The Commodore 128/MS-DOS Connection

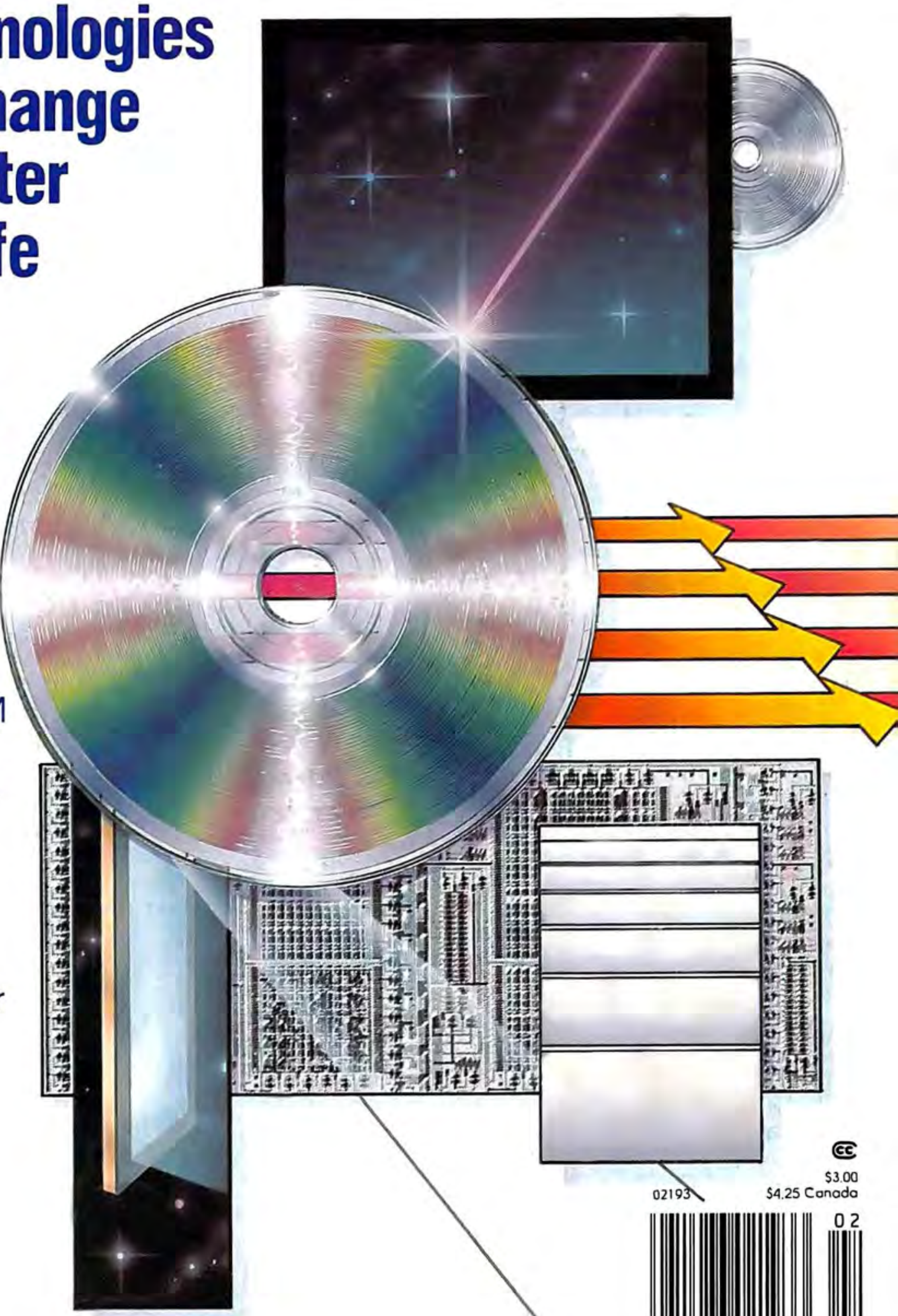
February 1988

COMPUTE!

The Leading Magazine Of Home, Educational, And Recreational Computing

6 New Technologies That Will Change Your Computer And Your Life

Laser Discs, Super Chips, And More



Tiles
A challenging, one- or two-player memory game for Apple II, IBM PC and compatibles, Commodore 64, Atari, Amiga, and Atari ST.

Rapid Reflex For IBM PC
A fast-paced game for the whole family—up to six players of any age

Two 64 Emulators For Amiga:
A First Look

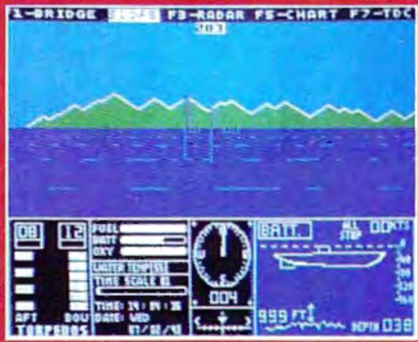
$3.00
$4.25 Canada

02193

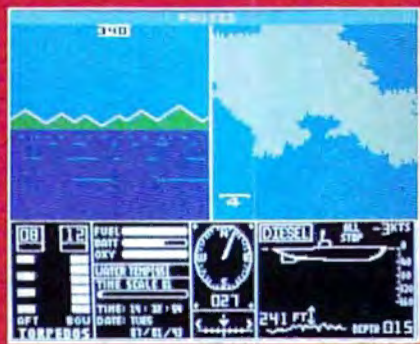
02

0 71486 02193 3

ON RIGHT OUT OF THE WATER!

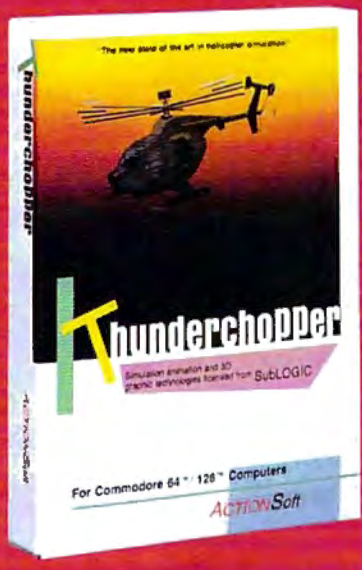


C64 Screens shown. Other computer versions may vary.



ThunderChopper

And don't forget Thunder-Chopper! For true helicopter action and realism, nothing else even comes close. Thunder-Chopper incorporates the most advanced graphics, flight systems, and game-playing factors to provide a sensational balance of strategy and fun.



See Your Dealer

Or write or call for more information. Up Periscope! and Thunder-Chopper are available on disk for the Commodore 64/128, Apple II, and IBM PC line of personal computers for the suggested retail price of $29.95. For direct orders please specify which computer version you want. Include $2.00 for shipping (outside U.S. $6.25) and specify UPS or first class mail delivery. Visa, MasterCard, American Express, and Diners Club charges accepted.

ACTIONSoft

"GENERATIONS AHEAD IN STRATEGY/ACTION SOFTWARE"

201 WEST SPRINGFIELD AVENUE, SUITE 711
CHAMPAIGN, IL 61820 (217) 398-8388

FEEL THE HEAT

No matter what kind of game you're in the mood for, you'll find that if it's in a box marked FIREBIRD, it's really hot!

Firebird offers the excitement of a whole range of challenges — Adventures and Strategies for long rainy weekends, Arcade games for exhilarating evenings, Simulations when you need to get away from it all, Productivity software when you're feeling — well, productive.

Look for the Firebird — it stands for top-quality software in virtually *every* category.

Adventures Ablaze

The world has been waiting for a sequel to the highly-acclaimed PAWN, with its stunning graphics and revolutionary text-handling system. Here at last is GUILD OF THIEVES, an extraordinary Adventure like no other! Look for the Firebird logo on other addictive Adventures, too — like the stunning GOLDEN PATH, or the magical illusion of KNIGHT ORC with its superb graphics.

Four-Alarm Arcade Action

When it comes to fast-paced Arcade games, nothing beats Firebird's STARGLIDER, a breathtaking combat flight simulation with incredibly rapid 3D vector effects.

Or try ELITE, a game so different it defies

categorization, or the endlessly puzzling CHOLO.

Sizzling Strategies

If you're in the mood for Strategy, and have exceptional concentration and near-perfect hand/eye coordination, you'll love the challenge of SENTRY with its more than 10,000 full-color landscapes and four-way scrolling. More Strategies from Firebird: the revolutionary UNIVERSAL MILITARY SIMULATOR* and TRACKER, games that introduce totally new concepts in tactical warfare.

