is a default value for this field. One nice feature here is that you may enter a field editing mode that allows you to reenter any of the fields or reposition them. (You may not add or delete fields during the editing mode, however.) The form that you design may have up to nine pages.

DB Master provides you the opportunity to specify both primary and secondary key fields in a record. This can significantly increase the speed with which you can retrieve a record for editing or display.

As you use *DB Master,* it becomes necessary to swap disks every now and then. There is a master program disk, utility disks that store housekeeping information about a file, and master disk for the actual data. You may even need to have sort disks. This swapping can become cumbersome, particularly if you don't have two disk drives.

When you wish to enter data to a file, you select the Opening Existing File option and insert the appropriate utility diskette. You enter records into a file by filling in a blank form on the screen. Pressing the return key moves you to the next field, and escape moves you to the previous one. The data entry procedure is quite easy to follow and a pleasure to use.

DB Master can find and display records quickly if you search by the primary key or even by the secondary key. All other searches are done sequentially. Searches (for display or for printing) may include a search for a range of values in a field, a wild card search (for a beginning series of characters or for characters included within a string), a relational search (> < > = <>), and an and/or search. A statistics feature enables an additional search that displays the count, sum, average, and standard deviation for a particular field.

"And pleose keep on eye on Consolidated Investments until we con fix the computer."

McCrae/The Bulletin/Sydney/reprinted from World Press Review/May 1981

Pressing control-F finds the selected records and displays them. Various other control key options allow you to page through a form, continue with the search, specify other search criteria, or print the current page of the form that is displayed on the screen.

An editing mode permits changing the contents of any field in your file, deleting a record, or using a calculator mode to adjust the contents of a particular field.

The report generator in *DB Master* includes a description of both short forms and long forms. A short form is a form that contains only a few of the fields in a file's main form. This can be quite helpful when you're entering data to a file or when you're updating or editing fields in your file.

Creating a report format is a rather complicated process, but the end result is a report that's truly user customized. The *DB Master* report generator can sort records on as many as six fields at a time, date reports, print comments at the top of each page, automatically number pages, use up to nine lines for column titles, use up to twenty-four computed fields, accommodate up to 132 columns of information, utilize a variety of line spacing options, place footnotes at the bottom of a report page, total and subtotal columns, and use a wide variety of record selection options. This report generator is one of the most powerful we've seen to date.

A file maintenance module contains procedures for reblocking a file (making more efficient use of storage space), obtaining file statistics, and changing passwords, among other things.

Finally, convenient appendices in the documentation list error messages and some of the more common problems that users may encounter, as well as a table that shows how many records can be stored of various record lengths and with various numbers of fields.

Although the user's guide takes a good deal of time to read and become familiar with, the programs supporting the system are quite easy to use. The effort expended in becoming familiar with this package is well worth it. This is a very comprehensive and complete data base management system.

Comparing Packages. It is not easy to compare packages in a particular area, especially when the packages are quality products showing a great deal of professionalism in their design but differing in their intents as well. Each has its own strengths and weaknesses, and each has a certain application area for which it is best suited. Ultimately, you must decide what characteristics are most important to you. Is price the deciding factor? The number of records that can be stored? The ease with which the package can be used?

The chart summarizes the key characteristics of each of the packages we have discussed, listing strengths and weaknesses of each and evaluating some of the more important features. These evaluations reflect my own reactions and should be so interpreted. You may wish to cut this out and file it under DBMS for reference. We will be adding to it in future issues.

In Conclusion. All these data base management systems were nicely done. You would not regret having bought any one of them. However, it's important to get the one that's just right for your needs. This requires you to do a little homework on what your needs are.

We recommend that you have two disk drives and a printer as components of your system if you expect to do a lot of work with a data base. Also, look for packages that allow you to interface your data with other packages or with programs you have written. Your needs will no doubt grow, and you want your system to be able to grow along with it.

Please send your reactions or comments about these or other packages you are familiar with. Reactions from users can be shared with others to benefit us all. ■

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

Taylor Pohlman

When last we left the Lone Ranger, huge boulders were crashing down the slope toward his tiny campfire. . . . No, I'm sorry to say that the September column wasn't quite that breathtaking or cliff-hanging. However, taking the Apple III out for a spin does excite a lot of people, and we hope that includes you. If you haven't read last month's column, we recommend you get a copy. This series is progressive in that each article builds on the previous one. We're going to assume that you have been following the series, so that each month we can cover a new topic in the least possible amount of purple prose.

Another reminder before we start: we welcome questions and comments on this series or on Basic in general. Because of deadlines, each article is being written before the previous month's is in print. Thus, reaction to your timely comment will be somewhat delayed. Let those cards and letters roll in, and the responses will show up just as soon as inhumanly possible.

The SOS File System Revisited. After a brief discussion of the Apple III SOS file system, last month's column concluded with something of a challenge for you. We were working with a program to dump the contents of the screen to the Silentype printer, and we mentioned that the program could be generalized for any file, including text files. The point was that SOS takes care of all the details about how each device works, so the user can change things at will. For reference, here's the program with which we were working:

```
50    OPEN#1,".silentype"
90    INVOKE"readcrt.inv"
150   FOR vp=1 TO 23
155     VPOS=vp
160     FOR hp=1 TO 80
165       HPOS=hp
170       PERFORM readc(@value%)
180       PRINT#1;CHR$(value%);
190       NEXT hp
200     PRINT#1
210     NEXT vp
900   VPOS=23:HPOS=1
1000  END
```

Before we modify this program to generalize it, did you try to simplify the program by using VPOS and HPOS directly in lines 150 and 160? By that we mean:

```
150 FOR VPOS=1 TO 23
160 FOR HPOS=1 TO 80
```

If you did, you know that Basic will respond to this change with the classically familiar syntax error, because VPOS and HPOS are reserved words and cannot be used as index variables.

To continue, the challenge was to generalize the screen dumping program so the output could go to any file. Here's one solution to that problem:

```
50    VPOS=23:HPOS=1
60    INPUT"Name af file ta dump screen to: ";filename$
100   OPEN#1,filename$
110   INVOKE"readcrt.inv"
120   FOR vertical=1 TO 23
130     VPOS=vertical
140     FOR horizantal=1 TO 80
150       HPOS=harizantal
160       PERFORM readc(@value%)
170       PRINT#1;CHR$(value%);
180       NEXT horizontal
190     PRINT#1
200     NEXT vertical
210   CLOSE
300   VPOS=23:HPOS=1
310   END
```

Several differences are worthy of note. First, the cursor has been repositioned in line 50 to the bottom of the screen to avoid overwriting any existing data. The user is then prompted in line 60 to type in the name of the output file. Note that this can be any filename legal on the Apple III that accepts output (printers, the communications port, a disk text file, even .CONSOLE itself).

Note also the addition of the *close* statement at line 210. This ensures that all files are properly written to and dispensed with at the conclusion of the program. Failure to close files properly can leave some data still in memory (since files aren't automatically closed at the end of the program). This can have some interesting consequences if the file in question is a disk file and you switch to another diskette that doesn't have that file created on it. Now is the time to form the habit of closing all files at the end of a program.

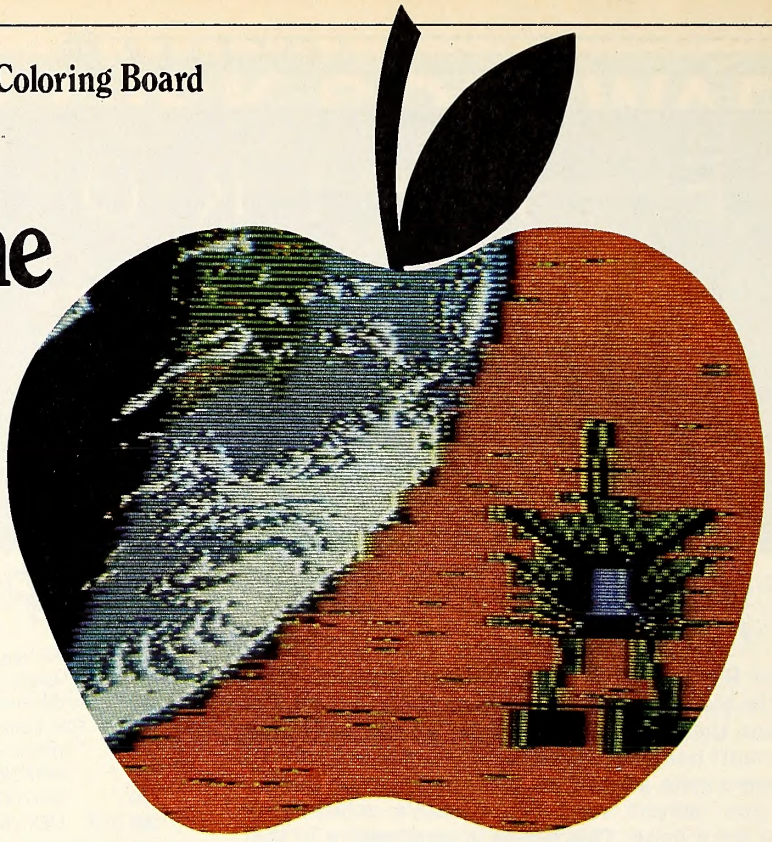
Running this program can be instructive. Obviously, if you reply ".SILENTYPE" to the prompt, it will work like the first example. Try replying ".CONSOLE" now. After the usual initial whirring of the disk to load the Invokable Module, the program appears to go to sleep for forty seconds or so. What's happening is that the program is reading a character and then copying it back on top of itself! The Apple III is working its little heart out, and the result is as exciting as watching bread mold.

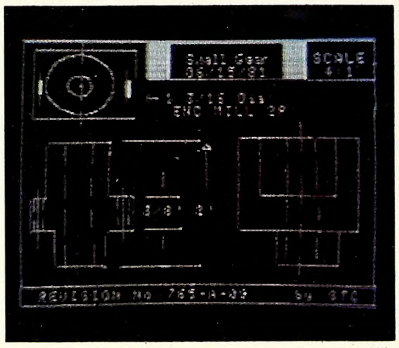
Now try replying with a disk file name (you can just make up a name, as long as it follows the filename rules). The disk will whir as before. This time Basic has a number of jobs to do. First, it must open the disk file using the name you gave it (let's assume you typed MYFILE.SCREEN). Basic tells SOS to create the file (assuming it doesn't already exist), by making an entry in the directory of the current disk volume and finding initial space for the file. Basic then sets up a buffer area in memory for communication of data to and from the file. Since Apple III divides the disk up into blocks of 512 characters, this internal buffer is 512 bytes. This buffer size is fixed no matter what record size you specify. Later on in this article, we'll look at techniques that use that piece of information to ensure maximum efficiency and performance in disk-based application programs.

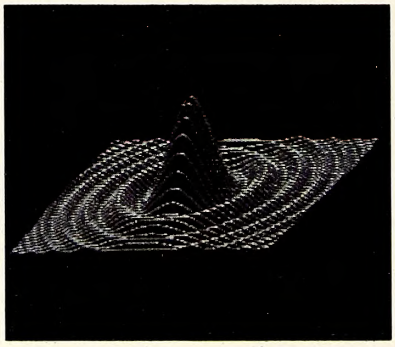
Once the file is opened, Basic then invokes the

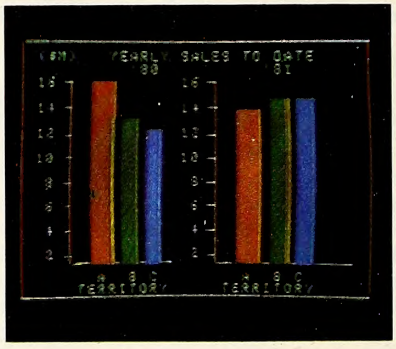
Introducing STC's Low-Cost Coloring Board

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.

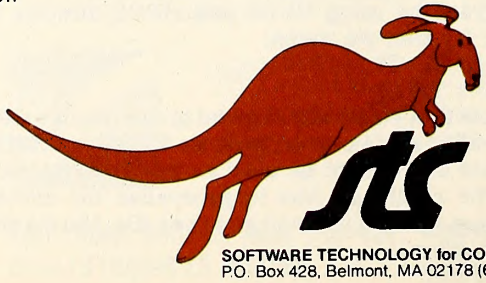
Contact Your Local STC Dealer.

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)



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

Readcrt module, and execution begins. Notice that, although the printer in our previous example started almost immediately, there is a noticeable pause before the disk spins into action, and it appears to spin only four times before the program stops. What's happening is this: line 170 prints one character at a time into the buffer. After eighty characters, line 190 prints a carriage return into the buffer and then starts the next line. After a little more than six lines of the screen (480 bytes plus six returns plus 26 bytes of line 7 to be exact) the 512 byte buffer is full and must be written to disk. That's the first spin of the disk that writes the first block of the file. Next, the block number is incremented, and more writing starts from line 7 of the screen. 512 bytes later the same process is repeated until all the screen is read by the program and written into the last buffer. Some arithmetic would convince you that Basic is in the middle of its fourth buffer when the program finishes reading line 23 of the screen. That's when the previous comment about being sure to close files comes in handy. The Close command in line 210 forces the current buffer to be written to disk, even if it's not full, and the directory entry is updated to reflect the new file information.

After running the program, the catalog listing of the file should look something like figure 1.

TYPE	BLKS	NAME	MODIFIED TIME
TEXT	00005	MYFILE.SCREEN	00/00/00 00:00

CREATED TIME	EOF
00/00/00 00:00	1863

Figure 1.

Notice that Basic identified the file as a text file automatically, because the Print# command was used to write to it. Notice also that the Blks used column shows five. That disagrees with what we had predicted, since the screen data should have been able to fit into four blocks (2048 bytes). The reason for the extra block is that SOS allocates an extra block as an index block to store information about where the rest of the blocks in the file are physically located. This ensures that a large file can be created, even if the disk is fragmented into small areas of unused space. If you look closely at the directories of various files, you will note that all of them have one more block than the EOF column would indicate, except for the one-block files, which have no need of an index block. In this case, the EOF (end of file) is after 1863 bytes. That works out to twenty-three lines of eighty characters (1840 bytes) plus twenty-three carriage returns for a total of 1863 bytes. "Close enough for folk music," as they used to say in high school.

One last subject before we move on to further explore files. The Silentype gave us a permanent record of what was on the screen, but since we wrote the results to a disk file this time, we need a way to dump the contents of MYFILE.SCREEN to the printer. The following program easily accomplishes the task and serves as a general file-to-file transfer program:

```
5   INPUT"Nome of file to dump: ";inputfile$
10  OPEN#1,inputfile$
15  console=0
20  INPUT"File to dump to: ";outputfile$
25  OPEN#2,outputfile$
30  check$=MID$(outputfile$,1,3)
35  IF check$=".co" OR check$=".CO" OR check$=".Co" THEN console=1
40  IF console THEN HOME
45  ON EOF#1 GOTO 65
50  INPUT#1;o$
55  PRINT#2;o$;
60  GOTO 50
65  IF console THEN HPOS=1:VPOS=23
```

```
70  CLOSE
75  END
```

There. As long as you don't try to read from the printer and print to the keyboard, it should work fine.

Note that we've checked in line 35 to see if the device being written to is .CONSOLE. If so, line 40 clears the screen to reproduce exactly what was there when the original program was run. Line 65 repositions the cursor to the bottom of the screen so that the prompt will not cause the top line to scroll out of view.

More on Files. The subtle and nefarious purpose of this lesson, if you haven't realized by now, is to provide more insight into Business Basic disk files. We've remained true to the promise of the first article and assumed that you are skilled in Basic, so hang on as things get more interesting. . . .

So far, we've considered only the type of disk files referred to as *text* files. These are files that contain ASCII characters, which are representative of what would be printed out if we wrote data to the screen instead of disk. For now we'll stick with this file type and later touch on *data* files, a useful and relatively unique file type on the Apple III.

We've already learned that the disk is organized into 512-byte blocks. In fact, Basic text file records can be of any reasonable size. Instead of using the Open statement, which assigns a default of 512 bytes, we could have used the Create statement, which allows up to 32,767-byte records to be used. Of course, the record size of a particular file is of no consequence if we are merely going to read each string in order (as we did with the contents of the screen).

The real power of creating files of various record sizes is to be able to read data on a particular item in the film randomly without having to deal with the other data in the file. For example, if we had wanted to print the twenty-first line of the screen in the previous example, it would be necessary to input the first twenty lines, discard the data, and then finally read and print the line we wanted. A much more efficient way would be to create the file as a random access file with record size of eighty-one bytes. Since each record will correspond with one line of the screen, we have an easy way to address the data in question. Compare the examples below with the previous sequential access examples:

```
50    VPOS=23:HPOS=1
60    INPUT"Name of file to dump screen to: ";filename$
70    CREATE filename$, TEXT,81
100   OPEN#1,filename$
110   INVOKE"readcrt.inv"
115   cum$=""
120   FOR vertical=1 TO 23
130     VPOS=vertical
140       FOR horizantal=1 TO 80
150       HPOS=horizontal
160       PERFORM readc(@value%)
170       cum$=cum$+CHR$(value%)
180       NEXT horizantal
190     PRINT#1,vertical;cum$
195     cum$=""
200     NEXT vertical
210   CLOSE
300   VPOS=23:HPOS=1
310   END
```

Note that we have added line 70 to create the filename with the proper record size. The notation of Text is extra baggage, since the Print statements in the program will automatically define it as a text file, but it is good practice to be specific. I have also added a new wrinkle in lines 115, 170, 190, and 195. Instead of printing each character as it is read, the variable "cum$" is used to accumulate characters as they are read

from the screen. Line 190 prints the entire line of the screen using the vertical position as the record number. The result when running this program seems the same as when running the sequential version, except for one thing. If you catalog the resulting filename, it should look something like figure 2 (assuming a name of SCR.DUMP.RND).