Power for Productivity

Firebird even offers economical, professional-quality Productivity software! ADVANCED ART STUDIO is a comprehensive art package with powerful graphics routines, while our MUSIC SYSTEM in both concise and advanced versions, brings new stimulation to music lovers. This is Firebird — the best in interactive software of every description. We'll prove to you that you don't have to keep switching brands to satisfy your obsession for challenge!

CREDIT CARD HOTLINE
(201) 444-5700

MasterCard VISA

FIREBIRD

"The First Full Line In Software"

Firebird Licensees, Inc.
P.O. Box 49, Ramsey, NJ 07446
(201) 444-5700

Firebird software for IBM-compatible systems, Amiga, C64, Atari 520ST, MacIntosh, and Apple II, is available from $24.95 to $59.95.

*Available January '88.

COMPUTE!

FEBRUARY 1988
VOLUME 10
NUMBER 2
ISSUE 93

The Leading Magazine of Home, Educational, and Recreational Computing

FEATURES

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 98 before typing in programs.

AP Apple, **GS** Apple IIGS, **Mac** Macintosh, **AT** Atari, **ST** Atari ST, **AM** Amiga, **64** Commodore 64, **128** Commodore 128, **PC** IBM PC, **PCjr** IBM PCjr, • General interest

COMPUTE! Publications, Inc. abc
One of the ABC Consumer Magazines, Inc.
A Capital Cities/ABC, Inc. Company
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019

COMPUTE! The Leading Magazine of Home, Educational, and Recreational Computing (USPS: 537250) is published monthly by COMPUTE! Publications, Inc., 825 7th Ave., New York, NY 10019 USA. Phone: (212) 265-8360. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. POSTMASTER: Send address changes to: **COMPUTE!** Magazine, P.O. Box 10955, Des Moines, IA 50950. Second class postage paid at New York, NY and additional mailing offices. ISSN 0194-357X.

Editor's Notes

Arthur C. Clarke once paraphrased the great scientist J. B. S. Haldane, pointing out that the future will not only be stranger than we imagine, it will also be stranger than we *can* imagine.

Fueling that strangeness will be the emergence of new and unexpected technologies. The story of this century—and particularly of the last three decades—is one of technology and the ways in which it changes our lives.

And the rate of technological change continues to accelerate. This speedup is nowhere more obvious than in microcomputers.

Clarke's hypothesis has often been confirmed, both in and outside our industry. Certainly any number of futurists foresaw a resounding impact of computers on society. Few, however, foresaw the delivery of mainframe power to every desktop for a few thousand dollars. Yet that is exactly what is happening at the moment. Over the next few years, the power and importance of the micro will continue to increase, augmented by the emergence of additional technologies, some of them not yet imagined.

Others are closer to becoming reality. This month, we look at six new technologies that are either based on or directly impinge upon the microcomputer. The six technologies we chose as topics for our article—superconductors, hypertext, flat screens, parallel/neural processing, microminiaturization, and optical storage—stand at various stages of readiness for making an impact upon our lives. There are hypertext processors and flat screens already on the market, for example. On the other hand, superconductors—long-time laboratory curiosities—have moved closer to the laboratory door lately, but they remain the focus more of experimentation and development than marketing.

Marketing will come, though, once the technology proves itself. Among the many lessons learned over the past decade is an important one about the business and consumer markets' hunger for increased power and capability. Where a decade ago microcomputers were a novelty, they are now ubiquitous. Each new generation of computers is embraced with enthusiasm, as developers and users alike continue to find new applications for new technology. We are confident that, as the technologies discussed in our article mature, they will be put to work.

Nor are these the only exciting technologies on the horizon. Far from it, in fact. In the past few weeks alone, we have seen news stories announcing breakthroughs in magnetic storage capacity, telecommunications, and computational speed.

And then there are those still-new advances that are already exerting a large impact on business, educational, and home computing. Intel's superfast 80386 chip, virtually unheard of a couple of years ago, is now the driving force powering the next generation of MS-DOS machines. It was only four years ago that Apple introduced the 128K Macintosh and a year earlier that Commodore introduced the 64. Today, the Macintosh II and the multitasking Amiga are making headlines. In the same period of time, Atari has moved from being known primarily as the manufacturer of a leading videogame system to marketing its ST and Mega ST.

One result of all the advances is the looming end to machine-specific concerns. When you have a mainframe on your desktop, you can do just about anything with it that you want. Already, interconnectivity is a major industry issue. Some software publishers are insisting that data files generated with their software be usable on any system running that software. It is increasingly possible for Apple, IBM, Commodore, and Atari users to communicate with one another. The *next* generation of computers—bringing more new technology—will doubtless present us with solutions to the remaining connectivity questions. And beyond that lies the next generation.

Clarke and others have made the point that any sufficiently advanced technology is indistinguishable from magic. There's nothing magical about computers; they rest upon a body of technology and hard work accumulated over the course of the century. Still, judging by the rate at which computer technology is advancing, we're in for a magical ride over the next few years, with the microcomputer serving as our flying carpet.

Keith Ferrell

Keith Ferrell
Features Editor

Publisher/Editorial Director — William Tynan
Managing Editor — Kathleen Martinek
Associate Publisher — Selby Bateman

Editor, COMPUTE! & COMPUTE!'s GAZETTE — Lance Elko
Assistant Editors, COMPUTE! — Rhett Anderson, Randy Thompson, Clifton Karnes
Production Director — Tony Roberts
Editor, COMPUTE!'s Atari ST Disk & Magazine and COMPUTE!'s PC Magazine — Tom R. Halfhill
Editor, COMPUTE!'s Apple Applications Magazine — Gregg Keizer
Features Editor — Keith Ferrell
Assistant Technical Editors — Dale McBane, Jim Fuchs
Assistant Editor, COMPUTE!'s Atari ST Disk & Magazine — Todd Heimarck
Assistant Editor — John Shadle
Assistant Features Editor — Tom Netsel
Programming Supervisor — Patrick Parrish
Editorial Programmers — Tim Midkiff, William Chin, George Miller
Buyer's Guide Coordinator — Caroline Hanlon
Copy Editors — Karen Uhlendorf, Karen Siepak, Jill Champion, Lori Sonoski
Editorial Assistant — Mickey McLean
Submissions Reviewer — David Hensley
Programming Assistants — Troy Tucker, Joyce Sides
Executive Assistant — Debi Nash
Administrative Assistants — Julia Fleming, Iris Brooks, Sybil Agee, Claudia Earhart
Receptionist — Anita Armfield
Associate Editors — Jim Butterfield, Toronto, Canada; Fred D'Ignazio, East Lansing, MI; David Thornburg, Los Altos, CA
Contributing Editor — Bill Wilkinson

COMPUTE!'s Book Division
Editor — Stephen Levy
Assistant Editors — Tammie Taylor, Robert Bixby, Lynne Weatherman
Programming Assistant — David Florance

Production Manager — Irma Swain
Art Director — Janice R. Fary
Assistant Art Director — Lee Noel, Jr.
Assistant Production Manager — De Potter
Designer — Tony Jacobson
Artists — Robin Case, Kim Potts, Scotty Billings
Typesetting — Terry Cash, Carole Dunton
Illustrator — Harry Blair

Group Advertising Director/Consumer Electronics — Peter Johnsmeyer
Advertising Director — Bernard J. Theobald, Jr.
Production Coordinator — Kathleen H. Ingram

Customer Service Manager — Diane Longo
Dealer Sales Supervisor — Jose Cruz
Individual Order Supervisor — Cassandra Green

William Tynan, Vice President
Richard J. Marino, Vice President, Advertising Sales
Ilene Berson Weiner, Vice President, Production

Editorial Board
William Tynan, Kathleen Martinek, Selby Bateman, Lance Elko, Tom R. Halfhill, Stephen Levy

COMPUTE! Publications, Inc. publishes
COMPUTE!
COMPUTE!'s Gazette
COMPUTE!'s Gazette Disk
COMPUTE!'s Apple Applications
COMPUTE!'s Atari ST Disk & Magazine
COMPUTE! Books