TYPE	BLKS	NAME	MODIFIED TIME
TEXT	00005	SCR.DUMP.RND	00/00/00 00:00

CREATED TIME	EOF
00/00/00 00:00	1944

Figure 2.

Everything is the same except the length. It turns out that, when a file is created, the first record is record 0, not record 1. This is consistent with the first element of an array being element 0. Therefore Basic has reserved twenty-four (not twenty-three) records of eighty-one bytes each for a total of 1944 bytes.

Now that we have associated a record number with every line on the original screen, we can locate a given line by just giving its number instead of having to read through all the other lines to find it. Witness the modified read program:

```
5     INPUT"Name of file to dump: ";inputfile$
10    OPEN#1,inputfile$
15    console=0
20    INPUT"File ta dump ta: ";autputfile$
25    OPEN#2,outputfile$
30    check$=MID$(autputfile$,1,3)
35    IF check$=".ca" OR check$=".CO" OR check$=".Co" THEN console=1
40    IF console THEN HOME
45    ON EOF#1 GOTO 65
47    INPUT"record number to dump: ";rec
48    IF rec=0 THEN 65
50    INPUT#1,rec;a$
55    PRINT#2;a$;
60    GOTO 47
65    IF console THEN HPOS=1:VPOS=23
70    CLOSE
75    END
```

This program is very similar to the previous program except that line 47 asks for the specific record to dump, line 48 gives us a way out by checking for zero, and line 50 has been modified to read directly to the record number previously entered.

Some experimentation with this program will produce interesting results. Try reading records 1, 6, 12, and 18. In each case, you will cause a disk access (the whirring is a clue) to read the particular record. Now try reading records 6, 7, 8, 9, and 10 in any order you choose. The first record you read will probably cause a disk access, but the others should occur virtually instantaneously without causing disk activity. This is because SOS is still buffering flies in 512-byte blocks, and all those records fall within one block. There was no need to reread the disk because the data was already in memory. Careful planning of your record sizes and reading sequences can have the effect of substantially increasing the performance of your program, if as many reads as possible occur within the current buffer.

One interesting postscript before we proceed: If you ask for record 6 there will typically be a disk read, as we've said. If you immediately request record 5, another disk read will be performed. This is what you might expect, but more is going on here than meets the eye. Simple calculation will prove that record 6 actually occupies space in both block 1 and block 2 of the file. The first six records, 0 through 5, occupy 6*81 or 486 bytes of the first block, leaving only twenty-six bytes in that first block for record number 6. The remaining fifty-five bytes are in block 2.

Thus a read to record 6 actually triggers two disk reads, one to load in block 1 for the first part of record 6, and one for block 2 to obtain the remainder of the record. Therefore, when you requested record 5, Basic had to go back and reread block 1 (remember, only one block is kept in memory per file).

A little more arithmetic will show which other records are in this same situation. The moral is simple: if possible, make your record sizes such that they evenly divide into 512 or are a multiple of 512. That may waste a little space, but the waste may be more than compensated for in the ability to predict when disk access will take place.

A Final Challenge. We just reviewed the last five or six paragraphs and discovered that our usual humorous style has been replaced by long, detailed discourses of unrelieved tedium. There is, unfortunately, no letup in sight.

To this point we have been using "record number" files (called *random access* by most people) with record numbers that span a rather narrow range. SOS permits random files to have record numbers in the range of 0 to 32767. However, SOS does not demand that a file actually have all the records present on the disk. Records are allocated as written, with only a little space taken up to keep track of where everything is. To illustrate the power this gives, consider the following problem:

A distribution company wants to keep track of their part numbers and descriptions. The part numbers are four-digit numbers. Following is a simple program to create the part number file.

Between now and next time, you could try writing a program to retrieve part number information randomly and make changes as required. Without further ado . . .

```
5     HOME
10    PRINT"Parts file Creote and Add pragrom"
20    PRINT
30    PRINT"Type 1 to Creote o parts file":PRINT
40    PRINT"Type 2 to add to an existing parts file"
50    PRINT:INPUT"Yaur selectian: ";a$
60    IF a$="" THEN 1000
70    a=VAL(a$)
80    ON a GOTO 100,400
90    GOTO 5
```

```
100   PRINT:INPUT"name of new parts file:";a$
110   IF a$="" THEN 5
120   CREATE a$, TEXT,64
130   PRINT"Parts file ";a$;" created."
140   GOTO 5
400   PRINT:INPUT"Name of existing parts file: ";a$
410   IF a$="" THEN 5
420   OPEN#1,a$
430   HOME
500   PRINT:INPUT"Part number to add: ";a$
510   IF a$="" THEN 5
520   a=VAL(a$)
530   IF a<1 OR a>32767 OR INT(a)<>a THEN 500
535   rec=a
540   rec$=a$+"\"
545   PRINT:INPUT"Description: ";a$
550   IF LEN(a$)>30 THEN a$=MID$(a$,1,30)
560   rec$=rec$+a$+"\"
570   PRINT:INPUT"location: ";a$
580   IF LEN(a$)>10 THEN a$=MID$(a$,1,10)
590   rec$=rec$+a$+"\"
600   PRINT:INPUT"Quantity on hand: ";a$
610   a=0: a=VAL(a$): IF INT(a)<>a THEN 600
620   rec$=rec$+a$+"\"
630   PRINT:PRINT"Record is: ";rec$;" OK? ";
640   INPUT"";a$
650   a$=MID$(a$,1,1):IF a$<>"y" AND a$<>"Y" THEN 430
660   PRINT#1,rec;rec$
670   PRINT:PRINT"Record added."
680   GOTO 430
1000  PRINT:PRINT"End of parts file program."
1010  CLOSE
1020  END
```

This does not presume to be a model program in terms of its error checking, efficency, or even logic design (note all the Gotos, patently offensive to the initiated). We tried to keep the program simple and straightforward, allowing plenty of room for improvements. One or two things are worth pointing out to help you with your inquiry program. Since each field could be of varying length within certain limits, the backslash character is used to delimit each item. You'll want to strip these out when you retrieve the record. Look up the function Instr; it'll make it easy.

Once you've typed this program in, trying it out can be interesting. Try several values for part number, including some larger ones (greater than a thousand, at least). Unless you add records that are sequential, each one will probably trigger a disk access as the appropriate block is written to disk. After adding several, get out of the program by typing Return to the part number and selection prompts and check out the catalog entry on the file. Assuming you used the name MY.PARTS as a file name when you used the create option, the entry will look something like figure 3.

TYPE	BLKS	NAME	MODIFIED TIME
TEXT	00007	MY.PARTS	00/00/00 00:00

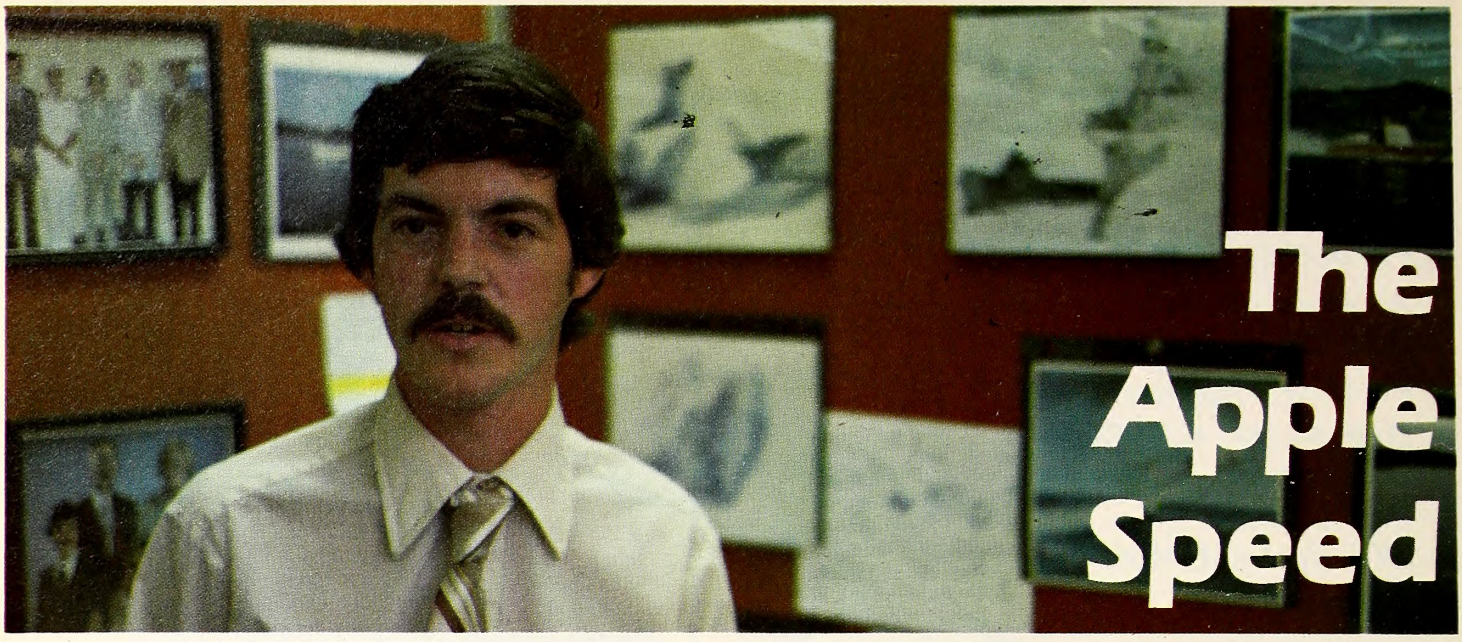
CREATED TIME	EOF
00/00/00 00:00	85376

Figure 3.

Look at that EOF value! It seems that you have a huge file until you notice that the Blocks Used column is still pretty small. What SOS has done is report the EOF at the end of the highest record number you used, while allocating only those blocks that it actually needed. Some micros (and some mainframes, for that matter) would require that all the blocks be allocated before any could be written.

Well, have fun until next time. Then we'll try to lighten it up a little as we talk about the mysterious data file type and start using the massive amount of memory in the Apple III for some really fast indexing schemes. Before this series is over, you should be able to write some pretty hot database programs. Till then, ponder the following: Is it true that disk-based programs are written by BLOCKheads? ∎


GOBLINS
Escape to Goblin Country,
An adventure to
challenge your skills,
Fantastic High Resolution
graphics and animation
help you on
your journey.
48K Applesoft Rom
and disk.
$27.50
Checks, Visa, MC, C.O.D.
E YOUR LOCAL DEALER OR WRITE
HIGHLANDS COMPUTER SERVICES
14422 S.E. 132nd
Renton, Washington 98056
(206) 228-6691



The Apple Speed

BY JIM SALMONS

The David Taylor Naval Ship Research and Development Center in Bethesda, Maryland, is one of those hush-hush military places where everything is clean and neat, and all the buildings are discreetly plain, so as not to reveal what is inside to nosy spy satellites. Within these halls, you can find aerospace engineer David Rousseau, who uses an Apple to do flight simulations of an exotic jet fighter aircraft.

Rousseau is a member of the Special Vehicle Office, housed in the Aviation and Surface Effects Building. Rousseau's work area is fairly typical, with the exception of the most incredible joystick you've ever seen sitting on a table next to an Apple.

The pictures on the walls aren't typical, either, and help explain Rousseau's professional excitement about his work. One is a drawing of the jet tank he designed to go fifty miles an hour on water before swooping up on the beach in a sea invasion. Another is a robotic reconnaissance plane that could fly for days, sniffing out enemy submarines.

"And here's my pride and joy—the Power Augmented RAM Landing Craft," Rousseau beams, pointing to a third picture. "I actually was the pilot for a one-fifth-scale prototype that we ran up to about sixty miles per hour."

The landing craft is similar to a streamlined barge with two large jet engines sticking out the front on the end of a forward angled mast. The blast of the jets propel the craft while being directed under the hull to lift the boat on a cushion of air. This would allow the lander to carry more than a hundred soldiers and equipment at speeds of seventy to eighty miles per hour.

Rousseau's obvious enthusiasm for his research is obvious as he explains his introduction to the Apple microcomputer: "VATOL is an acronym for Vertical Attitude Take Off and Land aircraft. The concept has been around for twenty years and has been proven in prototype flights of the Convair XYF-1 and Ryan X-13."

The idea, according to Rousseau, is simple in principle. Sit a plane on its tail and blast it off like a Buck Rogers rocket rather than using the traditional taxi and wing-lift take-off. Landing is similarly dramatic. Swoop the plane into a nose-up stall and blast the jets just enough to balance the plane like a pencil on the end of your finger. A hook on the nose engages a landing bar, and the plane hangs like a suit of clothes in the closet.

But there's no bureaucratically administered research without politics. The VATOL concept had been moved to the back burner since the initial flight tests were performed with the limitations of twenty-year-old engine technology. Assumptions about the lack of feasibility were based on outdated information, so research funding for the idea was very limited.

"You might think all we do is pursue research all day, wildly spending taxpayers' money," Rousseau says. "It isn't true. Our funding is competitive. I have to be my own salesman. I come up with a design, develop a proposal, and start pounding the doors at the Pentagon, looking for someone to support the project. A member of the New Vehicle staff was a VATOL advocate. Before leaving the group, he was able to attract a very small research grant to pursue modern VATOL technology."

It's clear that engine and avionics technology has developed to a degree that the aircraft could be built to carry a reasonable weapon payload. But what might it be like to fly a modern VATOL fighter?

"The higher-ups at the Pentagon were convinced that a pilot would be incapable of adjusting to the transitions of verti-

THE LANGU

Luck has a special meaning to Sam Cottrell, aphoristically communicated by a sign in his office:

"Luck": Where Opportunity Meets Preparation

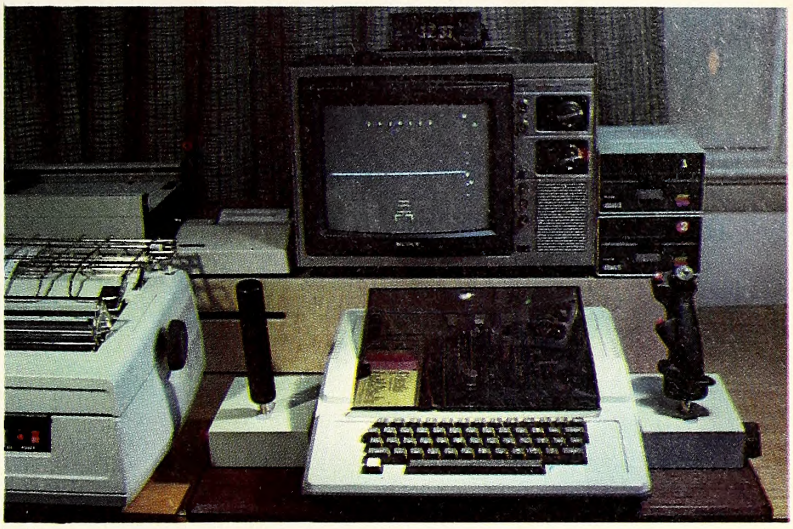
Performance demands of the Navy VATOL flight simulator contract provided the opportunity to create MicroSpeed. Sam Cottrell brought the preparation.

After leaving his childhood home of Greenwich, Connecticut, Cottrell earned an undergraduate degree in civil engineering at Cornell University. In 1954, he was called to active duty as a second lieutenant in the Air Force. First a civil engineer, then a student pilot, Cottrell became a flight instructor and logged a few thousand hours in the air.

With Air Force encouragement to continue developing his professional skills, he attended the University of Chicago and received the MBA in research and development management. The Air Force rewarded him with a management position at their Albuquerque research and development center. Cottrell administered projects in the development of directed-energy weapons—Buck Rogers things like particle beam and laser cannons. But three years flying a desk was too much. He was grabbing his desk-set pen instead of a cockpit joystick.

Cottrell volunteered to join the Starfighters when action began to accelerate in Vietnam. After an exhilarating crash course in air combat, he was off to Southeast Asia.

Souped-Up Supersonic Simulator



David Rousseau (left) takes great pride in his high-speed Apple outfitted with a joystick the like of which no arcade can match (above).

cal to horizontal flight. They felt the maneuvers would be physiologically disorienting," Rousseau explains, "so in 1979, with a tiny budget, we were challenged to prove them wrong."

Rousseau and his research colleagues needed to test pilot reactions to a plane that hadn't been built. The obvious solution was a flight simulator, but such simulators were routinely huge investments and amazingly complex devices using mainframe computers.

Limitations to problem solutions are often the stimulus to innovation. So it was the squeeze of the funding that brought the first Apple to the David Taylor Naval Ship R&D Center. It came under the arm of Sam Cottrell, professional problem solver.

"Sam had flown one hundred Vietnam combat missions in an F-104 Starfighter while I was still in high school," Rousseau says. "He retired from the military and has his own consulting firm near here. He's also a whiz at computers and engineering. So we were pretty confident that he could handle the simulator contract. I can't tell you exactly what he did, except that he developed his own super-high-speed language system for the Apple."

"It's a rather ingenious package that contains a hardware unit using an AMD 9511 math processor chip and a software unit that includes his own variation on the minicomputer Forth language," Rousseau explains. "Amazingly, Sam was able to get this Apple to calculate fast enough to give a realistic simulation of the head up display of a hypothetical aircraft that outperforms any jet fighter ever built!"

The head up display, or HUD, is like something out of *Star Wars*. At the speeds of today's fighters, the pilot cannot afford to look away from the view out the front of the cockpit. So you don't see dials and gauges in the planes; instead, there are up

GOTO 122

AGE OF SPEED

Two years and one hundred North Vietnam combat missions later, Cottrell came home to a creative research and development job in the combat applications group at Eglin Air Force Base. Working on problems surfacing in Vietnam combat, with the necessity for quick, practical solutions, refined Sam's problem-solving skills.

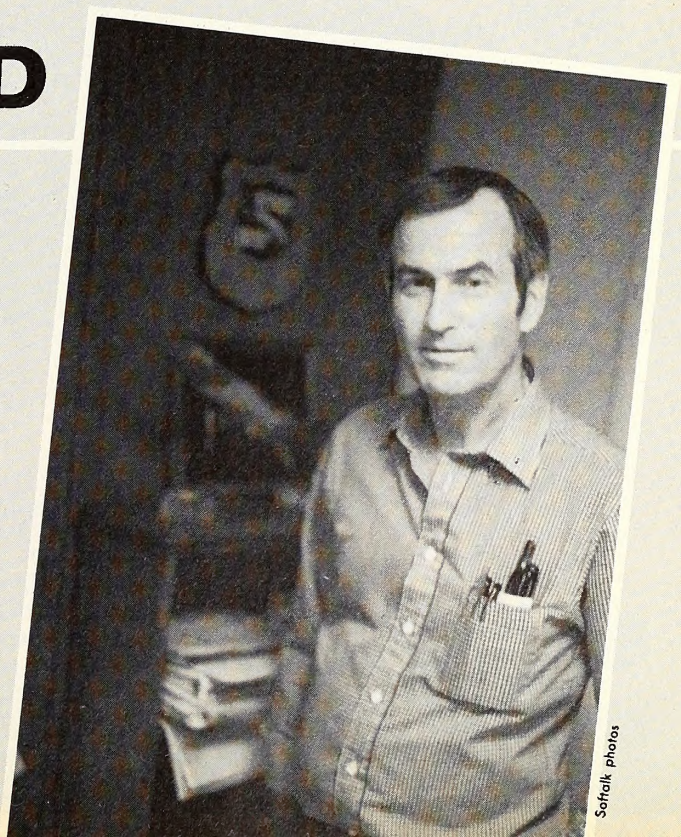
He also attended the Army Command and General Staff College, where he learned to study old wars using today's weapons. Cottrell started using computers in mathematical modeling.

After graduating, Cottrell hit the big leagues with a super-cerebral assignment at the Pentagon in the USAF studies and analysis group. Computers were still big and expensive, necessitating a whole floor of support personnel to produce air warfare simulations.

Cottrell developed an appreciation for the computer's potential as a vital tool in analytical problem solving, but he had little reason to expect his future would revolve around small and inexpensive computing. Steves Wozniak and Jobs were probably surfing or building hot-rods in their now-famous garage, and an Apple was still something you ate.

He next took his knowledge of computers to the management of an avionics and fighter maintenance organization. Then, after, precisely twenty years' service, he retired.

In 1974, Cottrell returned to the Washington, D.C., area and



Softalk photos

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.

*Applesoft is a trademark of Apple Computer, Inc.

hung out his shingle as an analytical consultant. When people asked him what kind of consultant he was, he asked what kind of problem they had. Cottrell got contracts by agreeing to perform complex analytical studies that would have cost a fortune to perform on mainframe computers. This often meant wearing his fingers to the bone on the keypad of a mid-seventies programmable calculator.

Cottrell founded Apple Analytics as the corporate identity for his growing consulting business in September 1976. He watched as the microcomputer left the wirewrapping hobbyist stage and emerged as a viable problem-solving tool. After a year of shopping, he bought a 32K Apple II. He calls Apple "the magic rubber computer" because of its flexibility, expandability, and Apple Inc.'s willingness "to open the black box"—to allow user adaptation through peripheral interfacing and software exploitation.

Using a tape recorder and the family TV to round out his first system, Cottrell took on analytic assignments unapproachable on his calculator tools. His clients were amazed at the analyses he could produce on very limited budgets. Of course they didn't see the endless hours Cottrell spent programming in Applesoft Basic to achieve the results.

Now back to opportunity. When the Navy was faced with the complex VATOL simulation and a limited budget, they naturally thought of Cottrell, who was establishing a reputation for doing big things on small computers.

Most would have laughed at a request to make a micro perform a real-time simulation of an aircraft faster and more maneuverable than any plane ever built—but not Cottrell. Faced with a problem in aerial combat, you assume solvability. You have to; it means survival. Cottrell approached the simulator problem in the same way and began investigating alternatives that would allow fast hi-res graphics generated by heavy computational models.

He started surveying available languages on the Apple and focused on Forth, which had the dynamic flexibility he wanted but lacked floating point calculation capability. Seeking a hardware contribution to the floating point problem, Cottrell's intuition was confirmed by an Alan Winston column in the June 1979 issue of *Call-APPLE* magazine. Winston suggested marrying Forth to the CCS 7811 arithmetic processor board. A whimsical suggestion to most readers, it gave Cottrell insight into the solution of the supersonic Apple challenge.

A kluge of Forth and the math chip powered the Navy VATOL simulator. Ben Mason, Air Force retiree and CCS software developer, and a handful of talented technical friends helped Cottrell expand and improve what has become Applied Analytic's first commercial product.

MicroSpeed is a threaded, stack-oriented, extensible language. The language begins as a small collection of syntactical rules and a dictionary of verbs, or words as they are called in most Forths. Instead of putting a specific set of commands together in convoluted ways as with traditional programming, the MicroSpeed user defines new verb-words where each definition uses only previously defined verb-words put together in grammatically acceptable sentences. A program is essentially a macro verb-word consisting of recursive nests of grammatically consistent verb-word sentences. The user extends the overall system rather than writing programs using the infrastructure of an imposed command set.

MicroSpeed is now used to enhance imaging radar on the NASA space shuttle. In Washington state, MicroSpeed is being taught abstract expressionist painting. Businesspeople are using it for commodity analysis and macroeconomics. Though the fundamental language is the same, each version of MicroSpeed takes on the characteristics given by its user and becomes unique.

Cottrell's product reflects his taste in fast, maneuverable fighter aircraft. The math chip provides the speed, and the user-extendable nature of Forth provides the dynamic flexibility. And just as Cottrell's life has been characterized by his pursuit of new skills and knowledge, MicroSpeed provides a tool that grows and becomes more powerful in solving the specific problems of the user. ◨

Supersonic

from page 119

to three small CRT monitors showing vital information in graphic symbols and alphanumeric readouts. Any one of these images can be projected on an angled piece of glass that's in the pilot's line of sight looking straight out the cockpit windscreen.

The effect is similar to the range finder of a camera where you see the real world image with important symbols and numbers superimposed. If the pilot switches to the targetting subsystem, the HUD symbology graphics are visual enhancements that come from the plane's computer analysis of radar information. The pilot switches on the missile system. When the little enemy plane symbol is maneuvered into the graphic flashing ring kill zone, an automatically fired rocket is guaranteed to hit the mark.

The HUD's graphics are strangely similar to the innocent displays of arcade space games—except what goes on out front of the HUD is anything but innocent. It's kind of frightening, and allusions to *Star Wars* are disconcertingly appropriate, especially when the modern VATOL's flight characteristics are considered.

Today's engine technology provides such thrust and nimble handling that the modern VATOL could perform as if it were out in space, defying gravity. Imagine a dogfight where the plane being chased can kick its tail around, shoot at the chaser, and blast off in another direction, even flying straight up!

A less gruesome image for the HUD is the take-off and landing display. The plane's sensors pick up signals from the airfield or aircraft carrier deck. These signals are converted into a line display depicting the take-off and landing site. Real-time simulation of this take-off and landing HUD was what Sam Cottrell was able to accomplish with a souped-up Apple computer.

"What you see on the monitor is the HUD without the real world on the other side," Rousseau explains. "The right angled, green goal post is the nose hook bar where the plane rests when not in flight. The red lines delineate the landing area and represent the width of the field at two-hundred-foot intervals. These red lines give the pilot a sense of depth during the approach for a landing. The blue line is the horizon. The little symbols above and to the right of the line graphics give the pilot instant information about the pitch, airspeed, and related information crucial to making rapid flight corrections.

"The stick on the left is the throttle, and this amazing one on the right is the pitch of the nose up and down. Since this simulator is for testing take-off and landing, we didn't worry about flying left and right. You blast off, lifting from the landing bar, and fly in a big loop to return and shoot a landing, like this," Rousseau demonstrates as he grabs the sticks.

Rousseau makes it look rather simple. The line graphics fluidly move on the screen giving a dramatic sensation of the hypothetical jet's movement. Rousseau hangs the plane back up on the landing bar, shakes the stick, and invites a fledgling to take over.

Thank heaven it's only a simulator! Taking off is a piece of cake. The green goal posts drop away, the red lines of airstrip slip behind. The newcomer then proceeds to demolish about two hundred million dollars worth of hypothetical fighters—undoubtedly qualifying him as an enemy ace. He smashes into the runway. He flies upside-down into the ocean. He even ditches into the ocean tail-first. The stall and balance-a-pencil-on-the-fingertip procedure escapes him.

Each time he crashes, he is given a mathematical readout telling exactly where he died, at what speed, and in what position the plane was at impact. This Apple is obviously doing a lot of complex aeronautical calculations at high speeds to provide such a sensation of fluid movement.

"Don't be distressed about your performance. You should have seen some of the pilots we tested," Rousseau offers the downcast novice; "in fact, after Sam dropped the simulator off, we flew it unsuccessfully for a week. We called Sam and told him something must be wrong, that the plane was too sensitive during landing. The balancing blasts kept causing a rapid pitchover. Sam assured us that he had delivered a simulator that mathematically matched the performance characteristics of the contract."

The result is a case where a simulator can provide design improvements before a prototype is ever built. The pitchover was the result of a flight dynamic that was traced to a portion of the mathematical model of the performance characteristics, dictated by the contract. Cottrell derived a mathematical compensation for the aerodynamic characteristic, modified the simulator software accordingly, and the plane became flyable.

You might think this is cheating, but this is exactly how such a problem would be addressed in a sophisticated jet fighter. In effect, the actual plane would be built with its on-board computers programmed to analyze and compensate for this pitchover characteristic automatically.

"There are planes in service today that wouldn't be flyable if it weren't for the on-board computers," Rousseau goes on. "Sam's simulator helped us improve this plane's design. In turn, this will help make modern VATOL a more serious contender for continued research to provide us with the best possible defense equipment. It may be a while before this VATOL plane gets built, if ever. But we've taken a very small budget and answered a lot of important questions using the Apple. VATOL has a brighter future now."

The science fiction future is here today.

Like a Buck Rogers sci-fi serial, the story doesn't end here. Sam Cottrell, president of Applied Analytics and creator of the VATOL simulator, is a multitalented man, and there's a parallel between his own creative style of problem solving and the characteristics of his commercial product, the Micro-Speed Language Development System. A visit with Cottrell accompanies this article. ❑■

apple in small letters:

a survey of lower-case adapters

BY JEFFREY MAZUR

While most programming and applications for a personal computer do not require lower-case capability, this feature is nearly essential for word processing. One of the first complaints about the Apple II was its lack of lower-case characters.

Solving this problem is relatively simple, however, and Dan Paymar was the first to offer the necessary hardware commercially. His Lower Case Adapter opened the door to develop sophisticated word processors, communications modules, and other software that could take advantage of the Apple's newly acquired power.

Several different approaches to adapting the Apple for lower case have since been developed. Some involve hardware modification, and others use software to perform the same task. This is a great example of the dichotomy of computer systems: most problems can be solved either with hardware or software. Before getting into the details, let's define the three areas where changes must be made to allow lower-case characters on the Apple.

Any computer can be simply described as the connection of four functions: input, memory, processing, and output. In the case of a typical Apple, input would refer to the keyboard; memory would be in the form of RAM in the computer or floppy disk storage. The Apple's 6502 CPU along with the system Monitor, operating system, and any application program constitute the processing function of the computer. Output is generally displayed on the video screen or possibly via a printer.

The Apple was designed to work with upper-case letters in all four areas, and it does this well. However, only the memory portion of the computer is capable of handling lower case without modification. Some changes must be made to the other three functions to allow complete upper-case/lower-case operation.

Lower Case Screened. Lower-case display by the Apple has a two-phase history that begins in the way video is generated by the computer. Basically, what is displayed on the video monitor starts off as a block of memory ($400 to $800) in the Apple's RAM. One byte of memory corresponds to each character seen on the screen. A special portion of the Apple's circuitry—the video generator—constantly reads this area of RAM and translates it into the rows of dots that make up each line of the display. The actual conversion from byte (eight-bit) data to character (5x7-dot) data is done by a single device called a character generator ROM.

In the original Apple II, the specific chip used was a 2513, which is a 2,560-bit ROM that can decode sixty-four different characters. This is just enough to include the alphabet, numbers, and necessary punctuation, but not lower-case letters. Fortunately, there is another 2513 ROM chip available, programmed with lower-case letters instead of upper-case.

These two chips can be connected in the circuit, and you can switch between the two, depending on the case desired. In the early days, this was accomplished by actually soldering the two ICs together pin-to-pin, except for one select pin on each chip.

Then Dan Paymar designed the Lower-Case Adapter, which made the job much simpler. Paymar's original adapter was a small piggyback board that contained the extra 2513 and one logic chip for selection, making the whole modification a matter of plugging things in and out.

The first dilemma that arose from this scheme was how to switch between the two chips. Since the normal Apple had only sixty-four characters, this required only six bits (two to the sixth

equals sixty-four) to represent any character, leaving two extra bits in each byte of display memory unused. The Apple's designers took advantage of this by allowing inverse and flashing modes for all characters.

The Paymar board maps the normal ASCII lower-case letters to display correctly but does not allow for inverse or flashing lower-case characters. It does this by selecting the lower-case 2513 whenever the top three bits are all ones.

Newer lower-case boards replaced the dual 2513s with a 2716 programmable ROM allowing custom-made character sets. Expanded connections to other parts of the Apple motherboard gave access to the complete character cell (the first and last column of dots for each character had been left blank to leave a space between), making it possible to define special symbols and line drawing graphic sets.

Although it is impossible to have complete normal, flashing, and inverse sets, which would require at least nine bits per character, it is possible to have a complete normal set and an inverse set. Some of the newer boards offer a switchable selection between flashing upper case or inverse lower case. This is important because some programs, such as the *Pascal Editor*, use an inverse cursor. When this cursor is placed over a lower-case letter, it will change it to some weird punctuation or digit if provisions aren't made for inverse lower case.

Complicating matters more, on its seventh revision of the motherboard (about July 1980), Apple changed the video generation circuits considerably. Enter phase two. With the new boards, all that's required to generate lower case is replacement of the character generator ROM, which is now 2716 compatible. None of the old lower-case adapters work with the newer boards, but, with them, obtaining lower case is as simple as changing one chip.

Or is it? The answer is a qualified yes. Unfortunately, the problem with inverse lower case is back again. Therefore, most of the sophisticated lower-case adapters have been modified for the newer revision boards.

No discussion on lower case would be complete without mentioning an altogether different approach, using the Apple's hi-res graphics display. Several RAM and ROM versions of programs to add lower case this way have appeared on the market, leading to the Keyboard Filter firmware for the Mountain Computer ROMPlus+.

This had the advantage of requiring no extra hardware (unless you count the ROMPlus+ and the firmware chip), no modification to the Apple (unless you add the shift-key wire), and mixed text with graphics. Also added was the ability to have colored letters, multiple user-defined fonts, keyboard macros, and a few more goodies.

The price, however, was too great: loss of 8K of RAM for the graphics display, incompatibility with certain programs (for example, those that call directly upon the video screen memory), very slow scrolling speed, and other disadvantages.