Editorial offices: 324 West Wendover Avenue, Suite 200, Greensboro, NC 27408 USA
Corporate offices: 825 7th Avenue, New York, NY 10019, 212-265-8360
Individual Orders: (Single Copy) 800-346-6767 (In NY 212-887-8525) 10:00 AM–12:30 PM, 1:30 PM–3:00 PM Monday–Friday
Dealer Sales: 800-638-3822 (In NY 212-887-8566) 9 AM–5 PM Monday–Friday

Coming In Future Issues

Casino Blackjack: A high-stakes card game for Commodore 64, IBM PC/PCjr, Amiga, Atari ST, Apple II, and Atari

Color Pad: A comprehensive multicolor drawing program for the 64

Math Graphics for the PC

Cursor Plus: Enhanced editing for Atari

Catalog To Text File for Apple

Amiga Math Draw

Subscription Orders & Inquiries

COMPUTE!
P.O. Box 10954
Des Moines, IA 50340

TOLL FREE
Subscription Order Line
1-800-727-6937

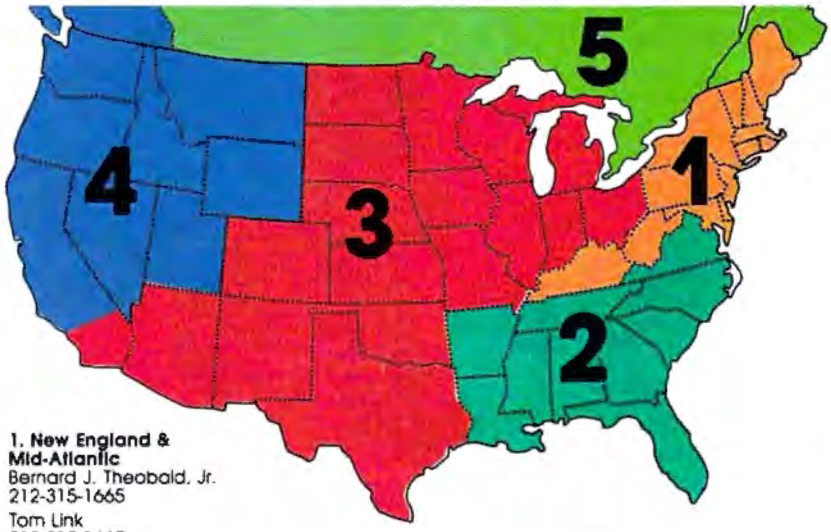
Foreign Subscription
515-247-7631

COMPUTE! Subscription Rates (12-Issue Year):

US		Canada & Foreign	
(one yr.)	$24	Surface Mail	$30
(two yrs.)	$45	Foreign Air	
(three yrs.)	$65	Delivery	$65

Member Audit Bureau of Circulations
Magazine Publishers Association

Advertising Sales



1. New England & Mid-Atlantic
Bernard J. Theobald, Jr.
212-315-1665
Tom Link
212-315-1665

2. Southeast & Foreign
Harry Blair
919-275-9809

3. Midwest & Southwest
Jerry Thompson
312-726-6047 (Chicago)
713-731-2605 (Texas)
303-595-9299 (Colorado)
415-348-8222 (California)
Lucille Dennis
415-348-8222

4. West, Northwest & British Columbia
Jerry Thompson
415-348-8222
Lucille Dennis
415-348-8222

5. Canada
Harry Blair
919-275-9809

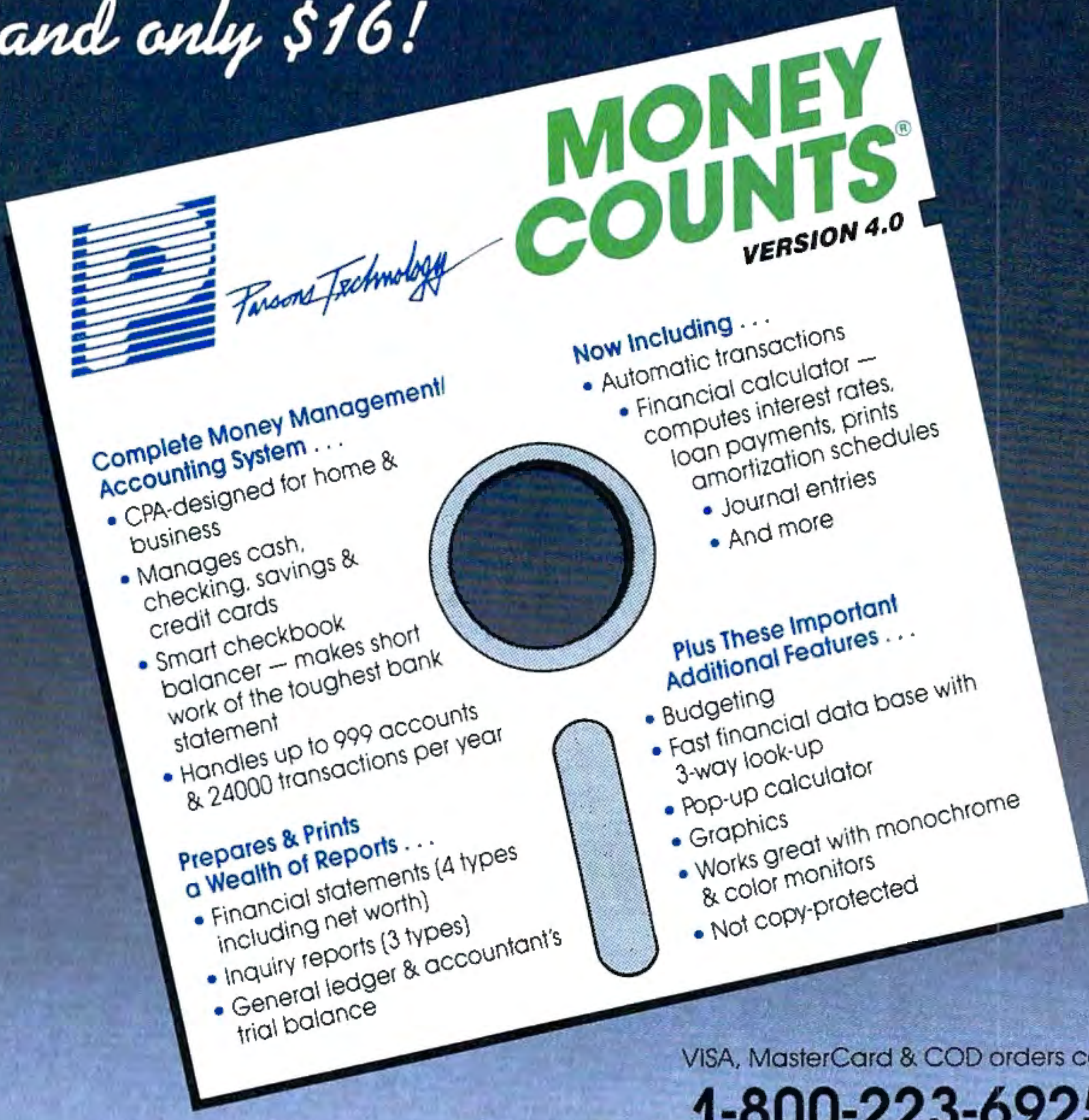
Advertising Director:
Bernard J. Theobald, Jr.

COMPUTE! Sales Office 212-315-1665

Address all advertising materials to:
Kathleen H. Ingram
Advertising Production Coordinator
COMPUTE! Magazine
324 West Wendover Avenue
Suite 200
Greensboro, NC 27408

The COMPUTE! subscriber list is made available to carefully screened organizations with a product or service which may be of interest to our readers. If you prefer not to receive such mailings, please send an exact copy of your subscription label to: COMPUTE! P.O. Box 10955, Des Moines, IA 50950. Include a note indicating your preference to receive only your subscription.

Complete and Easy . . . and only $16!



VISA, MasterCard & COD orders call:

1-800-223-6925

(In Iowa 319/373-0224)

Here it is. You won't find software that's easier to use:

- Requires no accounting knowledge
- Completely menu-driven with on-line help
- 93-page fully indexed tutorial manual (on disk)

A Proven Track Record. Hard to believe the $16 price? Don't worry. MoneyCounts® has been marketed for three years and is continually receiving rave reviews. There's **no** catch. This is a fully functional system that compares with products selling for $99 and more.

Same-Day Shipping. Order today and own MoneyCounts® for only $16! Add $3 shipping and handling (outside North America add $7). Iowa residents please add 4% sales tax. Send check or money order payable to Parsons Technology. VISA and MasterCard orders welcome.