Keying in Lower Case. Once the Apple is capable of displaying lower case, there needs to be some way of entering it from the keyboard. As it stands, all letters typed on the keyboard come out as upper case; the shift key has no effect, except on a few special characters. The simplest solution to this problem is to write an input routine to handle upper/lower case shifting.

Usually, the escape key or the forward arrow key is used to signify a shift operation. Pressed once, it will capitalize only the next letter; pressed twice in a row, it activates a shift lock, capitalizing all letters until it's pressed once more. This software approach has the advantage of requiring no hardware modification to the Apple. However, the two-step shifting operation does take getting used to.

By adding one small wire from the keyboard's shift key to the game I/O connector (or by using the ROMPlus+), it's possible to regain use of the shift key as it would work on a typewriter. This modification still requires some software to

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

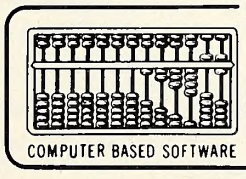
SEND FOR FREE CONTROL PAGE

* 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



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

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)
Add $1.75 for shipping and handling.

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)

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.

read the shift key status from the I/O port, but this is incorporated into many existing programs. This is a good mix between hardware and software solutions to the problem.

For the purist, it would be desirable to modify the Apple's keyboard so that it can generate lower-case characters. Several items have appeared recently that accomplish this. Since there is no real shift-lock key, this function still has to be simulated with some other key like ctrl or reset.

The remaining ASCII characters unavailable on the standard Apple keyboard are also usually offered by pressing various combinations of keys. One such keyboard expander even has a typeahead buffer that stores what you type until the computer is ready to read it.

Lower-Case Ability Must Be Processed. Even after lower-case keyboard and display adapters have been added to the Apple, no changes will be evident when the computer is first turned on. This is because the Monitor routine, used by the operating system and Basic to read the keyboard, converts all lower-case input into upper case. This isn't a problem if lower case is used only when running a word-processing or text-editor program that has its own software to read the keyboard. However, to include lower case in Basic programs requires a new input routine to be added to the operating system. Many lower-case adapters come with software that allows these amenities. However, although lower-case text may be included in a Basic remark or print statement, all other program text or commands must be in upper case for the Apple to recognize them. If you're particularly enterprising and own a language card, even this obstacle can be overcome by making appropriate patches to the command processors in Basic and DOS.

Now . . . to the Hardware: Dan Paymar's Lower-Case Adapters. The Paymar Lower Case Adapter for older Apples (LCA-1) is a small plug-in board and header socket connected by a wire. Installation involves removing two ICs, replacing them in sockets on the board and header, and then placing the board and header into the empty sockets on the motherboard. The LCA-2 for newer Apples involves changing only one IC. Instructions are adequate, and the patches needed to use lower case with Pascal and *Apple Writer* are given.

For the price of the disk, a software support package is available called *DICE* (Dan's I/O Control Enhancements). This software allows entry of lower case using the escape key. The size of the cursor identifies whether the upper-case or lower-case mode is in effect. Normal cursor movements (given up by using the escape key for shifting) are duplicated by new control key sequences. On top of all this, there are a lot more features available with just a few keystrokes. (Setting text/graphics modes, monitor entry, and user jump commands are a few.)

The character set uses pushed up lower-case letters to give descenders the illusion of descending. Whether or not you like the look of some characters, there's no way to change them because they're in ROM. The Lower-Case Adapters carry a one-year warranty and sell for $60 (LCA-1), $50 (LCA-2), and $5.00 for the *DICE* diskette.

Dockside Computer Unitext. Installing the Universal Text Display Module is a little more complicated than the Paymar chip. For the Model A (Revision 6 or earlier), a total of five separate pieces are inserted at various locations on the motherboard, including the game I/O socket. An additional wire is included for making the shift key attachment, if desired. (This involves a small bit of soldering.)

A comprehensive and well-written owner's manual describes the many features and operation of the Unitext in Basic and Pascal. Complete software is included on diskette for both languages, including automatic switching to the inverse lower-case mode when in Pascal.

The basic software uses a control-Q as a shift key (unless the shift-key wire has been added), with automatic return to shift lock after each line is entered. The control-Q is also used as a lead-in key to produce some of the extra ASCII characters not found on the Apple keyboard.

The character set is stored in a 2716 EPROM and is thus changeable. However, the set you get is good-looking. It provides true descenders while keeping lower-case letters on the same baseline as upper case. The only tradeoff here is that, whenever a lower-case descender is just above an ascender from the character below on the next line, there's a collision—that is, there's no space between the two letters. Most of the time these collisions are not objectionable, and in most text, the collisions occur less than two percent of the time.

Two models are available for any Apple; the price is $80.

Lazer Microsystems Lower Case +Plus. The Lower Case +Plus is unique in that it's been designed to work in any revision Apple. A set of printed circuit pads are arranged so that connection to one side configures the board for early Apples, while connection to the other side converts it to work with new Apples. Although the board will usually come set up for the particular revision needed, it's nice to know it can be changed easily to go in a different machine.

Another difference in this unit is the way it's installed. Although a total of four ICs are removed from the Apple's motherboard, all those that go back plug into sockets on the Lower Case +Plus. Then the entire board is placed into the Apple in one piece, with pins going into

all four empty motherboard sockets at the same time. It takes a bit of pushing, but the end result is a very neat piggyback module with no extra wires.

The character set supplied does not have true descenders, and even some of the normal characters have been changed. Fortunately, the board is 2716 compatible, so any character set (in fact, two complete 128-character sets) can be programmed.

Also, the Lower Case +Plus format is compatible with the Keyboard Filter and ROMPlus+. Therefore, if you like, you can use the font editor program supplied with the Keyboard Filter to examine and make changes on the Lower Case +Plus character set. After everything is just the way you want it, you can burn the primary set, along with a secondary set, into EPROM.

Other features of the Lower Case +Plus include alternate character set selection, inverse lower-case mode, complete 7x8 font size access, and an expansion socket for future products. Included on disk are the *LSIE* (Lazer Systems' Input Editor) software and a program to patch *Apple Writer* for lower case. The *LSIE* allows lower-case entry using the escape key for shifting. Normal cursor movements are still supported using control keys. The Lower Case +Plus sells for $65.

Revision 7 Character Generator Replacement. Since lower case on the newer Apples is only a ROM away, several companies have begun to offer lower-case adapters, generators, or whatever for $25 to $30. This is a no-frills approach, but the low cost makes it attractive. The only differences among units in this category seem to be in the character set appearance and in whatever software is included. Deltrex has a low-priced ROM, for instance, but some of the character formations are not impressive. Legend Industries sells a ROM with a set including true descenders. Simple instructions and a very crude lower-case entry routine are also included. The LJK Enterprises Lower Case Character Generator at $35 has true descenders and software for direct input of lower case. Lazer Microsystems has the Lower Case +Plus II, which is simply a ROM and includes their *LSIE* input software, and sells for $30. The Paymar LCA-2 at $50 also comes under this heading.

Lazer Microsystems Keyboard +Plus. This clever board mounts on the inside wall of the Apple's case. Velcro strips enable you to remove the board if you need to. Installation involves one IC header and soldering two wires to the keyboard. After all this is done, the keyboard takes on several new dimensions.

First, the reset key becomes the shift-lock/unlock control with the keyboard defaulting to shift-lock for normal Apple operation. After the first press of the reset key, all letters typed will be entered as lower case unless the shift key is pressed simultaneously—just like a real typewriter. Reseting the computer is now performed by the control-reset sequence.

Next, by pressing the control key in conjunction with most of the nonalphabetic keys, you can obtain the rest of the ASCII set normally unavailable on the Apple keyboard.

Finally, this board contains a keyboard typeahead buffer. This means that, while the computer is busy processing (loading a tape, going to the disk, sitting in a loop, and so on, you can keep on typing and the buffer will store up to sixty-four characters until the computer is ready to read it. This can greatly speed up typical operations with the computer. There are a few quirks to watch out for, but this feature is quite nice to have.

A special clear-buffer command is included to empty the buffer. This is vital to preserve compatibility with nonbuffered Apples—for example, using the control-C to interrupt a Basic program.

The biggest problem the keyboard buffer may cause is that some programs—most notably games—won't function correctly. As a temporary solution for gamers, Lazer Systems has come up with a slight modification, adding a bypass switch to the keyboard buffer. If you already own a Keyboard +Plus or are contemplating buying one, be sure to write the company for details. The Keyboard +Plus retails for $100.

And Still More Hardware. Two other units were unavailable for evaluation. The Keyboard and Display Enhancer from Videx sells for $129, and a keyboard encoder board from Basis sells for $125.

Choosing the Right Lower-Case Hardware. Before considering the purchase of any lower-case hardware, you must determine which type of Apple you have. The easiest way to do this is by opening the cover of your Apple and looking at the area of the motherboard along the left edge next to the letters D, E, and F. If there are three black boxes labeled "memory select," then you have a revision 6 or earlier motherboard.

Next, you should consider whether direct lower-case input will be needed. Your pocketbook usually makes this decision, since lower-case display may cost as little as $25, but full lower-case operation will set you back at least $125. If your only use of lower case will be with commercial software packages such as word processors, you won't require input hardware. Just make sure the software you choose supports lower-case add-ons and the hard-wire shift-key modification if you want it.

Finally, don't forget that all eighty-column boards include lower case. ■

ON-LINE systems *is pleased to make available to you, the following Assembly Language development aids:*

LISA 2.5

Assembly Language Development System for Apple Computers.

LISA 2.5 is a proven method for 6502 Assembly language development that has been in use in the homes and offices of both novice and professional programmers for over two and a half years. LISA 2.5 is by far the most popular assembler ever developed for the APPLE and is endorsed by masters of the field such as Steve Wosniak, Jay Sullivan, Ken Williams, Olaf Lubeck, Mark Polzarski, and of course Randy Hyde.

LISA 2.5 is complete and easy to use. The commands for programming procedures are easy to learn and the assembly is quick (20,000 lines a minute). The LISA 2.5 assembler uses over 30 psuedo opcodes (which turn your APPLE II into a 16 bit machine, requiring less code to perform a desired task); more extended mneumonics (a greater memory aid) and more commands which allow the flexible use of DOS.

LISA 2.5 is now available for $79.95 at your local computer store and runs on any 48K APPLE II/II+ with or without a 16K memory extension board.

SPEED/ASM

IMAGINE being able to simulate common basic statements and executing them in Assembly language! Well, it's this easy . . .

```
BASIC:                  SPEED/ASM:
    10 FOR I=1 to 10        JSR FOR
    20 PRINT I              ADR I,1,10
    30 NEXT I               JSR PRTINT
                            ADR I
                            JSR NEXT
```

SPEED/ASM is a powerful new assembly language program development tool which gives the BASIC programmer power to build complex assembly language programs. SPEED/ASM is a set of subroutines that can be used to simulate BASIC, in developing an assembly language program. Thirty-nine pages of well written, informative documentation explain how you can use SPEED/ASM, (and your knowledge of BASIC) to make programs of professional quality, quickly & easily. SPEED/ASM is an independent Assembly language subroutine package. Due to this, SPEED/ASM Programs produced on an Apple Computer would be compatable with ATARI SPEED/ASM and vice-versa. Also, it is not necessary to own a "LISA" assembler in order to operate SPEED/ASM.

Remember:
SPEED/ASM *IS* Assembly language (that the basic programmer can easily understand). SPEED/ASM is available for $39.95 at your local computer store, and runs on any 48K Apple II or II+

To Order Direct

Send purchase price plus one dollar for shipping to:

ON-LINE SYSTEMS

36575 Mudge Ranch Rd. - Coarsegold, CA 93614

209-683-6858

by Randy Hyde
author of "Assembler Programming For Your Apple"
Visa, Master Charge, C.O.D. or Checks Accepted.

The Controller Even You Can Make

BY SILAS WARNER

A fast-growing section of the small computer market today is in controllers—small computers with inputs and outputs specially designed to sense and control electrical equipment. The average packaged home microcomputer, though, is designed to display its output on a TV screen or monitor, sound it through a speaker, or record it on a cassette or disk. It is designed to take its input from a keyboard, cassette, disk, or handheld controllers. These are enough for interacting with people, but not very good for connection to other electronic equipment.

Apple II computer users are luckier than most in making these connections, since the Apple II comes with a game I/O connector. This connector has four paddle connections that sense the resistance across them. It also has three switch connections that sense switch closings and four annunciator outputs that send TTL signals from the computer. But seven inputs and four outputs are still very few when you want to control, say, a multizone security system or a robot.

The Apple also has seven peripheral slots—connectors designed to feed Apple I/O signals to special devices. But the signals on these slot pins are too complex for the beginner to wire. What is needed is a simple circuit to convert the complex Apple signals into outputs that can be simply wired to the devices to be sensed and controlled.

Enter the APMOD, a simple peripheral card manufactured by Connecticut MicroComputer. The APMOD is designed to connect an Apple to CMC's AIM16 line of analog-to-digital converters. But, to an experimenter, it's a simple interface card that generates eight TTL-level outputs and accepts eight TTL-level inputs. Wiring those outputs and inputs to other electrical devices is the subject of this article.

Photo 1 shows the two sides of the APMOD card: the left, or circuit, side and the right, or component, side. Note that there

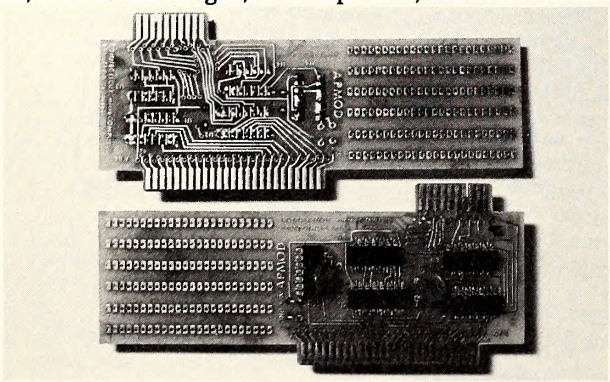
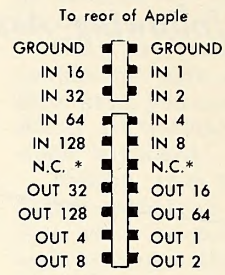


Photo 1.

are two sets of pins projecting from the two long edges of the card. The longer pin-set goes toward the bottom and plugs into the Apple's peripheral slot. (Any slot can be used—even slot 0—though that slot probably should be kept free for language or ROM cards.)

The shorter pin-set on the top is where the input and output signals are connected. Diagram 1 shows the layout of the signals at this pin-set. There are two ground pins and two unconnected pins. There are also eight output pins, which can be switched from low to high logic level, and there are eight input pins, which sense the TTL logic level of the wires feeding them.



*Moy be connected to +5 volts: see Photo 2

Diogrom 1
APMOD pin-set loyout (top view)

TTL Logic Levels. Up to now, I've been tossing the term *TTL logic level* about pretty freely. Before we go any further, let's stop and find out what that means. TTL stands for "transistor-transistor logic," an older but still popular form of integrated circuit used in digital electronics, including some parts of the Apple II. In TTL logic, a *low* level is 0 volts with respect to ground. (Any voltage below 0.5 volt will do.) A *high* level is +5 volts with respect to ground (anything between three and five volts will do.) Never put a negative voltage or a voltage above 5.0 volts on a TTL circuit; chances are, you'd blow out the circuit!

In general, TTL circuits work by grounding low outputs rather than sending power through high outputs. A TTL input not connected to anything will be assumed by most TTL circuits, including the APMOD, to be in the high state. This also means that though an average TTL circuit can send twenty-five milliamps to ground in the low state, it cannot provide more than about ten milliamps at +5 volts in the high state. Later, we will discuss special TTL circuits designed to handle higher voltages and currents.

Connecting to the APMOD. The upper pin-set is made to accept a twenty-conductor card edge connector, with the pins on .010-inch centers. This connector is not readily available, but a forty-pin connector with the proper spacing is available at most Radio Shack stores as part number 276-1558. You can also get five feet of twenty-conductor ribbon cable to fit this connector as Radio Shack part number 278-0770. You can attach the ribbon cable to one half of this connector only, and then press only this half onto the APMOD; or you may insert two pieces of ribbon cable into the two halves of the connector and run them to different pieces of equipment. All you need to do to disconnect one unit and attach the other is turn the connector 180 degrees on the APMOD—which can be done while the computer is running.

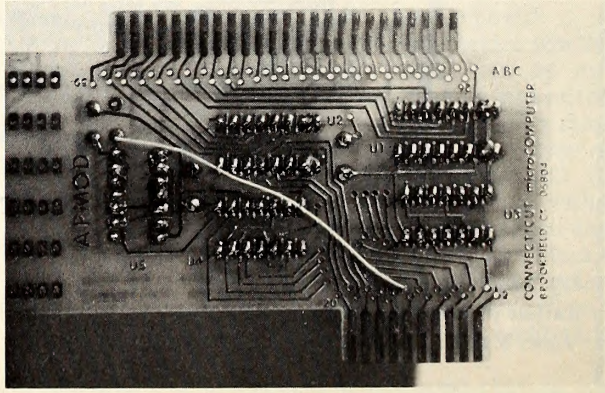


Photo 2.

As delivered, the APMOD does not have a power output. There are only the two ground pins and the two unconnected pins. The AIM16 unit designed to connect to the APMOD has its own built-in +12-volt power supply, which is run to the two unconnected pins on the APMOD. When the APMOD is used without AIM16 devices, it is convenient to have a +5-volt power supply as well as a ground.

Photo 2 shows a wire run from a +5-volt tie point on the APMOD to a hole joined to the two unconnected pins. With this wire in place, the two pins are supplied with +5 volts from the Apple's power supply. Shorting these +5-volt pins to the ground pins can short out the Apple's power supply. While I am assured by Apple's engineering department that this cannot harm the computer, it causes a frightening array of symptoms—squealing power supplies, lost data, and TV blackouts—that will condition you pretty quickly to avoid future shorts!

Testing the APMOD. Unlike most Apple peripheral cards, the APMOD cannot be activated with a PR# or IN# command. To control the card, you must peek or poke to its address. The address of the APMOD card depends on the slot in which you insert it. An APMOD card in slot 0 has the address -16256. The address of an APMOD in any slot is given by:

ADDRESS = (16 * the slot number) - 16256.

For instance, the address of an APMOD in slot 3 is (16 * 3) - 16256, or 16208.

Poking a value into this address will send the various output lines high or low, depending on the binary digits of the number you poke. For instance, to set the output pins numbered 128, 32, and 8 high and the rest low, you poke address, 128 + 32 + 8. To set all output pins high, you poke address,255. To set all output pins low, you poke address,0. The eight input lines also appear as a binary number, which appears as peek(address). For instance, if peek(address) = 48, all the APMOD input pins except 32 and 16 are connected to ground. (48 is 32 + 16.) If all the pins are connected to ground, peek(address) will be 0. If all the pins are connected to +5 volts, or disconnected, peek(address) will be 255.

The example program in listing 1, written in Applesoft Basic, can set any desired combination of outputs high or low and then display the state of all eight inputs. The routine at 1000

```
 10 DIM IN(8),OUT(8)
 20 TEXT : CALL - 936
 30 REM
 40 REM GET APMOD SLOT NUMBER
 40 REM
 60 PRINT "SLOT #";: INPUT SLOT
 70 ADDRESS = (16 * SLOT) - 16256
 80 PRINT "ADDRESS - ";ADDRESS
 90 REM
100 REM PRINT NUMBERS
110 REM
120 X = 128
130 VTAB 8
140 FOR N = 1 TO 8
150 PRINT X
160 X = X / 2
170 NEXT N
180 PRINT "NUMBER"
190 REM SET LEFT MARGIN TO 10
200 REM      THEN
210 REM GET 8 OUTPUT VALUES
220 REM
230 POKE 32,10: POKE 33,30
240 VTAB 6: PRINT "OUTPUT": PRINT
250 FOR N = 1 TO 8
260 INPUT A$
270 OUT(N) = 0
280 IF A$ = "HIGH" THEN OUT(N) = 1
290 NEXT N
300 REM
310 REM FORM OUTPUT NUMBER
320 REM AND POKE IT INTO APMOD
330 REM
340 GOSUB 1000: PRINT NUMBER;" "
350 POKE ADDRESS,NUMBER
```

```
360 REM READ INPUT NUMBER
370 REM AND DECODE IT
380 NUMBER = PEEK (ADDRESS)
390 COPY = NUMBER: GOSUB 2000
400 REM
410 REM DISPLAY INPUTS ON
420 REM RIGHT SIDE OF SCREEN
430 REM
440 POKE 32,20: POKE 33,20
450 VTAB 6
460 PRINT "INPUT": PRINT
470 FOR N = 1 TO 8
480 A$ = "LOW"
490 IF IN(N) = 1 THEN A$ = "HIGH"
500 PRINT A$
510 NEXT N: PRINT COPY;" "
520 GOTO 230
1000 REM
1010 REM ROUTINE TO PACK 8
1020 REM 0S OR 1S INTO 1 NUMBER
1030 REM
1040 X = 128:NUMBER = 0
1050 FOR N = 1 TO 8
1060 IF OUT(N) = 1 THEN NUMBER = NUMBER + X
1070 X = X / 2
1080 NEXT N
1090 RETURN
2000 REM
2010 REM ROUTINE TO DECODE
2020 REM NUMBER INTO 8 0S OR 1S
2030 REM
2040 X = 128
2050 FOR N = 1 TO 8
2060 IN(N) = (NUMBER > = X)
2070 IF IN(N) THEN NUMBER = NUMBER - X
2080 X = X / 2
2090 NEXT N
2100 RETURN
```

Listing 1. General APMOD I/O driver program.

converts the eight values of the array *out* into a single variable for poking to the slot address. The routine at 2000 converts the peek of that slot address into eight values in the array *in*.

To use the program, first type in the slot number of the APMOD. Then type high or low at each of the eight prompts. When the APMOD output lines are set, you'll see a display of the state of the input lines. To quit the program, press ctrl-c and then type TEXT to return the prompt to the left side of the screen.

Driving LEDs with the APMOD. One of the simplest tasks the APMOD can do is turn on or off a number of lights. Diagram 2 shows how to wire an output pin of the APMOD to a standard three to five volt light-emitting diode (LED). Wire

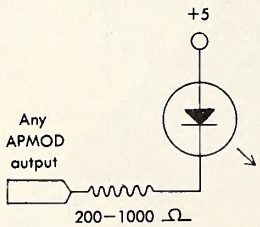


Diagram 2
Wiring an LED

the anode (short) lead of the LED to +5 volts and the cathode (long) lead to the output pin of the APMOD through a resistor of from 200 to 1000 ohms. The resistor is needed to prevent high currents from damaging the LED, and a higher resistance means a dimmer light. Eight LEDs can be connected in this way, one to each output pin.

After wiring the lights, run the program in listing 1. Each light should turn on when the associated output pin is set low and go out when the pin is set high. Use the subroutines at 1000 and 2000 in listing 1 to control the lights from your programs.

Driving a Seven-Segment Display. A common task for the APMOD is to drive a seven-segment LED display—for example, to display a number on a console remote from the Apple's screen. A common-anode LED display can be wired in

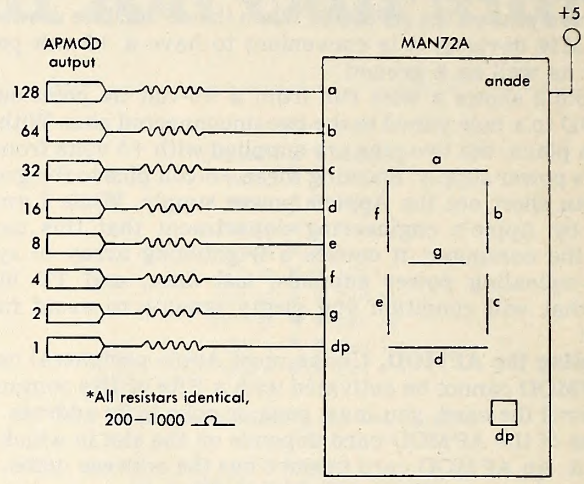


Diagram 3
Wiring a seven-segment display

just the same fashion as separate LEDs. Diagram 3 shows the complete wiring for a common-anode LED display such as a MAN72A. The resistors can be any value from 200 to 1000 ohms but should be all the same value.

The program in listing 2 drives the LED display to produce sixteen hexadecimal numbers in sequence. The data

```
10 DIM NUM(16)
20 TEXT : CALL - 936
30 REM
40 REM GET APMOD SLOT NUMBER
50 REM
60 PRINT "SLOT #";: INPUT SLOT
70 ADDRESS = (16 * SLOT) - 16256
80 PRINT "ADDRESS - ";ADDRESS
90 REM
100 REM PREPARE NUMBER TABLE
110 REM
120 FOR N = 0 TO 15
130 READ NUM(N)
140 NEXT N
150 DATA 3,159,37,13
160 DATA 153,73,65,31
170 DATA 1,9,17,193
180 DATA 99,133,97,113
190 REM
200 REM HEXADECIMAL DISPLAY
210 REM
220 FOR N = 0 TO 15
230 POKE ADDRESS,NUM(N)
240 REM
250 REM DELAY LOOP
260 REM
270 FOR I = 1 TO 1000: NEXT I
280 NEXT N
290 GOTO 220
```

Listing 2. Seven-segment numeric display program.

statements 150 through 180 contain the numbers that are poked to the APMOD address to turn on the proper segments if the display segments are connected as in diagram 3. If you connect the display segments differently, you may have to change these numbers to get good-looking figures on your display. Or you may wish to design your own combination of seven segments, which can be converted into a number and displayed at your command.

High-Voltage and High-Current Circuits. Never, *never* connect any voltage higher than +5 volts to the APMOD—not even +6 volts! And never, *never* connect the ground of the APMOD to any ground that might come in contact with household voltages. Doing so may ruin your APMOD, your computer, and you.

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.

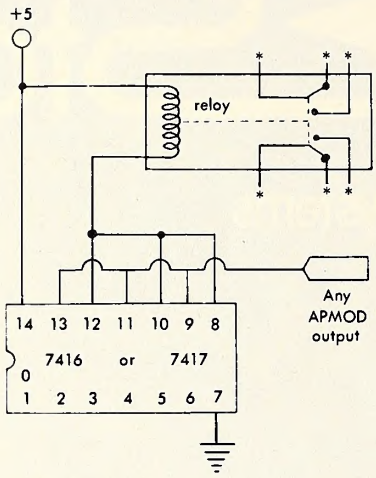
SDS
southwestern data systems

P.O. Box 582-S, Santee, CA 92071 • 714-562-3670

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

To control high-voltage circuits from the APMOD, you can use a relay driven by five volts. A relay with a five-volt coil, capable of controlling three amps at 125 volts, is available from Radio Shack as part number 275-215. But this relay draws 100 milliamps, or about four times the current the APMOD can safely deliver.

To provide the high current, a buffer circuit must be used between the APMOD and the relay. The 7416 and 7417 circuits contain six buffers, each able to stand 40 milliamps. Connecting three of them in parallel, as in diagram 4, provides enough current capacity to operate the relay safely. If you wish, you



*Wire reloy contocts to circuit to be controlled

Diogrom 4
Wiring o relay

can use the other three buffers on the same chip to drive another relay.

The difference between the 7416 and 7417 buffers is this: the 7416 sends current to ground and operates the relay when the APMOD's output pin goes high. But the 7417 operates the relay when the output goes low. Because of this, the 7416 is called an inverting buffer, and the 7417 is a noninverting buffer. Pick whichever you need for your particular application.

Sensing Switches and Keyboards. So far we have only discussed output from the APMOD. The APMOD also has eight input lines that can be used to sense the positions of switches. Diagram 5 shows the simplest way to sense the closing of a switch. Remember, TTL circuits like the APMOD assume that unconnected inputs are in the high state. Closing the

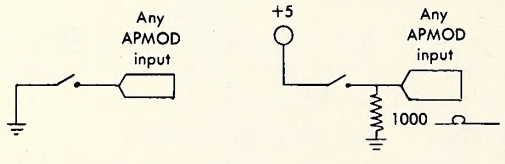
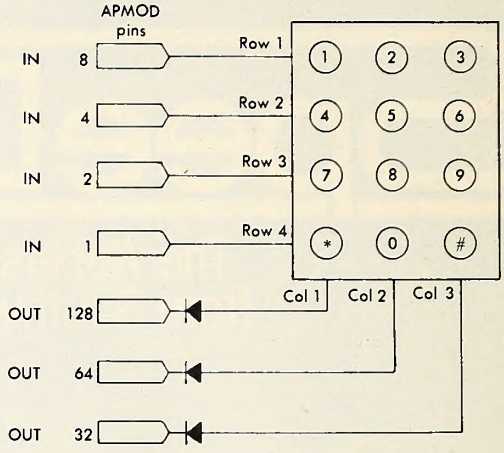


Diagram 5
Wiring switches to inputs

switch in diagram 5(a) sets the APMOD input pin low. In diagram 5(b) the 1000-ohm resistor keeps the input pin low until the switch is closed; then +5-volt power forces the pin high. The logic state of each pin can be sensed by peeking the APMOD address and using the input routine at 2000 in listing 1.

Since a keyboard is just a lot of switches, it would seem easy to connect a keyboard to the APMOD. But there are some problems. For one thing, there are only eight inputs to the APMOD, and a keyboard may have sixty-four or more switches on it. To sense sixty-four switches with the system of diagram 5 would require eight APMODs, filling up all the slots in the Apple! We can use only one APMOD by sensing only eight switches at a time but scanning through the switches fast enough to pick up even the briefest closure.

Diagram 6 shows how a keyboard is scanned with a twelve-key numeric keyboard, available as Calectro part number E2-146. This keyboard has four row lines and three column lines.



Diogrom 6
Wiring o sconned keyboord

Pressing a key connects a row line to a column line. The row lines are connected to four APMOD input pins so that a key pressed on a grounded column line will send an input low. That tells us a key on one row is pressed, but not which key on that row.

To tell us that, the column lines are connected to three of the APMOD's output pins through 1N914 switching diodes. As long as an output is low, its diode passes current from the column to ground. If a key is pressed in that column, the row line connected to the key will also go low. But when the output goes high, the diode blocks current, disconnecting the column line and any keys connected to it.

The program in listing 3, written in Applesoft Basic, shows how the keyboard is scanned. Normally, all three column lines are grounded. When a key is pressed, one of the APMOD inputs goes low. The program then sequentially sets each of its outputs high. When the column with the pressed key is sent high, its signal disappears, telling the program what column the key is on. Which input line is grounded tells what row the key is on. The program then looks up the key name in the data statements and prints it.

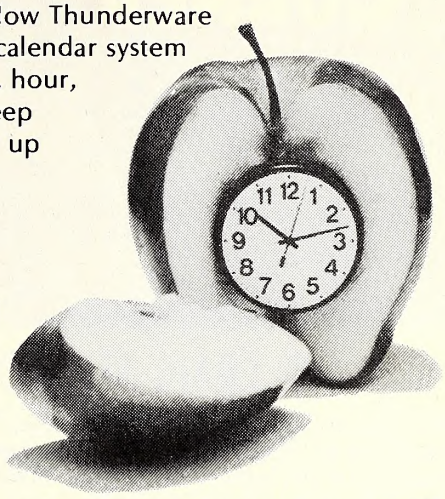
This principle can be expanded easily to cover eight column

```
 10 DIM IN(8),OUT(8)
 20 TEXT : CALL - 936
 30 REM
 40 REM GET APMOD SLOT NUMBER
 50 REM
 60 PRINT "SLOT #";: INPUT SLOT
 70 ADDRESS = (16 * SLOT) - 16256
 80 PRINT "ADDRESS - ";ADDRESS
 90 REM
100 REM SET ALL 3 LINES LOW
110 REM
120 POKE ADDRESS,0
130 REM
140 REM WAIT FOR A KEYPRESS
160 IF PEEK (ADDRESS) = 255 THEN 160
170 RESULT = PEEK (ADDRESS)
180 REM
190 REM NOW SET ONE COLUMN
200 REM AT A TIME HIGH AND
210 REM SEE IF RESULT CHANGES
220 REM
230 X = 32:COLUMN = 0
240 FOR N = 1 TO 3
```

GOTO 142

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!

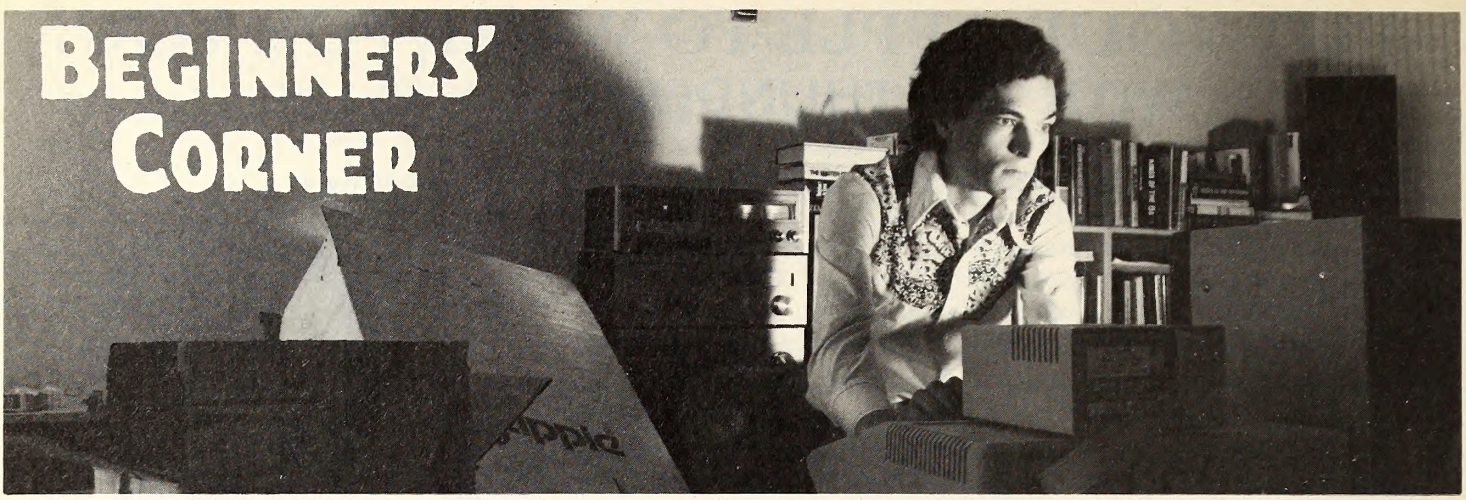
Suggested retail prices:

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.