Parsons Technology

MONEY COUNTS®

Dept. COM
6925 Surrey Drive NE
Cedar Rapids, IA 52402

$16 + $3 Shipping

MoneyCounts® requires IBM or compatible computer with two disk drives (or a hard disk drive), 192k or more memory, DOS 2.0 or later and printer.

Name: ______

Address: ______

City/State/Zip: ______

Phone: ______

Check ___ Money Order ___ VISA ___ MasterCard ___

Card # ______ Exp. Date ______

Readers Feedback

The Editors and Readers of COMPUTE!

If you have any questions, comments, or suggestions you would like to see addressed in this column, write to "Readers' Feedback," COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Due to the volume of mail we receive, we regret that we cannot provide personal answers to technical questions.

IBM Compatibility

What are the differences between IBM-compatible computers and MS-DOS computers? Can compatibles run the same software as IBM computers?

Hollie Vizier

Today most computers running MS-DOS are completely IBM-compatible. IBM PCs use an operating system called PC-DOS, and compatibles run MS-DOS. Both operating systems were written by Microsoft and function almost identically, from the user's point of view. The problem arises because computers that are not compatible with the IBM PC can run MS-DOS, too.

When the first MS-DOS machines appeared, some had problems with compatibility. These early machines ran MS-DOS, but they would not run all of the PC's software. During this period, all PC compatibles ran MS-DOS, but not all machines running MS-DOS were compatible with the PC.

Now, almost all MS-DOS machines are "true compatibles," meaning they can run all PC software without problems. Still, it's best to verify that an MS-DOS computer is compatible before you buy—especially if you are buying a used computer or an older model.

What Is DOS?

I own an Atari 800XL, and I have two questions. First, since I can load DOS both with and without BASIC, is DOS a machine language program? Second, is there a way to print machine language programs to a printer?

Mark Williams

In answer to your first question, yes, DOS is written in machine language. DOS (Disk Operating System) is an extension of the operating system that's responsible for handling file-oriented disk access. A short program in the ROM operating system attempts to boot a disk when you turn on your computer. If the disk in the drive is a DOS disk, the file called DOS.SYS is loaded into RAM. When you type DOS from BASIC, another part of DOS is loaded, the DUP.SYS file. This is the "DOS menu" program.

In answer to your second question, there are several ways to print machine language programs. If you have a disassembler, you can use it to print out reconstructed assembly language source code. Some disassemblers can't write to the printer, but can save the source to a disk file. If this is the case, you can print the file out from within a text editor or use the DOS Copy File option to copy from the disk file to the printer.

To print out the machine language program as numbers (not as assembly code) you can use this small program:

```
5 PRINT "This program prints a binary file as decimal numbers."
10 OPEN #1,4,0,"D:TEST.OBJ":REM TEST.OBJ is the name of the machine language file.
20 OPEN #2,8,0,"P:"
30 FOR I=1 TO 6:GET #1,A:NEXT I:REM Discard header info.
40 LINELENGTH=8:REM Eight numbers per line.
50 TRAP 1000
60 COUNTER=1
70 GET #1,A:PRINT #2;A;:IF COUNTER<>LINELENGTH THEN PRINT #2;",";
80 COUNTER=COUNTER+1:IF COUNTER>LINELENGTH THEN PRINT #2:COUNTER=1
90 GOTO 70
1000 PRINT #2;"END OF FILE":END
```

Time For The ST

I read "Time for BASIC" in the February 1987 "Readers' Feedback." It talked about timer functions for different BASICs of different computers, and it said that Atari ST BASIC didn't have any. I know that the ST has an internal clock, and I was wondering if there was a way to access it from BASIC.

Robert Fletcher

Four bytes starting at location 1210 contain the number of system clock ticks (which occur at a rate of 200 per second) since the system was booted. ST BASIC can access this location by using the PEEK function to return a four-byte integer. The program below examines this location and prints the number of seconds elapsed. ST BASIC doesn't fully support double precision numbers, so some accuracy will be lost as the number grows larger.

```
10 DEFSEG=0:DEFDBL I
20 I=1210
30 PRINT USING "##########"
   ;PEEK(I)/200
```

Super Text

I am desperately seeking a way to purchase a word processor that I saw advertised in one of your older issues. It's called *Super Text* and was published by Muse Software. Is Muse still in operation? If they are, is it possible to purchase *Super Text* by mail?

Michael Debyah

You're in luck. Muse Software has changed hands and has moved, but they still sell Super Text *(versions for the 64, Atari 400/800/1200XL, Apple II+, IIe, IIc, and IBM PC and compatibles). It can be purchased by mail for $25 plus $2 shipping and handling. Order from Muse Software, 5 West Ridgeville Blvd., Mount Airy, MD 21771.*

INPUT Problems

I am writing a utility program for my 64 that uses the INPUT statement to get and store a filename into a string variable. The problem occurs if the user types a comma or a colon. The program responds with an EXTRA IGNORED error message.

Is there some way to alter the INPUT statement so the user can type in commas and colons?

Matthew Bathke

The problem you mention is a common one when using the INPUT statement with strings. INPUT thinks commas and other separators are telling it that the variable has ended. If you type a double quotation mark as the first character of the input line, however, everything after the quotation mark will be assigned to the string variable.

WORK WITH US ON A PROFESSIONAL LEVEL.

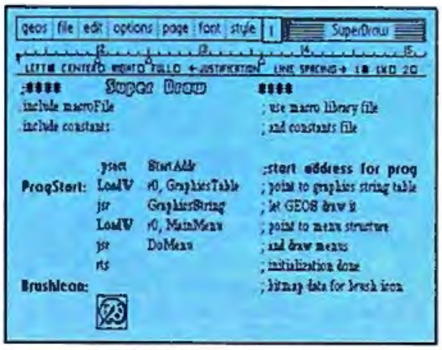
Some people really like working on their machines. And then there are some who prefer to do their tinkering under the hood. For those of you who can't wait to get your hands greasy, we proudly introduce geoProgrammer. The most sophisticated machine language programming tool on the market.

With geoProgrammer, you get the same kind of technology we use for developing our other products. Which means you can write supercharged software. Pump your own programs. And assemble just about any kind of application you can imagine.

It reads and writes and stomps on bugs.

The first three tools that any serious programmer needs are an assembler, a linker and a debugger. So we've installed all three, complete with sample GEOS applications that teach you the latest programming tricks in seconds.

geoAssembler not only reads directly from geoWrite files, but contains enough Pseudo Ops to allow all kinds of conditional assemblies. It has all the state-of-the-art features you'd expect, including some you probably never thought possible. For example, integrating graphics is as simple as cutting and pasting the image from geoPaint directly into your program.



*Also available for 80 col. C128's.

geoLinker ties your program modules together, supporting GEOS SEQ and VLIR applications and desk accessories. You can even use geoAssembler and geoLinker to create non-GEOS applications.

geoDebugger allows your program to be tested in memory with full symbolic disassembly, along with line assembly for patching code in memory. It also allows your code to be single-stepped or top-stepped, with sub-routines fully executed. It can stop a running program with one key, or use up to eight conditional breakpoints. When your program hits the breakpoint, it prominently displays the error in an overlay window, leaving the applications screen intact.

Pro enough for you yet? Well, that's only a *partial* list of what you're in for.

If you understood all that, read on.

Chances are that we lost a lot of readers by now. But if you're still with us, hang in there. You have the makings of a real GEOS pro. All you need now are the right tools.

And all of them come in this one handy box.

So if you're serious about programming, consider geoProgrammer. After all, you've got nothing to lose — except your amateur status.

To order call 1-800-443-0100 ext. 234

geoProgrammer $69.95

(California residents add 7% sales tax.)

$2.50 US/$5.50 Foreign for shipping and handling. Allow six weeks for delivery.

GEOPROGRAMMER

Berkeley Softworks

The brightest minds are working at Berkeley.

It would show more consideration for the user, however, to arrange things so he or she doesn't have to bother with starting certain strings with a quotation mark. Luckily, there's an easy way to do this—simply put the quotation mark character into the keyboard buffer yourself with two POKE statements. Use a line like the following in your program:

```
100 POKE 198,1:POKE 631,34:INPUT X$
```

The first POKE tells the 64 that there is one character in its keyboard buffer, and the next POKE puts a double quotation mark there. With the quotation mark as the first character in the buffer, the INPUT statement doesn't look for terminators—like commas and colons—but views the whole line as a string and assigns it all to the variable X$.