BY CRAIG STINSON

When you run a program on your Apple, the computer does essentially three things. It takes input of some kind from you, processes the input according to your instructions, and then provides a form of output. The output may be to your television screen or monitor, to a printer, to a cassette tape or floppy disk, or to some other device, such as a modem or paper tape punch. Similarly, the input can be from various sources—tape, disk, modem, or whatever. This month's column will be devoted to the most commonly used input device for the Apple—its keyboard.

The first time you saw an Apple you may have been struck by the fact that it looks a lot like an oddly shaped electric typewriter. Indeed the front end is most typewriterlike, with the same familiar layout of keys and numbers that we all came to know and love in the ninth grade or thereabout.

On closer inspection, however, the Apple keyboard proves to have a few oddities like *esc* and *ctrl*, as well as the usual alphabetic and numeric (auf computerisch, that's alphanumeric) keys. We will be talking presently about those oddball keys and what they do for the Apple, but first we should discuss what happens when any key on the keyboard is pressed.

The 6502 understands only zeros and ones. So what happens when you press the key marked Q? Naturally, the circuitry behind the keyboard translates Q—or any other keystroke—into a combination of zeros and ones.

Toss a Coin. This array of zeros and ones gets stored as a byte of information in the Apple's random access memory. A byte, you may recall, is eight bits, and each bit is a switch that can be in either of two states, representing zero or one.

If you have eight coins, each with a head and a tail, how many permutations of heads and tails can you arrange? Without going into mathematical theory, the easiest way to answer this question is to look first at a single coin. Obviously, one coin can be in either of two conditions—head up or tail up. Now add a second coin. For each of the two possible conditions of coin one, coin two may be showing either its head or its tail. So now we've got four unique arrangements—head-head, head-tail, tail-head, and tail-tail.

If you add a third coin, you'll find you can come up with eight permutations. What you can see from this inductive analysis is that each time a new element is added to the array, the number of unique possibilities doubles. If you do the arithmetic for eight coins, you'll find that you can arrange them in 2x2x2x2x2x2x2x2 or 256 different ways.

So it is with the eight bits in a byte. There are 256 different possibilities for arranging the zeros and ones of a byte. This fact has enabled computer scientists to develop systems for encoding all the different letters, numbers, and punctuation that a person can enter at a computer keyboard.

The code used by the Apple—and by many other computers as well—is called ASCII (pronounced askee), which stands for American Standard Code for Information Interchange.

It may occur to you that 256 code possibilities is a little more than is needed to cover all the letters of the alphabet—both capital and lower case—as well as the numbers and punctuation. In fact, the Apple only uses seven bits for ASCII codes. The eighth bit has a different function: it tells the computer whether a key has been pressed.

Does the Left Byte Know What the Right Byte's Doing? What? You mean the computer doesn't know if a key has been pressed? In a sense it does, and in a sense it doesn't. You could say that that particular byte of memory—the one that holds keypress information—"knows," because it is directly and immediately affected by the pressing of a key. But a program running on the Apple has to include an explicit instruction to look at that byte before it can act upon anything entered at the keyboard. If the last bit of the keypress byte is a one, the program knows that someone has hit a key, and it can act accordingly. The program's action of inspecting the byte then automatically resets that bit to zero, so if it looks at that byte again, a thousandth of a second or so later, it won't mistakenly think the same key has been pressed again.

At any rate, the fact that only seven bits are used for ASCII values reduces the possible permutations by a factor of two, so now there are only 128 to contend with.

Here's a little Applesoft program you can enter that shows how the ASCII code works. Get Applesoft up and running (so that you see the right-hand bracket prompt on your screen) and type the following, exactly:

```
NEW
10 HOME
20 SPEED = 50
30 MAX = 127
40 FOR CODE = 0 TO MAX
50 PRINT CODE,
60 PRINT CHR$ (CODE)
70 PRINT
80 NEXT CODE
90 SPEED = 255
```

When you've got the program typed and you're satisfied that it matches our list exactly, type RUN and hit return. What you'll see is 128 numbers in one column (from zero to 127) and all the different characters that ASCII will encode in a second column.

Several things may be puzzling at this point. Why does nothing show up in column two between 0 and 32? Why does the Apple beep when column one gets to 7? What have the numbers in column one got to do with all this, anyway?

We'll deal with the last question first. We have quietly sidled up here to the subject of binary numbers. We will reserve a proper treatment of binary numbers for a subsequent installment of this column; for now we'll just point out that the 256 possible arrangements of eight bits can be used not only to encode any individual stroke at a computer keyboard, but also can represent all the numbers from 0 to 255. One particular pattern of zeros and ones can be assigned to be equivalent to 145, another to 17, and so on. In fact there is a very logical,

Riverbank Software Inc.

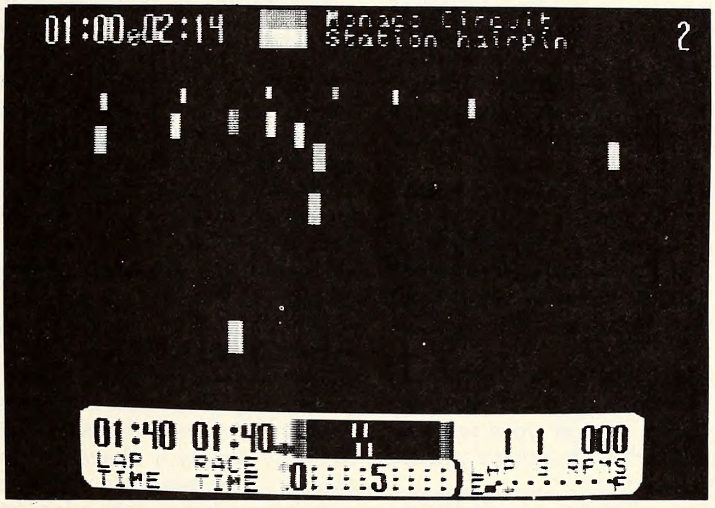
PROUDLY INTRODUCES

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.

orderly method of counting that will represent any positive whole number, using only the symbols 0 and 1. A number written in this fashion is called a binary number.

So now we have two codes. Any unique pattern of the eight bits of a byte represents, via ASCII, a character, and, via the binary number system, some whole number between 0 and 255.

A Ghostly Alphabet. Now, what about all that blankness in column two while column one reels off the numbers 0 through 32? Well, the ASCII equivalent for 32 is a space—what you get when you hit the spacebar—so that doesn't produce any visible output in this program. As for 0 through 31, those are called control characters, which brings us to the subject of the ctrl key on the left side of the Apple keyboard.

The control key—ctrl stands for control—enables the Apple to produce characters in addition to the standard letters, numbers, and punctuation. It works much like the shift key on a typewriter. To get a capital letter on a typewriter, you hold down the shift key and hit the appropriate letter. To get a control character on the Apple, simply press the appropriate character while holding down the control key. The ASCII code includes thirty-two control characters. They are the twenty-six letters of the alphabet (each in conjunction with the control key) plus six others, which we'll discuss presently.

So now the ASCII code structure can be conceptualized as having four thirty-two-character blocks. Positions 0 through 31 in the code are for control characters. Places 32 through 63 cover most of the punctuation symbols. The alphabetic characters (in upper case form) plus six additional symbols occupy positions 64 through 95; and from 96 through 127, something interesting happens.

If you have an unmodified Apple—that is, if you've done nothing since you got your Apple to modify the way the keyboard works—then positions 96 through 127 duplicate positions 32 through 63. However, if you have installed a device known as a lower-case adapter in your Apple, then at positions 96 through 127 you'll see the lower-case alphabet, plus six symbols.

Lower-case adapters are a fairly popular and relatively inexpensive add-on for Apple users (see the article on page 123). They're especially handy if you do a lot of word processing, although you do not have to have one to produce a printed document with upper and lower case letters.

If you don't have a lower-case adapter, but you do have a printer hooked up to your Apple at this moment, add the following lines to the program:

```
15 PR#1
95 PR#0
```

If you haven't done anything with your computer since the last time you ran the program, you can just enter those two lines; the rest of the program is still in memory, so you don't have to retype it. Turn on your printer now and run the program again with the two additional lines.

By doing this, you'll see that even if you don't have a lower-case adapter and the original version of the program you typed in will not display lower-case letters, a printer will recognize them and print them. The Apple manuals are not abundantly clear on this point. Page 139 of the *Applesoft Reference Manual*, for example, indicates that ASCII 96 through 127 will generate characters that repeat those from 32 to 63. In fact this is true only if the characters in question are sent to the television screen or monitor; if they're sent to another output device, like a printer, they appear as their normal, robust, lower-case selves.

Even if you do have a lower-case adapter, you've undoubtedly noticed that, unless you've run the software that came with your adapter or you're working within some kind of program that uses your Apple's lower-case capability, there's still no way you can type lower-case letters directly on the screen. If you've got the Applesoft prompt on your screen, for example, and you type the word PRINT, adapter or no adapter, whether you hold down the shift key or not, you're still going to get capital letters on your screen. The only way to get your screen to show a lower-case letter is by means of indirect statement, like PRINT CHR$(123), and that only works if you have the lower-case adapter.

This brings us to the general question of how anything gets from the keyboard to the screen.

Roadmap to the Screen. We've said that when you press a key on the Apple keyboard, a certain byte in the computer's memory stores certain information. The first seven bits display the binary equivalent of the ASCII number that corresponds to the key pressed, and the last bit becomes one—momentarily at least.

Up to this point all that's happened is that the contents of a certain byte have changed. For a character to appear on your monitor, some kind of program has to look at that byte and take action that will result in the character's being displayed.

You may be wondering how the devil you can ever type in a program and see the line numbers and instructions on the screen, since you have to have a program running in order to get anything displayed. The explanation is that Applesoft and Integer Basic are themselves programs. They're programs written in machine language and usually stored in read-only memory. When you see the right-hand bracket on your screen and a flashing cursor next to it, that means you are running a program called the Applesoft interpreter. One of the things this program will do is interrupt its customary activities to run a program of your devising, provided you encode the program in a way that meets certain syntactical requirements imposed by the Applesoft interpreter. In the meantime, until you start running your own program, the interpreter will do certain other things, including displaying that familiar bracket and cursor and printing characters on the screen as you type them.

It will, however, print no lower-case letters, even if you have an adapter. And this is the point: what appears on the screen is controlled by software, within the limits imposed by the hardware. It's perfectly possible to make a keypress of Q appear as X on the screen. It's not possible, however, to make it look like a frog, except by bypassing the Apple's character-generating hardware altogether and using its graphics capability instead. This you can do, and in fact a number of word processing programs available for the Apple do take this approach. When you press a key in these programs, the software actually draws a character in hi-res graphics. Programs of this kind can generate upper and lower case alphabets without requiring a lower-case adapter; they can also allow the user to design original type fonts.

Since output to the television screen or monitor is controlled by software, it's meaningless to talk about what key on the keyboard produces what symbol on the screen, except within the context of a particular program. However, since you are likely to spend a good deal of time interacting with the Basic interpreter, we will discuss the way the keyboard functions in that context.

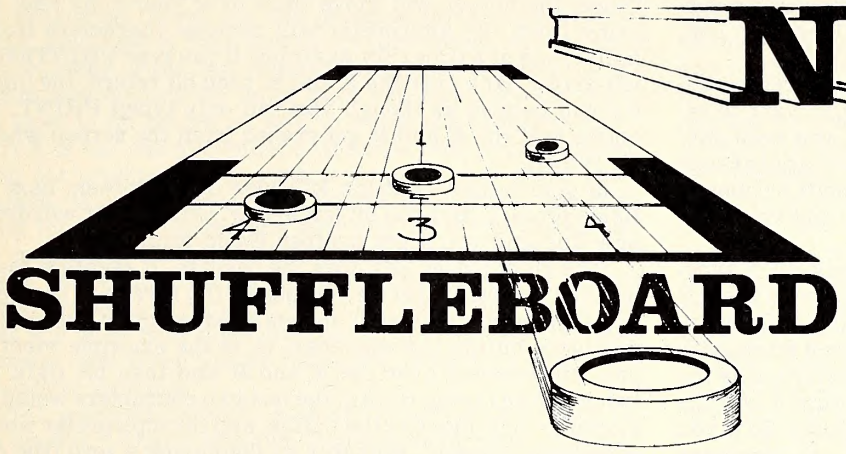
Dealing with Two-Faced Keys. You'll notice that some of the keytops on your keyboard have two characters marked on them and some have one. Those with two behave just like keys on a typewriter; when you hold down one of the shift keys in the lower corners of the keyboard and hit a two-faced key, you get the character on top; otherwise you get the one on the bottom.

If you do a little experimenting, you'll find there are two exceptions to what the previous paragraph said. If you hold down a shift key and hit G, you do not get the word BELL. Nor does anything ring inside your computer. You get a capital G. On the other hand, if you type a shifted M, you don't get any kind of M at all; you get that old familiar Applesoft prompt—the right-hand bracket.

You should feel free, by the way, to type anything you want at your keyboard. Your Apple may occasionally beep and give you plaintive commentary, but you can ignore it. It's pretty set in its ways and doesn't understand that you're only fooling.

Many of the characters that you can get only by holding down the shift key have special significance as part of program code in Applesoft or Integer Basic. We'll mention a cou-

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

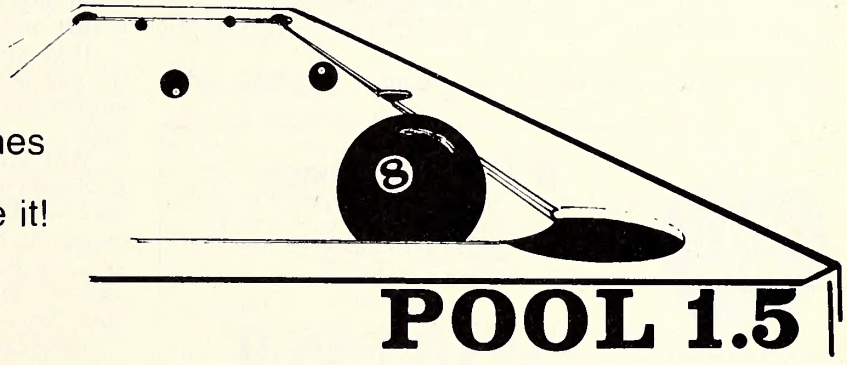


- *Real time HIRES Color Graphics*
- *Play Against your APPLE or another opponent*
- *An accurate and challenging game following in the tradition of Pool 1.5*

— *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

ple here. The asterisk, which is not your little footnote pointer sitting above a line of text, but rather a character of full weight and stature, is read by many programming languages as a symbol for multiplication. The computer needs some distinct symbol as a times sign, since it would ordinarily read X as a letter, not a mathematical operator. And the upward-pointing carat, achieved by hitting a shifted N, stands for exponentiation. The expression 3^2, in many programming languages, means three squared.

Keys That Change Key. Now we'll consider the various special keys on the Apple keyboard. Reset we'll leave for last, since it's a most exceptional character.

Rept is for repeat. Hold down any character and rept at the same time and that character will repeat. If you want to repeat a shifted character or a control character, you need only hold down the shift key or control key for the first appearance of your character. After that you can let go of shift or control and just hold down the character key and rept, and you'll get all the repetition you desire.

Return is much like the carriage return key on a typewriter. It terminates a line and restores the cursor to the left side of a new line. If you're in direct contact with the Basic interpreter—that is, if the current line on your screen starts with the Applesoft or Integer prompt—then hitting return tells the interpreter to regard everything between the prompt and the point where you hit return as a line of code in Basic. So if you type something like PRINT 4+21 and hit return, the computer will treat that as an instruction and will answer you accordingly. If you precede return with something that doesn't meet its exacting standards for Basic syntax, you'll get Basic's version of the Bronx cheer—a honk and an error message. (If you've started the line with a number, you can type any gobbledygook you like before hitting return, and you won't get an error message until you happen to hit RUN and return on a line by itself).

Moving along clockwise: the arrow keys do a little more than just move the cursor to the left or right. As you type in characters, besides displaying them on the screen, the Basic interpreter also loads them one by one into an area of memory called a buffer. This place is different from the byte that holds the current keypress. That location can only hold the most recently invoked character; the buffer will hold up to 255 characters. When you hit return, the interpreter empties the buffer and decides what to do about what you've said to it.