Micro-To-Micro Connection

At the National Educational Council in Pakistan, we have been working on Apple II computers for the last few years. Lately we have switched to IBM PCs, and we're worried that the data we've stored on the Apples will be lost. Is there any way to transfer the data between the two computers?

Faisel Saeed

I have two 64s and two modems. I would like to be able to hook them up without using the phone system. Is this possible?

Larry Sandillo

I own an Atari 800 XL and a Commodore 64 and use IBM PCs and Apples at work. I would like to be able to transfer ASCII files between these machines without calling someone on the phone. I had hoped that I could simply hook the modems together, but I soon discovered that no connection was being made.

Sonny Stephens

To transfer files between two microcomputers, you need to connect them with what is called a null modem *or* null modem cable. *To understand what a null modem does, just think about what happens when you talk to someone on the telephone. You speak into the mouthpiece, and the person at the other end of the line hears your voice on the earpiece of their phone. The phone's circuitry has crossed the send and receive lines. Otherwise, your voice would come out of the other phone's mouthpiece—an unsatisfactory arrangement.*

A null modem simply crosses the sending and receiving lines in an RS-232 connection between two computers. You can buy null modem cables at most computer stores at a price ranging from $15 to $50 dollars. If you're the adventurous type, you may want to make your own.

Atari owners will need an Atari 850 interface or equivalent. Commodore 64 and 128 users will need an RS-232 interface that plugs into the user port. One such interface, the Omnitronix Deluxe, has several switches on board that allow the interface to be used either with a modem or a printer. What's interesting about the printer configuration is that it turns the interface into a null modem, so no additional hardware needs to be purchased or made. For $49.95 plus shipping and handling, you can order the interface from Omnitronix, 760 Harrison St., Seattle, WA 98109.

To make your own null modem, you'll need some parts, available from Radio Shack, plus a soldering iron. The parts you'll need are:

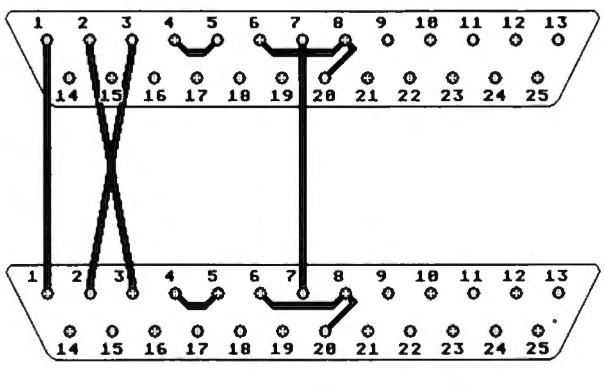
- *2 25-pin D submini connectors (part # 276-1547—male, or 276-1548—female)*
- *2 25-pin D submini hoods (part # 276-1549)*
- *6 feet 4-conductor cable (part # 278-365)*

First, cut six short pieces of wire for some jumper connections. Pins 4 and 5 on each plug need to be connected, as well as pins 6 and 8, and 8 and 20 (see the accompanying diagram). Make the pieces of wire short enough for each connection and solder them.

Next, push the 4-connector cable ends through the hoods. On one connector, solder the black wire to pin 1, the green wire to pin 2, the red to pin 3, and the yellow to pin 7.

On the other connector, solder the black to pin 1, the red to pin 2, the green to pin 3, and the yellow to pin 7. (If you'd prefer, you can use any color with any pin as long as pins 2 and 3 are crossed.)

Now attach the hoods with the screws provided, and your null modem cable is ready to go. Appropriate terminal software for each computer is all you need to transfer any file.



Connectors Viewed From Rear

Two Generations

I am a beginner computer user and am a new subscriber to your magazine. I have an Atari 800 with 48K RAM. I've seen a lot of advertising for the Atari ST models. What is the difference between the older Ataris and the newer ones?

Charles E. Lemieux

The first generation of home computers, the Apple II, Atari 800, and Commodore 64, were all built around the 6502 microprocessor. These computers were designed to be used with a standard television. They use 40 columns of text on the screen and have a limited number of colors (the Atari has 256 colors maximum). All three computers first used tape drives for storage, but then they moved up to 5¼-inch disk drives for storage. Game, word processor, and spreadsheet programs are popular with users of these machines. Many people have found that these computers can do everything they need to have a computer do.

The most recent machines from Commodore, Atari, and Apple are the Commodore Amiga, Atari ST, and Apple Macintosh. These machines use the powerful 68000 microprocessor, which can directly access 16 megabytes of memory (as opposed to the 64K bytes that the 6502 can access.) These computers were designed from the start to use colorful, hi-res monitors (the Mac II is the only Macintosh that can display color). In general, the new computers are faster, have better graphics and sound, and are more useful than the previous generation of home computers. Of course, they are more expensive as well.

Shifted Spaces On The 64

Most of the time, I can save and load disk files without any problems. Occasionally, however, when I save a file such as FARC 9/11/87, it appears in the disk directory as "FARC"9/11/86. Not only is the closing quotation mark in the wrong place, but I also have trouble accessing the file. What am I doing wrong, and is there any way I can correct my mistake?

Charles N. Tanton

Your problem has to do with shifted spaces. Although they look exactly the

EASY Working™
From Spinnaker

Easy Working™ is a line of high quality, low cost personal productivity programs designed for the person who wants to spend time doing useful work, not struggling with complex computer commands and long manuals. Each Easy Working product is carefully designed to work alone or in combination with other members of the Easy Working family.

On-screen menus provide "at-a-glance" summaries of the features of the program. When you select a feature from the menu bar, a menu drops down to show all of the available options. Every command available is displayed in the drop-down menu. You will find that you will spend less time getting to know the program and more time on your projects. And since each member of the Easy Working family uses the same command structure, once you learn how to use one, you'll find a head start on learning to use the other two. But don't be fooled. Easy doesn't mean incomplete. Easy Working products are full-featured.

Easy Working™: The Writer™:
Complete word processing features allow you to create your letters and reports and arrange them to suit your needs, includes 100,000 word spellchecker. Editing functions include insert, delete, cut, paste, and copy.

Easy Working™: The Filer™:
A multi-purpose program which simplifies the storage, selection, and reporting of information. Maintain or create mailing lists, inventories, club memberships, and other types of information.

Easy Working™: The Planner™:
The Planner provides you with all of the professional features of an electronic spreadsheet. Extensive mathematical operations easily create your spreadsheet for budgeting, tax calculation, expense reports, financial statements, and other applications.

The Writer, The Filer, and The Planner fully integrate with each other.

Available at fine software dealers or for orders only call:

1-800-826-0706
Others call: (617) 494-1200

Apple II+, IIe, IIc	$9.95
Commodore 64/128	$9.95
IBM/PC/AT	$9.95

SPINNAKER®

Spinnaker Software, Corp., One Kendall Square, Cambridge, MA 02139 *same, a normal space is CHR$(32), while a shifted space is CHR$(160). Most likely, when you entered your filename the SHIFT LOCK key was depressed.*

Commodore DOS reserves 16 bytes for each filename in a disk directory. When a filename is less than 16 characters in length, the remaining bytes are filled with 160's—shifted spaces. So, when your disk drive saw the shifted space following FARC, it assumed that it had reached the end of the filename and printed a closing quoation mark.

To access your file, you must be sure to enter the filename exactly as you did when you created the file—shifted space and all. Next time, simply avoid using shifted spaces in filenames.

No More Swaps

I have a single drive Amiga system. Whenever I try to work with a non-Workbench disk, the Amiga always asks for the Workbench disk whenever I give a command. This results in a lot of annoying disk swaps. I tried creating a command directory in the ramdisk, but when all the CLI commands are copied there, I don't have much room left (I only have 512K). I have overcome this problem using the PATH ADD command on Workbench 1.2. Instead of copying all the CLI commands into RAM, I just copy the ones I know I'll use frequently (like CD, DIR, TYPE, and so on). I then type "PATH RAM: ADD". This works very nicely, because the Amiga first looks in RAM for commands. If it finds the command it's looking for, it quickly loads and executes it. If it doesn't find it, the Amiga will ask for the Workbench disk (which doesn't happen that often because frequently used commands are in RAM). This setup is very convenient because it gives you the power of commands in RAM, while still leaving plenty of memory to do other things.

Haley Carter

Thanks for the tip.

Printer-Only Output For Apple

In the December issue of COMPUTE!, Richard J. Kuhn asked how to stop printer output from also being sent to the screen. There were several errors in the answer, so we print the corrected answer in full:

Almost every printer interface for the Apple uses the same command for turning off video output. To issue this command, you must first open the printer for output with a PR#1. Next, print a CTRL-I (CHR$(9)) followed by the desired printer interface command. In your case, you want to send the three characters 80N, telling the interface to print 80-column text and to turn off video output. The following program illustrates this technique by printing a familiar sentence to the printer, and not to the screen.

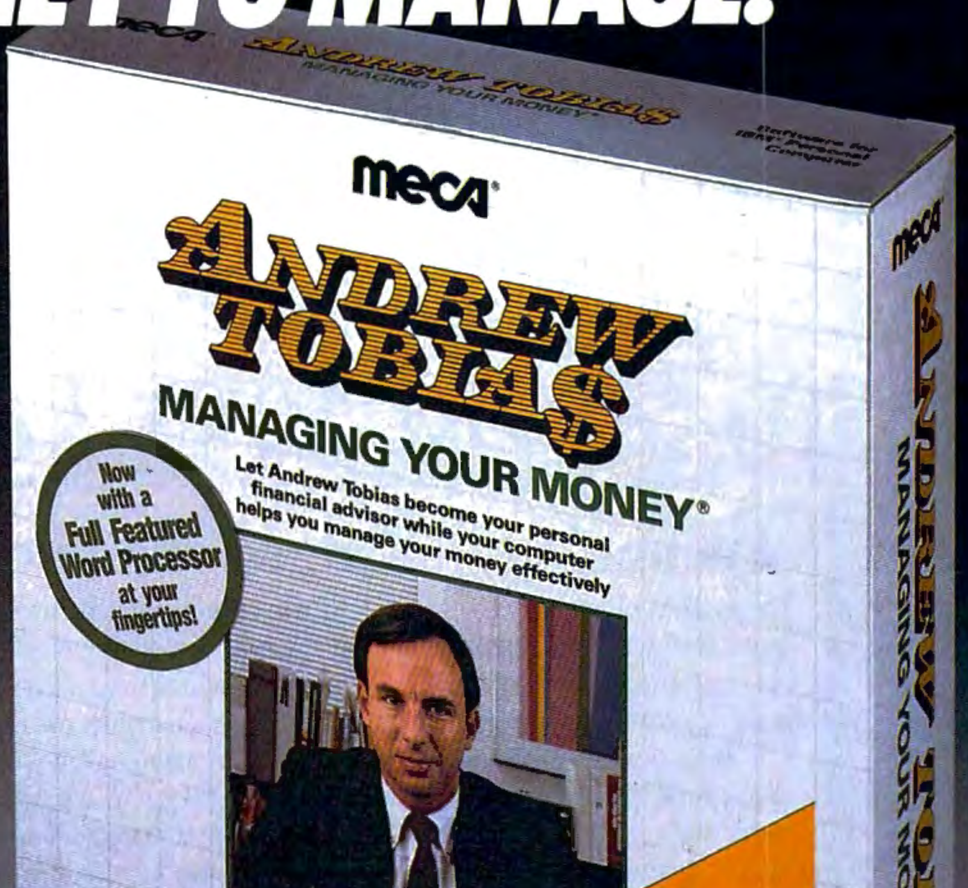
```
10 PRINT
20 PRINT CHR$(4)"PR#1":REM OPEN
   PRINTER AS OUTPUT DEVICE
30 PRINT CHR$(9)"80N":REM 80-
   COLUMN TEXT/NO VIDEO
   OUTPUT
40 PRINT "THE QUICK BROWN FOX
   JUMPED OVER THE LAZY DOG."
50 PRINT CHR$(4)"PR#0":REM RESET
   40-COLUMN SCREEN AS OUTPUT
   DEVICE
```

There are several standard printer interface commands available to Apple owners. All are preceded by the CTRL-I command character. It is even possible to send these commands directly from the keyboard. For example, if you use a serial printer, you can change your interface's baud rate to 9600 mode by typing the following lines (press RETURN after each line):

```
PR#1
CTRL-I 14B
PR#0
```

This changes your serial interface's output to 9600 baud, overriding whatever baud rate the dip-switch settings may specify. Of course, turning your computer's power off and on resets the interface to its default condition.

MANAGE YOUR MONEY WHILE YOU STILL HAVE SOME MONEY TO MANAGE.



Managing Your Money® is the most comprehensive, easy to use, personal financial software tool available today.

It's everything you'll ever need, it's all you'll ever need, to master your own personal finances:

- PERSONAL BUDGETING
- CHECK WRITING
- NEW TAX LAWS
- TAX PLANNING
- INSURANCE PLANNING
- FINANCIAL CALCULATOR
- PORTFOLIO MANAGER
- NET WORTH
- HOME BANKING
- AND MUCH, MUCH MORE.

"NOTHING BEATS MANAGING YOUR MONEY."

PC Magazine, 1987

Software that makes your personal computer worth having.

meca®

For 256K IBM PC, XT, AT, PCjr, PS/2, TANDY 3000, 1200HD, 1000, APPLE IIe (128K, Two Drives) IIc, IIGS. Available in 3½" and 5¼" diskettes.

6 New Technologies That Will Change Your Computer And Your Life

Technology changes our lives.

Today that change is constant and the rate at which it touches us is accelerating. That acceleration is fueled by many things: greater numbers of researchers, larger amounts of money applied to specific research problems, a greater base of knowledge from which to mount assaults on new questions, and more.

But one essential ingredient in the increasing pace of technological change is the computer. Computers amplify the capabilities of researchers, permitting larger and more sophisticated theoretical modeling in the early stages of research, and more effective design and implementation as research is translated into products.

Much modeling today is focused upon the computer itself. The evolution of computers over the past 20 years is nothing short of stunning. The next two decades promise to be even more amazing.

Here, we cover six areas where breakthroughs have recently been announced or are eagerly anticipated. Some of the areas—hypertext, superconductors—have been around a while, at least in theory. Others—optical disc storage and retrieval, and superchips—are already growing large industries. Parallel processing is already finding its way into the marketplace; the development of neural networks is still primarily taking place in laboratories. Flat screens promise to affect not only computers, but also the most ubiquitous of technologies, the TV.

Six exciting technologies; it could easily have been sixty or six hundred. Let's look at some of the ways in which changing technology will change computers and our lives.

Computers That See, Computers That Think

Electronic brains, they were called, *thinking machines* that filled gymnasiums with tubes and wiring connected for the purpose of making calculations. Forty years or so ago, people wondered how long it would be before computers could think.

The path that began with those machines—that of digital information electronically processed in analog devices—has resulted in today's information revolution. We still rely upon digital information, processed now on digital machines. Today's computers churn through data at a rate far greater than that of the brain.

But we still haven't seen any machines that actually think or, for that matter, that process information the way our brains process information. And there are many things we take for granted as a result of our brains, things that computers simply can't do.

The Persistence Of Vision Research

Vision is a good example. Despite years of research backed by billions of dollars, artificial vision remains very much in its infancy. Computerized robots, equipped with state-of-the-art artificial eyes, can move at a mile or so an hour along carefully proscribed tracks, stopping when they encounter something they haven't seen before.

When *we* see something, the process involves input via our eyes, transmission of the input along nerve cells to the brain, translation of data inside the brain, and interpretation and response based on our recognition of visual patterns that coincide with our stored memories of what we have seen before—our collected inventory of associations and reactions, all of it taking place not only instantly, but also *constantly.*

Like the brain, computers process and sort information; unlike the brain, most computers work with one piece of information at a time. The data is retrieved, worked on, stored, and the processor moves on the next piece of information.

The neurons in our brains, however, accomplish thousands of processing tasks a second, sharing data throughout the brain, all of it taking place with rough simultaneity despite the fact that neurons work at far slower physical speeds than electronic computers. Each neuron, though, may be thought of as a processor, sharing associations with other neurons to which they are linked by thousands of branching axons, which in turn branch again in fibers called dendrites. The fibers stretch and intertwine, forming a network that can be visualized as not unlike the tangled roots of a tree. There are billions of neurons in our brains, kilometer after kilometer of fiber along which information travels, is received, processed, and shared.