If you type a line of characters and then hit left arrow a few times, the cursor will move back over your copy and, at the same time, the interpreter will remove characters from the buffer, one at a time. For example, if you type PRINTER, then left-arrow back over the E and R, then hit return, the interpreter will behave as though you had only typed PRINT. You'll notice that the E and R get erased from the screen when you hit return.

If your cursor is at the left edge of the screen, next to the Basic prompt, and you hit left arrow, your cursor will drop to a new line and you'll get another Basic prompt.

The right arrow key does the reverse. It reenters material into the buffer. If there's nothing on the screen to reenter, hitting right arrow will enter spaces into the buffer, just as though you were hitting the spacebar. If, in the example mentioned, you left-arrowed over the E and R and then hit right arrow twice before typing return, the last two characters would be restored to their place in the buffer, and the interpreter would respond accordingly—probably by displaying a zero (the reason why you get a zero has to do with the Basic language, not with the keyboard or the functioning of the arrow keys).

Those Shiftless Keys. The shift keys, as you've seen, work like their counterparts on a typewriter. You have to hold one of them down while you type the character to be shifted. The big difference between shift on the Apple and shift on a typewriter is that only certain of the Apple's keys can be shifted.

If you type shift F, for example, you get the same character you would get by hitting F alone. In both cases—with or without shift—you would store the same value in the keypress byte.

With all the alphabetic keys on the Apple keyboard except P, N, and M, the keyboard hardware simply does not notice or care whether you are holding down the shift key or not. Because the hardware doesn't distinguish between, say, shift F and just F, you can't write software that will make that distinction, either. Most word processing programs use some other key—like esc or the right arrow—as a shifter.

Take Control of Your Apple. The control key works in many ways like the shift keys. You hold it down while pressing another key. Like the shift keys, control can affect only certain keys on the keyboard. One of the differences between control and shift is that, for the most part, control can affect only those keys that shift cannot. Shift-2, for example, gives you a double quote mark, but ctrl-2 gives you merely 2. The jurisdictions of shift and control overlap in only three places—P, N, and M.

The other big difference between shift and control is that most control characters don't produce any visible output on the screen (at least not while you're in the Basic interpreter). Type control-A, for example. You see nothing. Hit return, however, and you'll discover that the buffer has noticed your control-A, even though the screen did not.

The control characters are like ghosts—sensible by their effects, not their fleshly form. When you're naming a program or other file for the purpose of saving it on disk, you can embed control characters in the middle of an otherwise visible name, and no one but you and the disk catalog will know they're there. Thereafter, if you want to recall the file from disk, you'll still have to type the same control characters in the same places or the disk operating system will not recognize the name. (There are probably exceptions to everything said in this column; it is possible to produce software that will seek out and reveal control characters in file names, so it's not quite true that only you and the catalog will know.)

Certain control characters affect the positions of visible material on the screen. Ctrl-H and ctrl-U, for example, mimic the

left and right arrows, respectively, and hitting ctrl-M does the same thing as hitting return. Ctrl-J is unique in that it moves the cursor straight down; it produces what's called a line feed.

Ctrl-C and ctrl-S are handy characters to know about when you're running programs coded in Basic. Ctrl-S will halt the output of a program. If you just want to stop something that's going by on the screen too fast for you to read, you can hit ctrl-S; hitting any other key will then cause the program to resume. (If you have an older Apple, without the autostart ROM, ctrl-S will not function this way). Ctrl-C will terminate most Basic programs. If you hit this program while a program is running, you'll get a beep and the report "break in" followed by the number of the last line the program executed. (If the program was waiting for input from you when you hit ctrl-C, you'll have to hit return as well to make the program stop.)

Ctrl-X is the one exception to the rule that control characters do not produce direct visible output. Ctrl-X gives you a backslash. More importantly, it erases the current line of input and moves the cursor to a new line. So if you should suddenly look up at the screen and notice that you've had your hands in the wrong position on the keyboard and you've been typing gibberish, you can just erase the whole line with a ctrl-X. This will work both when you're talking directly to the Basic interpreter and when you're providing input to a Basic program you're running.

Ctrl-G provides familiar audible output—a toot from the Apple speaker. That's why the G keytop says BELL and why when you ran that program that appeared many long-winded paragraphs ago, you got a beep when column one printed the number 7.

Get Out of Jail Free. There's one more control character to consider, and that's the esc key. Esc—short for escape—doesn't look like a control character, since you don't have to hold down the control to get it—but its ASCII value is 27, so it's generally regarded as a control character.

Escape has several useful functions, when followed immediately by certain other specific keystrokes. Try hitting escape and then A. Do it a few times and you'll see your cursor scoot across the screen to the right. This is not the same thing as hitting right arrow; escape-A is what's called a pure cursor move. It moves the cursor but does not put characters into the buffer the way right arrow does.

Experiment with escape-B, escape-C, and escape-D. They're also pure cursor moves. Now try putting a lot of garbage on the screen. Just type anything at all; hit return occasionally and ignore the syntax error messages. Now get your cursor up into the middle of this clutter by hitting escape-D a few times. Now hit escape-E. The screen clears from the position of the cursor to the end of the current line. If you now hit escape-F, everything on the screen after the cursor will disappear. If you hit escape-shift-P (escape *at*-sign), the entire screen will go blank and your cursor will appear at the upper left corner. Try putting some more garbage on the screen and typing HOME; HOME is an Applesoft Basic command that does approximately the same thing as escape-shift-P, with one small difference, as you will note.

Escape-K, escape-J, escape-M, and escape-I are very handy characters. They do the same things as escape A-D, except that they allow you to move your cursor around without having to keep hitting the escape key. Once you hit escape with any of those four keys, you go into what's called the editing mode; you can keep hitting I, J, K, and M then, and you'll keep moving the cursor. To get out of editing mode, once you have the cursor where you want it, just hit any other key. Notice that the four keys in question form a diamond on the keyboard; that'll help you remember what key moves in what direction.

There are two control characters in the ASCII code—numbers 28 and 31—that are not available directly on the Apple keyboard. They're included in the ASCII code because other computing systems use them; and in fact you can send them to or receive them from other computers over modems by means of Basic statements like PRINT CHR$(28), but there's no way to produce them by a simple two-stroke combination at the keyboard. It's all right, though; you won't miss them.

!#@!!%#!?Reset! Now, as for reset . . . many people feel that Apple erred in the design of the keyboard: they put the reset key right next to return.

Reset is not really a key in the same sense as any of the others. It's more like a kind of off-on switch that reboots the system. It actually does a lot of things, and you can read about it on page 36 of the *Apple II Reference Manual.* The long and short of it, though, is that you want to avoid hitting that devil unless your system is hung up and you can't find any other way out.

Unfortunately, it's right up there next to return, ready and waiting for your outstretched little finger. Newer Apples have a feature that will prevent most accidental resets. If you remove the top of the computer and look inside the machine, directly in front of the crack between the 3 and 4 keys, you may see a little black lever protruding from the keyboard hardware. Slide this lever to the left, and the reset key won't work unless you're holding down control at the same time that you hit reset.

The Apple III, by the way, has its reset key in an altogether different place, off the main body of the keyboard, out of reach of unguided digits. ■

The Controller

from page 134

```
250 POKE ADDRESS,X
260 X = X * 2
270 IF PEEK (ADDRESS) > RESULT THEN COLUMN = N
280 NEXT N
290 REM
300 REM GET ROW FROM RESULT
310 REM
320 ROW = 0
330 IF RESULT = 251 THEN ROW = 1
340 IF RESULT = 253 THEN ROW = 2
350 IF RESULT = 254 THEN ROW = 3
360 KEY = (3 * ROW) + COLUMN
370 REM
380 REM NOW FIND THE KEY NAME
390 REM IN DATA STATEMENTS.
400 REM
410 RESTORE
420 FOR N = 1 TO KEY
430 READ A$
440 NEXT N
450 PRINT A$
460 DATA "1","2","3"
470 DATA "4","5","6"
480 DATA "7","8","9"
490 DATA "*","0","#"
500 REM
510 REM WAIT UNTIL THE KEY
520 REM IS RELEASED, THEN
530 REM START CHECKING AGAIN.
540 REM
550 POKE ADDRESS,0
560 IF PEEK (ADDRESS) < 255 THEN 560
570 GOTO 120
```

Listing 3. Keyboard input scanning program.

lines and eight row lines, thus allowing you to sense sixty-four keys. Note, too, that the keys need not be physically arranged in rows and columns; in fact, so long as the proper connections are made, they could be switches scattered about your house. Extensions of the technique in listing 3 will allow you to sense the position of any of the sixty-four switches individually.

Opto-Isolators. So far we have discussed only circuits powered by the Apple's power supply. *Never* connect a circuit powered by an outside supply directly to the APMOD's inputs. To sense high currents or voltages, you could use relays powered by the outside circuit and sense their contacts in the way I have already described. However, a cheaper part called the *opto-isolator* is now available. Opto-isolators (also called optical isolators or opto-couplers) are very useful when you must pass a signal from one circuit to another without electrically connecting the two circuits.

An opto-isolator is a relaylike device worked by a light beam instead of a magnetic field. Instead of a coil to generate magnetism, there is an LED to generate light. And instead of the contacts, there is a phototransistor—a transistor that conducts when the light from the LED shines on it. A very sensitive TIL-119 Darlington-type opto-isolator is available from Radio Shack as part number 276-133.

Diagram 7(a) shows how to connect the opto-isolator to a *DC* voltage. The resistor in the supply voltage line allows only one milliamp through the LED. This resistor should be a thousand times the voltage on the output circuit; a five-volt input should pass through a 5000-ohm resistor; and a twenty-four-volt circuit should have a 24K resistor. The value is not critical; anything from half to twice the value will pass sufficient current without destroying the opto-isolator. These values are for Radio Shack's opto-isolator; other devices may need more current and less resistance. But the pin layout of most opto-isolators is the same as in diagram 7.

Never feed negative voltage into the positive input of an opto-isolator. The LED in the opto-isolator cannot take more than about three volts backward. This means the circuit of diagram 7(a) cannot be used to sense *AC* circuits. For AC, a diode must be added to block reverse current, as in diagram 7(b). This circuit also adds a capacitor with sufficient capacity to keep the LED lit while the current is reversed.

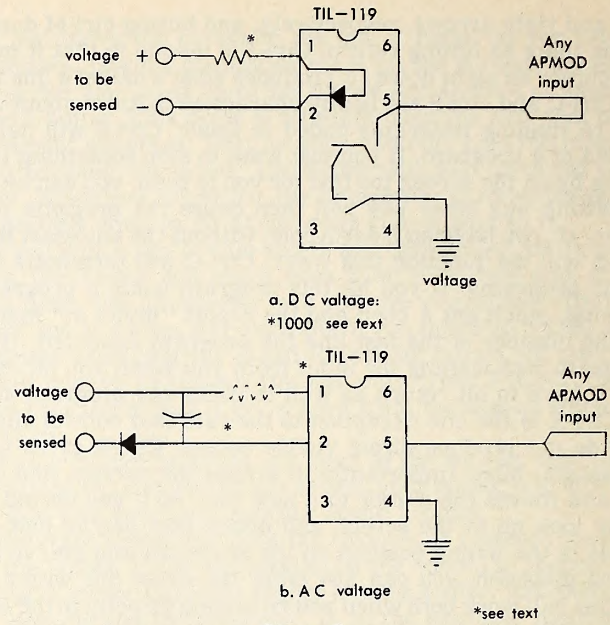


Diagram 7
Wiring Opto-Isolators to Inputs

The values of the resistor and capacitor depend on the frequency and voltage of the AC current. The formulas below should work for fifty to sixty Hz current with the opto-isolator shown. The exact values are not critical; from twice the calculated value to half the calculated value can be used. The formulas are: for C in microfarads, R in ohms, and V in volts:

$$R = 1000 \times V \quad C = 50 / V$$

For instance, for a 100-volt AC circuit, a resistance of 100,000 ohms and a capacitance of 0.5 microfarads can be used. Actually, the resistor can be anywhere from 50,000 to 200,000 ohms and the capacitor anywhere from 0.25 to 1 microfarad. Higher resistance and capacitance values will take longer to register changes in the state of the input circuit.

More Applications. A computer is by nature a general-purpose tool. Although designed for a specific purpose, the APMOD is also a general-purpose interface. Other ideas for the APMOD, not discussed in this article, are:

An intelligent logic test bench. With appropriate software, feed any desired combination of eight logic patterns, timed any way you like, from the APMOD to test equipment and show the results from up to eight test points as waveforms on the screen.

A turtle robot controlled by buffers and driven by the Apple's twelve-volt power supply. The APMOD's output lines would control the motors, and the input lines would sense bumper switches located around the perimeter of the robot.

A digital-to-analog converter, to be used as a sound generator. By connecting a resistor network to the APMOD's output pins, try to produce high-quality sounds by generating smoothly varying voltages which can then be amplified.

Where to Get The Parts. The APMOD card is available for $59.95 from Connecticut MicroComputer, 34 Delmar Drive, Brookfield, CT 06804. From Jameco Electronics, 1355 Shoreway Road, Belmont, CA 94002, you can get: 7416 TTL buffer circuit, $1.19; 7417 TTL buffer circuit, $1.19; 1N914 diodes, package of ten, $.99. From GC Electronics, 400 South Wyman Street, Rockford, IL 61101, twelve-key row column keyset, part number E2-146, $10.65. The following parts are available at most Radio Shack stores: Forty-pin edge connector, part number 276-1558, $5.95; five-foot twenty-wire cable, part number 2780770, $3.95; five-volt DIP relay, part number 275-215, $4.49; opto-isolator, part number 276-133, $1.99. ■

Race for Midnight

by STEVEN SACK
75 Rooms
FULL HI-RES
COLOR AND SOUND
requires Applesoft*
ROM, 48K, disk,
DOS 3.2 or 3.3

$29.95

Ask your dealer or order direct from:

Avant-Garde Creations
PO Box 30160
Eugene, OR 97403
(503) 345-3043

VISA MasterCard
Visa/Mastercard Accepted

*Apple is a trademark of Apple Computer, Inc.

You live in a small town in the 14th century. You were awakened this morning by a terrible pain in your arm. Upon examining your arm, you find a bloody gash in it. Wisely you cover it so that nobody will see it. Later, you find that the townspeople had seen a werewolf last night and one person had shot an arrow at it, but evidently he had missed, because the werewolf continued running. You instantly deduce that you must have been the werewolf and realize that you must find an antidote.

You decide to go to a nearby dungeon that is deserted. Legend says that a powerful wizard, Evro, once lived there, but he became a victim of his own experiments. The rumor is that he had strange and deadly creatures under his power. You decide that you might be able to find some sort of recipe for a potion to cure your affliction.

Unfortunately, it is right around the time of the full moon, and you know that you will almost certainly become a werewolf tonight again. In fact, the moon rises tonight at exactly midnight, which has always had a mystical aura about it. You arm yourself with a fine sword and a full lantern and set off in search of the dungeon. After finding the ruins which used to be Evro's castle you look for some sort of entrance, such as a stairway.

While searching, the ground suddenly gives way under you. Breathing heavily, and trembling with fear. . .

PRESS (RETURN) TO CONTINUE

DISCOVER THE UNIVERSE

PROUDLY ANNOUNCING

SPACE ADVENTURE

A REAL-TIME SPACE FLIGHT SIMULATOR

Welcome aboard the Callisto.

The Callisto offers you an excellent view of the vast universe before you.

You take the helm and enter the coordinates: quadrant 56, 58, 53; sector 14, 62, 12. Instantly the autopilot takes control of the ship. The viewport reappears as your course automatically changes and you begin your hyperwarp journey. Upon arrival, the message indicator flashes. You take the message and ask the computer for a tactical plot of the situation.

SPACE ADVENTURE is this and much more!

SPACE ADVENTURE offers you real-time instruments, real-time graphics, and an onboard computer that will amaze you.

SPACE ADVENTURE gives you the chance to pilot a space craft through true 3-D space for an adventure you will never forget. **SPACE ADVENTURE** . . . for any Apple II or II+ with 48K and disk. Suggested retail price is $29.95.

Sierra Software

EPISODE TWO COMING SOON
DEALER INQUIRIES INVITED

SIERRA SOFTWARE • 536 E. SAHARA AVENUE • LAS VEGAS, NEVADA 89104

Softalk Presents The Bestsellers

Softalk's bestseller list takes on both a new look and an old look on its first anniversary.

The new look, of course, is the expanded breakout of various categories to show Apple owners with special interest what is hot in their areas.

The old look has to do with the leader of the pack. *VisiCalc* was the first program to head the Top Thirty list and for months it appeared that Personal Software's electronic calculator had taken permanent possession of that lofty position.

But from Christmas season of 1980, hot new entertainment product has consistently pushed *VisiCalc* from first to a contending position. But this anniversary, *VisiCalc* returned to the top with a vengeance, swamping its gaming competitors.

It's more than a little ironic that the program that first nudged *VisiCalc* from the top spot, *Apple Galaxian*—now known as *Alien Rain*—slipped from the Top Thirty this month for the first time since it bested *VisiCalc*.