These are the challenges facing researchers wishing to duplicate, or even to imitate, actual brain functions: To accomplish the processing of information in parallel, and to achieve an effective network of shared associations. These are hardware and software challenges that promise great rewards but pose challenges as large as anything the computer industry has undertaken.

Processing In Parallel

Now a new generation of computers is being developed. This generation attempts to mimic that sort of processing, dividing computational tasks among hundreds or thousands of processors simultaneously.

It's easy to see the advantages: Ten processors, each working on one-tenth of a problem, could arrive at an answer ten times faster than a single processor faced with the same problem. The time spent on actual computations may not be dramatically reduced; great time savings, though, are realized in the amount of time the processor spends retrieving and storing information. A parallel system could accomplish all necessary computations before the information is returned to memory.

There are other, more subtle advantages over conventional, one-step-at-a-time serial computing. Consider a database: essentially, a set of memories. On a serial computer, if you were searching the database for all references to, say, parallel and neural processing, the machine would work its way through the database one piece of information at a time, moving sequentially through the store of information.

A parallel system, however, would be able to pursue many or all of the references at once, moving through the multiple sections of the database simultaneously, returning your answer far more quickly than is possible on traditional computers.

There are several approaches to parallel processing architectures themselves. Closest to conventional computers is the *Single Instruction Multiple Data* approach, which as its name implies uses one instruction to operate on more than one piece of data at a time. *Multiple Instruction Multiple Data* raises the stakes a bit: Each processor in the parallel structure can operate independently, following different instructions. *Dataflow* architectures go even farther, with processors freed from any serial sequence; in a dataflow architecture, each processor goes to work when data becomes available, sharing its results with other processors which go to work and communicate their results, and so on, all of the flow overseen by sophisticated software which guides the flow and dynamically reconfigures the array of processors for most efficient operation at any given time.

Next Step: Neural Networks

Parallel processing is making steps toward solving hardware problems associated with making computer operation more efficient, more thorough, more brain-like. *Neural processing,* the development of neural networks within computers, seeks to mimic the networking of information with the human brain. The sharing of processing tasks must be matched by the sharing of input along several paths at once, with those paths focusing upon shared and recognized patterns, associations, memories, and responses.

In a successful neural processing network, pattern recognition would be reinforced each time a pattern is recognized, just as activity strengthens the neurons in our brains. Furthermore, to be effective, a neural network must be able to build associations among patterns, with those associations further strengthening nodes and interconnections within the computer's memory. Thus, the pattern recognized yesterday would be reinforced when seen again today; in the case of artificial vision, a computer sophisticated enough would be able from its store of patterns to navigate its way through any set of obstacles or objects, recognizing them and responding in the proper manner.

Neural networks for neural processing remain largely theoretical constructs, with research and modeling being funded by the still growing government and industrial interest in artificial intelligence applications. Some of those applications will use neural networks in concert with parallel processing computers; others are following different paths.

Parallel processing is closer to reality, with a variety of companies actually manufacturing and selling parallel systems. Some industry observers feel that these systems are the strongest competition yet for serial supercomputers such as the Cray.

Huge obstacles remain before true neural computers are in place, but huge obstacles have been overcome before. The actually perceptive and ultimately intuitive computer, as opposed to machines that only partially mimic thinking, is a development on which we should keep our eyes, ears, and minds—all processing information simultaneously—in the months and years ahead.

—*Keith Ferrell*

Laser Disc Technology: All The World On A Disc

What's hotter than the compact disc? It makes old songs crystal clear, and it lets current musicians make the purest-sounding music ever recorded. Even if you don't have a CD player yet, your favorite radio station probably uses one to make radio sound better than it ever has before. The sparkling star of the audio world is the compact disc.

But compact discs (CDs) aren't just for music. Even now, CD technology is beginning a successful crossover act that is taking it into the world of computers. You may already have heard about some of the varieties of Laser Disc storage—CD ROMs, WORM drives, and CD-I.

Why the sudden flurry of success for CDs? Mass production capability is the answer. The tremendous size of the worldwide consumer market prompted the industry to quickly boost both CD player and disc production into the millions. At volumes like these, just about anything can be produced at a reasonable price. Now that the technology has been perfected by the consumer industry, the smaller home and business computer industries can take advantage of it.

CD ROM

The first computer application for compact disc technology is CD ROM (Compact Disc–Read Only Memory). The term refers to the permanence of the data. Like computer ROM chips, CD ROMs come to the consumer with information already recorded on them. However, CD ROMs contain much more information than common silicon ROM chips. A single CD ROM can hold 550 megabytes—more than enough room for an entire encyclopedia of information. And you can change CD ROMs just like you change floppy disks. Imagine having an entire library of information that fits on a bookshelf.

With so much information available, how can you possibly sort through it? CD ROMs are often cross-indexed so thoroughly that the indexing takes up more of the available space on the disc than the information itself. Designers are working on new techniques to make data retrieval faster and more natural. Hypertext may play a part in solving this problem.

A CD ROM player is a modified Compact Disc player. Already, IBM PC users can take advantage of CD ROMs. Atari showed a CD ROM device for their ST line of computers at the November COMDEX computer show. It shouldn't be long before interfaces for other computers become available.

What kind of software can you expect to find on CD ROMs? Mostly information that has already been translated into electronic form. Hundreds of titles are available, covering everything from agriculture to black fiction to the *Wall Street Journal*. We recently received a single CD ROM disc that contained the entire public domain library (605 floppies worth) of a large users' group. As CD ROMs become more commonplace, more and different kinds of information will become available.

WORMs

The next step in laser disc technology is the WORM (Write Once Read Many) drive. This is a CD drive that can record data as well as play it back. Although being able to write only once sounds restrictive, the great amount of storage available on the CD makes this limita-

IT'S NEW, COMPATIBLE, EXPANDABLE, PORTABLE AND EVEN INCLUDES COLOR GRAPHICS.

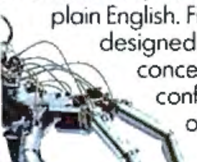
It's a unique, up-to-date reference series on computing unlike any other. It's compatible with any micro, mini, mainframe or supercomputer you use, own, or may buy in the future.

It's UNDERSTANDING COMPUTERS from TIME-LIFE BOOKS.

A NEW AND BETTER WAY TO PRESENT COMPUTERS.

UNDERSTANDING COMPUTERS is designed, written and illustrated especially to help you learn a lot more about computers and computing, no matter what your level of experience is now.

Each volume of UNDERSTANDING COMPUTERS takes you on an intensive, inside tour of a different aspect of computing in plain English. Full-color graphics, including artwork designed exclusively for this series, clarify every concept. So you enhance your knowledge and confidence, and go beyond the ABCs to the XYZs of topics like peripherals, programming languages, software and graphics.



Guided by complex software, robot fingers grip an egg with near-human gentleness.

For example, *Computer Basics,* your first volume, explains why most computer operations are based on only two decisions: True or False. And then goes on to cover such topics as chip design and how logic gates work. *Software* reveals how translators, utilities, memory mapping and languages are used in a programmer's toolbox. *Computer Images* discusses the secrets of creating spectacular graphics.

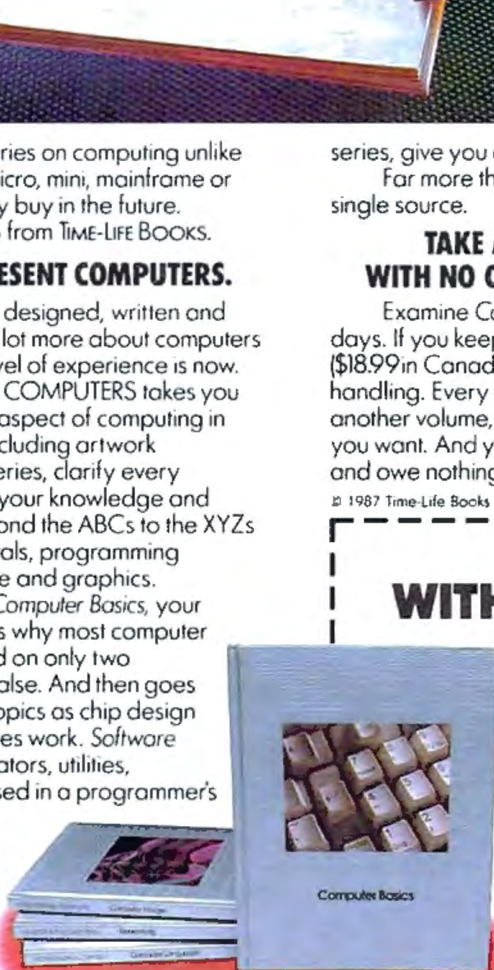
These volumes and others in the series, give you an encyclopedic knowledge of computing.