Raster Blaster, after reigning in the top spot for three months, dropped all the way to fourth. While last month's runner-up, *Gorgon*, maintained its position, *Cranston Manor* proved its heritage as an On-Line offspring by joining *VisiCalc* and *Gorgon* ahead of the fallen monarch.

Personal Software's *CCA Data Management System* regained a spot in the Business 10 after an absence of two months, even as Personal announced that it would be discontinuing the program in favor of new data base management system.

As to the new breakouts, they still cause some problems in categorizing software. Some of the divisions were fairly clear, but the software doesn't necessarily want to fit the niches defined.

Strategy 5

1. *Robot War*, Silas Warner, Muse
2. *Flight Simulator*, Bruce Artwick, SubLogic
3. *Computer Baseball*, Charles Merrow and Jack T. Avery, Strategic Simulations
4. *Sargon II*, Dan and Kathe Spracklen, Hayden
5. *Warp Factor*, Paul Murray, Strategic Simulations

Adventure 5

1. *Hi-Res Adventure #3: Cranston Manor*, Harold DeWitz and Ken Williams, On-Line Systems
2. *Hi-Res Adventure #2: The Wizard and the Princess*, Roberta and Ken Williams, On-Line Systems
3. *Oo Topos: An Extraterrestrial Adventure*, Michael Berlyn, Sentient Software
4. *Zork*, Mark S. Blank, Timothy Anderson, Bruce Daniels, P.D. Leblins, Scott Cutler, and Joel Berez/Infocom, Personal Software
5. *Softporn*, Chuck Benton/Blue Sky Software, On-Line Systems

Fantasy 5

1. *Ultima*, Lord British, California Pacific
2. *Kaves of Karkhan*, Rodney Nelsen, Level-10
3. *Dragon Fire*, Rodney Nelsen, Level-10
4. *Dragon's Eye*, Robert Leyland, Automated Simulations
5. *Crush, Crumble and Chomp*, Automated Simulations

Business 10

1. *VisiCalc*, Software Arts/Dan Bricklin and Robert Frankston, Personal Software
2. *DB Master*, Alpine Software/Stanley Crane and Jerry Macon; and Barney Stone, Stoneware
3. *VisiTrend/VisiPlot*, Micro Finance Systems/Mitch Kapor, Personal Software
4. *VisiDex*, Peter Jennings, Personal Software
5. *Personal Filing System*, John Page, Software Publishing Corporation
6. *PFS: Report*, John Page, Software Publishing Corporation
7. *Apple Plot*, Apple Computer
8. *CCA Data Management System*, Creative Computer Applications, Personal Software
9. *Data Reporter*, Synergistic Software
10. *Payroll*, BPI, Apple Computer

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

As an example, it seemed natural to break out the Home/Hobby 10 into a Home section and a Hobby section, with the hobbyist being considered a person who does programming. But do Sensible Software's utilities fall into that category or into the general home use category? Likewise, are *Graphtrix* and *Hand Holding Basic* the tools of hard-core hobbyists or are they more home-oriented?

As with *Softalk*'s original breakouts, the divisions were made arbitrarily and are susceptible to knowledgeable second-guessing by one and all.

Breaking the word processors out from Business packages recognizes the broader market served by the text handlers. Many home users buy *Apple Writer*, but few buy a $500 general ledger accounting package.

The three new game categories recognize subsets of enter-

Home 10

1. *Typing Tutor*, Image Producers, Microsoft
2. *Graphtrix*, Steven Boker, Data Transforms
3. *Home Money Minder*, Bob Schoenburg and Steve Pollack, Continental Software
4. *Personal Finance Manager*, Apple Computer
5. *VisiTerm*, Tom Keith, Personal Software
   ASCII Express, Bill Blue, Southwestern Data Systems
7. *Hand Holding Basic*, Apple Computer
8. *Dow Jones Series Portfolio Evaluator*, Apple Computer
9. *Dow Jones News & Quotes Reporter*, Apple Computer
10. *Financial Management System II*, D. R. Jarvis, D. R. Jarvis Computing

Hobby 10

1. *DOS 3.3*, Apple Computer
2. *The Complete Graphics System*, Mark Pelczarski, Penguin Software
3. *Expediter II*, Stewart Einstein and Dennis Goodrow, On-Line Systems
4. *DOS Tool Kit*, Apple Computer
5. *Super Disk Copy III*, Charles Hartley, Sensible Software
6. *Multi-Disk Catalog*, Roger Tuttleman, Sensible Software
7. *Bill Budge's 3-D Graphics Package*, Bill Budge, California Pacific
8. *LISA Assembler*, Randy Hyde, On-Line Systems
9. *Higher Text II*, Synergistic Software
10. *Applesoft Compiler*, Jonathan Eiten, Hayden

Word Processors 5

1. *Apple Writer*, Apple Computer
2. *Word Star*, Micro Pro
3. *Super Text*, Ed Zaron, Muse Software
4. *Magic Window*, Gary Shannon and Bill Depew, Artsci
5. *Easy Writer*, John Draper, Information Unlimited Software

tainment software, members of which all too often hover just under the Top Thirty mark and seldom receive their due recognition.

Even here, arbitrary definitions were needed. Arcade games are not broken out separately because so many enjoy high places on the Top Thirty already. *Arcade games* are defined as games where hand-eye coordination and dextrous manipulation of the Apple keyboard or game controllers are the keys to success.

By this definition, *Pool 1.5* and *Olympic Decathlon* got thrown in with the arcade games.

Strategy games are defined as adversary games—person against person or person against computer—where a successful result would more likely occur from mental prowess than from physical dexterity.

Adventure games are considered those in which success requires solving several riddles or puzzles as you work your way through the program.

Fantasy games are those in which you create one or more characters with whom you identify as the game progresses.

Four new programs made the Top Thirty in August. They were *Sneakers* and *Epoch* from Sirius Software, *Star Thief* from Cavalier Software, and *Graphtrix* from Data Transforms. Three packages rejoined the Top Thirty after a month off. They were *Apple Writer, Olympic Decathlon,* and *Sargon II.*

Sneakers, which leapt from nowhere to number 9 on the list without being available the entire month of August, was expected to become a one-of-a-kind collectors item even before it hit the market. Its author, a medical student, designed *Sneakers* for fun and had no intention of ever writing another commercial program. But an Apple a day seems indeed capable of keeping the doctor away; and another Turmell program is on the way.

Business in August was slightly up, portending an increase from now through the Christmas season. Sales of individual titles were impacted, however, by the stiffer competition now evident in the Apple market. ◨

Apple-franchised retail stores representing approximately 4.9 percent of all sales of Apples and Apple-related products volunteered to participate in the poll.

Respondents wee contacted early in September to ascertain their sales leaders for the month of August.

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 September represented every geographical area of the continental United States as well as Hawaii.

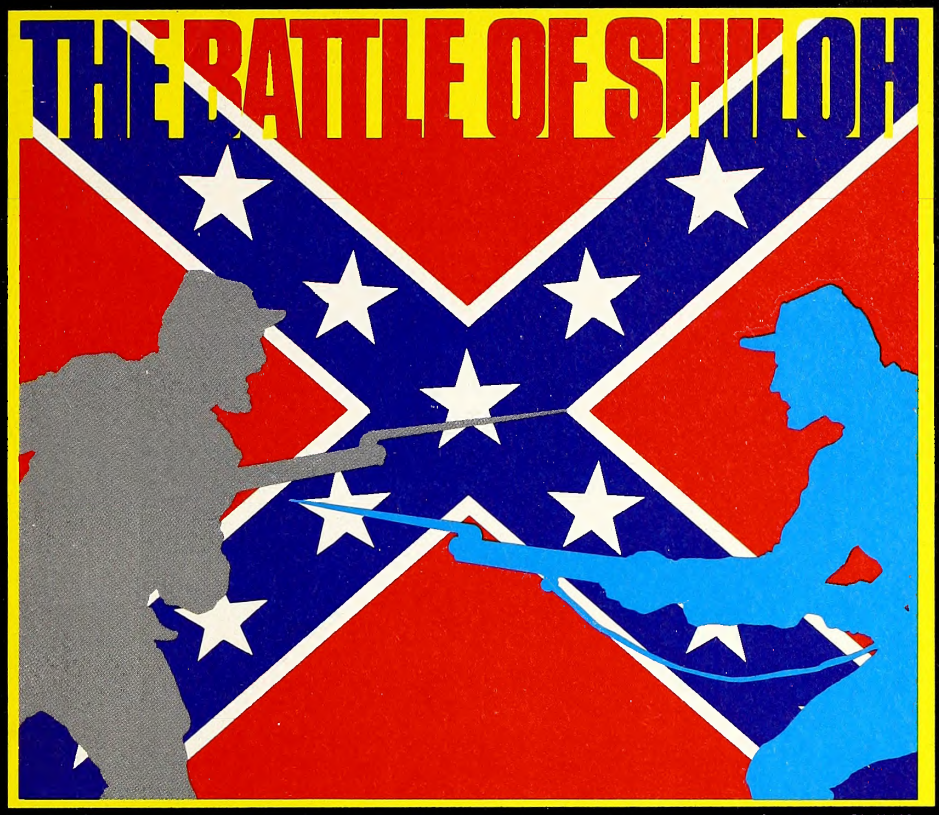
Results of the responses were tabulated using a formula that resulted in the index number to the left of the program name in the Top Thirty listing. The index number is an arbitrary measure of relative strength of the programs listed. Index numbers are correlative only for the month in which they are 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 8.1 percent, which translates roughly into the theoretical possibility of a change of five points, plus or minus, in any index number.

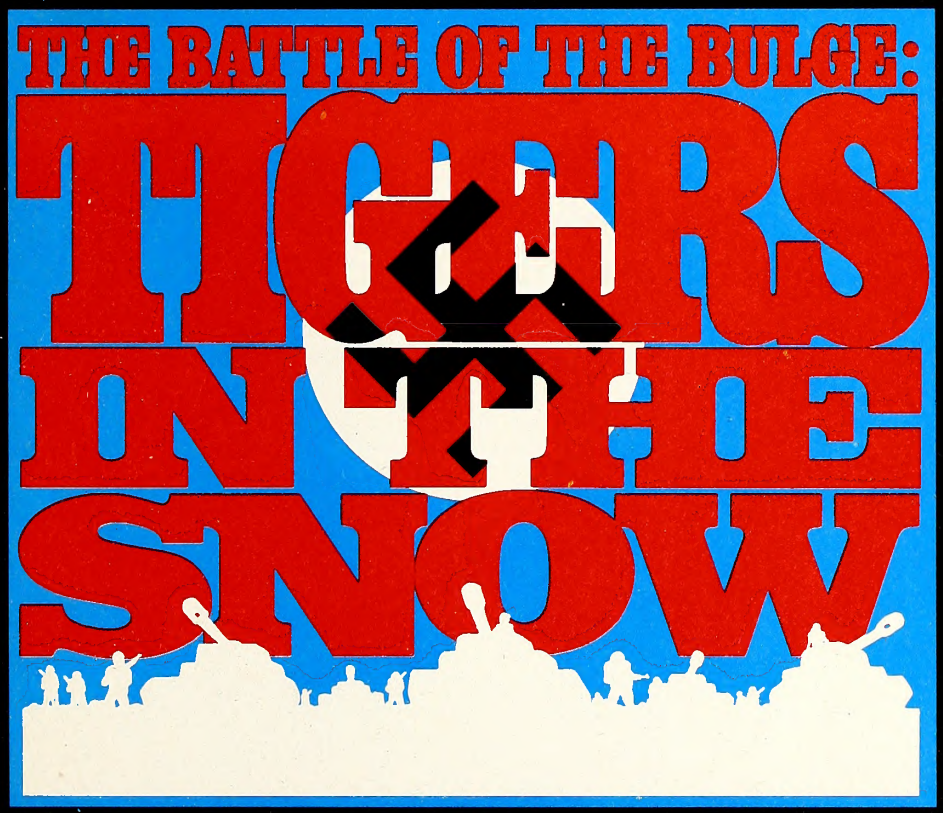
The Top Thirty

This Month	Last Month	Index	
1.	3.	92.96	*VisiCalc,* Software Arts/Dan Bricklin and Robert Frankston, Personal Software
2.	2.	70.08	*Gorgon,* Nasir, Sirius Software
3.	9.	56.94	*Hi-Res Adventure #3: Cranston Manor,* Harold DeWitz and Ken Williams, On-Line Systems
4.	1.	56.46	*Raster Blaster,* Bill Budge, BudgeCo.
5.	7.	55.00	*Apple Panic,* Ben Serki, Broderbund Software
6.	11.	40.88	*DB Master,* Alpine Software/Stanley Crane and Jerry Macon; and Barney Stone, Stoneware
7.	9.	36.02	*Snoggle,* Jun Wada, Broderbund Software
8.	8.	33.10	*Ultima,* Lord British, California Pacific
9.	—	32.61	*Sneakers,* Mark Turmell, Sirius Software
10.	6.	30.66	*Robot War,* Silas Warner, Muse Software
11.	4.	30.18	*Pool 1.5,* Don Hoffman, Howard de St. Germaine, and Dave Morock, Innovative Design Software
12.	12.	26.28	*Space Eggs,* Nasir, Sirius Software
13.	14.	25.80	*VisiTrend/VisiPlot,* Micro Finance Systems/Mitch Kapor, Personal Software
14.	16.	23.85	*Flight Simulator,* Bruce Artwick, SubLogic
15.	17.	23.36	*Gobbler,* Olaf Lubeck, On-Line Systems
16.	—	22.87	*Apple Writer,* Apple Computer
17.	17.	22.39	*VisiDex,* Peter Jennings, Personal Software
18.	12.	19.47	*DOS 3.3,* Apple Computer
19.	24.	18.98	*Personal Filing System,* John Page, Software Publishing Corporation
20.	—	17.03	*Star Thief,* Jim Nitchals, Cavalier Computing
21.	—	16.55	*Olympic Decathlon,* Tim Smith, Microsoft
22.	15.	16.06	*Hi-Res Adventure #2: The Wizard and the Princess,* Roberta and Ken Williams, On-Line Systems
23.	26.	13.63	*Typing Tutor,* Image Producers, Microsoft
24.	—	13.14	*Graphtrix,* Steve Boker, Data Transforms
	22.	13.14	*Computer Baseball,* Charles Merrow and Jack T. Avery, Strategic Simulations
	27.	13.14	*The Complete Graphics System,* Mark Pelczarski, Penguin Software
27.	—	12.65	*Sargon II,* Dan and Kathe Spracklen, Hayden
	26.	12.65	*Space Warrior,* Marc Goodman, Broderbund Software
29.	—	11.68	*Epoch,* Larry Miller, Sirius Software
	5.	11.68	*Expediter II,* Stewart Einstein and Dennis Goodrow, On-Line Systems

TWO CLASSIC BATTLES WITH A FRESH, NEW, COLORFUL APPROACH — THE SSI APPROACH.



THE BATTLE OF SHILOH: A brigade-level simulation of the first grand battle of the Civil War, pitting the Confederate Army against Grant's troops and Union gunboats.



TIGERS IN THE SNOW: Ghostlike Nazi Tiger tanks and infantry sweep across the dark, frozen forests of the Ardennes against a surprised U.S. force in this division/regiment-level simulation of Hitler's last desperate attack.

When we set out to design games based on the Battle of Shiloh and the Battle of the Bulge, we knew we had to give these classic, favorite themes a fresh, new look and feel. After all, we've established a reputation in strategy gaming for unsurpassed innovation, excitement, and sophistication.

First, we put the games on the Apple® II.* programming it perfectly to eliminate all organizational and administrative headaches so you can give your undivided attention to strategy planning.

Then we gave both games the realism and playability you're looking for. An elegant yet easy-to-use movement system for unimpeded play. Historical detail and accuracy for richness and color. Hi-Res graphics that add even more color. Great solitaire scenarios against the computer. Zones of control. A step-reduction combat system. And more.

Finally, we threw in features you wouldn't expect. For example, TIGERS IN THE SNOW has artillery and airpower allocations along with fuel and supply limitations. THE BATTLE OF SHILOH allows you to fine-tune combat strengths for each side, providing for the ultimate in play balance. It even lets you select risk levels and ferocity of attack (or defense).

For $39.95 each, these are extraordinary games at quite an ordinary price. So head on down to your local store and check them out today!

VISA and M/C holders can order by calling 800-227-1617, ext. 335 (toll free). In California, call 800-772-3545, ext. 335.

To order by mail, send your check to: Strategic Simulations Inc, 465 Fairchild Dr., Suite 108, Mountain View, CA 94043.

All our games carry a 14-day money-back guarantee.

*48K disc for Apple® II with ROM Card or Apple® II Plus.

SSI

STRATEGIC SIMULATIONS INC

As part of our demanding standards of excellence, we use maxell floppy discs.

Apple is a registered trademark of Apple Computer Inc.

Ulysses

and the Golden Fleece

HI-RES ADVENTURE #4 by Bob Davis & Ken Williams

ON-LINE systems
209-683-6858 • Visa, Master Charge, C.O.D. or Checks Accepted.