Far more than you can get from any single source.

An 8-processor chip for a "non-Von" computer. Actual size at right.

TAKE A 10-DAY LOOK WITH NO OBLIGATION TO BUY.

Examine *Computer Basics* free for 10 days. If you keep it, pay only $14.99 ($18.99 in Canada), plus shipping and handling. Every other month you'll receive another volume, also on a free 10-day trial. Keep only the ones you want. And you can cancel anytime. Or return *Computer Basics* and owe nothing. Mail the reply card or coupon now.



Each volume is a big 9¼" x 11⅛" hardcover book with approximately 128 pages.

START HERE WITH COMPUTER BASICS.

If card is missing, mail this coupon to:
TIME-LIFE BOOKS
Box C-32066, Richmond, VA 23261-2066

YES! Send me *Computer Basics,* as my introduction to UNDERSTANDING COMPUTERS. I agree to the terms outlined in this ad.

D1CJ95

Name ____________________

Address ____________________

City ____________________

State or Province __________ Zip or Postal Code __________

All orders subject to approval. Price subject to change.

TIME LIFE BOOKS

UNDERSTANDING COMPUTERS

tion acceptable.

For example, suppose you write a BASIC program that's 10K long. That translates to about 1/36 the storage capacity of a floppy. But that same program would use only 1/55,000 of the space of a CD ROM. You could change and resave your program as many times as you like with no noticeable loss of capacity.

Some WORM discs can be erased. So far, erasing a disk means erasing the whole disk. Many companies are working on drives that could be called CD RAM drives, which would allow you to actually delete old copies of your programs and data.

CD-I

There's a new standard that could change entertainment and education: Compact Disc Interactive (CD-I). A CD-I player is a combination audio CD player, home computer, and videodisc player. The three parts blend together to create a machine that's vastly more capable than the sum of its parts. Several well-known software publishers—including Electronic Arts, Spinnaker Software, and Aegis Development—are working to find out just what can be done on a CD-I machine.

Let's take a closer look at the elements that make up a CD-I machine. Foremost is a CD player that can play all current and future CD audio disks. The CD player also handles broadcast TV-quality video that can be displayed on your TV or monitor. The computer built into the CD-I player is based on the powerful 68000 microprocessor, the same one found in the Macintosh, Atari ST, and Amiga computers. It will have its own powerful graphics system which can be integrated with the CD video.

Imagine a possible CD-I application. You place a disc titled *Biology Class* in the CD-I player. A high-school classroom appears on your television screen. You use the CD-I player's controls to become an active participant in the simulated classroom. Walk down the aisle and stop at a desk. The student here might be dissecting a frog. Help him or her find various organs in the frog. After you've finished, take a look at the other experiments. You might want to help conduct Mendelian genetics experiments with mice or see how sunlight affects a sunflower.

CD-I is far more ambitious than CD ROM, and it is stirring up a great deal of controversy. Some industry observers doubt that consumers want to interact with their televisions. Others feel that limitations—for example, the format is not capable of full-motion video—in the standard will kill its chance for consumer acceptance. To further complicate matters, a competing standard known as DVI (Digital Video Interactive) has appeared.

Regardless of how CD-I evolves or the CD-I/DVI battle works out, laser discs are already beginning to play an increasingly varied and important role in our lives. They're changing the way we see, hear, and think.

—Rhett Anderson

Superconductors: A Technology Coming Of Age

Incredibly fast mainframe computers that fit on your desktop, high-speed trains that float above their tracks, and long-distance power lines that transmit electricity without losing any power: These are just some of the products that may result from superconductor technology.

Recent scientific breakthroughs have superconductors working their way into the public's eye. Promising great new products, superconductors may have as much impact on modern life as the invention of the transistor.

New Technology That's Old

The phenomenon of superconductivity was first discovered in 1911 by a Dutch physicist named Jeike Kamerlingh Onnes. He found that by cooling mercury to −452° Fahrenheit (approximately 0° on the Kelvin scale, also known as absolute zero), this common material carried electricity with absolutely no resistance. Resistance slows down the flow of electrons, decreasing voltage and increasing the wasteful (often damaging) dissipation of heat.

At such extremely low temperatures however, superconductivity is practically useless. To keep the conductant cool, it had to be immersed in liquid helium—not the most practical or inexpensive of operating conditions.

Recently, by combining ceramic materials with small amounts of elements known as *rare earths*, physicists have been able to produce superconductors that operate at much higher temperatures. It took several years to move from −452° F to just −424° F, but within the past year, we have gone all the way from −424° F to 9° F—a jump of over 430 degrees.

With these recent developments, researchers have superconductors working at room temperatures. There has even been evidence that superconductivity in a ceramic material may be possible at a scorching 90° F. Allowing current to flow without any resistance, superconductors open up a whole new universe of possibilities in electronics.

Super Products

To minimize the loss of electricity, power plants must be placed relatively close to the population that they serve. With nonresistant superconductor transmission lines, power could be ported great distances without any loss of energy.

One of the most talked about superconductor applications is the *maglev* (magnetic levitation) train. A characteristic of superconducting material is the powerful magnetic field that they create. By generating superconductive magnetic fields with opposing polarity, trains could travel 6 to 12 inches above their tracks, being pulled along by other, constantly alternating magnetic fields. Without the wheel-against-track friction experienced by regular trains, the maglev could attain cruising speeds of up to 300 miles per hour—twice the speed of highly touted Japanese "bullet" trains.

Superconductors may have their most profound effect on computers. Today's computers rely on a multitude of tiny switches called *gates* to process information. With superconductive gates, computers could operate literally hundreds of times faster than they do with conventional silicon gates. And because superconductors do not generate heat, computers can be made much smaller. No longer requiring fans or extra ventilation space, electronic components can be packed together without overheating and breaking down.

In the never-ending quest for faster, more powerful computers, superconductors may be the answer to a long sought after dream. Imagine a box similar in appearance to a pocket television running Macintosh software at ten times the normal speed, or a desktop PC more powerful than a Cray supercomputer. Computers could be made so small that even the most superior machines would be considered portable by today's standards.

Superconductors are already being used in medical scanning equipment and giant atom smashers. But until certain barriers are broken—such as tolerance to warm temperatures, ability to handle high voltages, and flexibility of materials (the ceramics used in high-temperature superconductors are far from flexible)—superconductors will simply be a phenomenon at which physicist and science students will marvel.

—Randy Thompson

Super Chips: Smaller, Faster

Today, microcomputer chip designers are reaching the physical limits of a technology that began less than 30 years ago. Silicon Valley and Japan, Inc. are locked in a battle to be the first to overcome these limits and develop a new generation of microcomputer chips that contain one billion transistors. What will this new generation of microcomputer chips mean to the average person? To answer this question, we need to go back to the New York Coliseum on March 24, 1959.

As the Fifties draw to a close, Texas Instruments has made a mad rush to produce a few working prototypes of one of its engineer's latest creations—Jack Kilby's integrated circuit. They plan to announce this circuit-on-a-chip at the annual Institute of Radio Engineers meeting. At the meeting, TI's president predicts that the integrated circuit will be the most important invention since the transistor. His enthusiasm is not shared by everyone.

Electronics magazine gives a blow-by-blow description of the meeting's new innovations in a special issue, but doesn't even mention the integrated circuit (IC). And when the new IC is mentioned in the press, it is regarded with a cold eye. (After all, silicon is considered an inferior material.) Few people at that meeting guessed that the integrated circuit would soon change the world.

The Chip

The transistor had been the champion of technology just a few years before, solving many of the problems encountered with vacuum tubes. Transistors were smaller, more efficient, and more reliable, but there was one important problem they didn't begin to solve.

No matter how small, efficient, or powerful you could make transistors, you were left with the interconnections problem, the dilemma of how to physically make the huge number of connections between transistors required by sophisticated devices. This problem may not be apparent when you think of the connections needed in a transistor radio, for example, but, with computers, the numbers quickly grow into the millions.

Chips solved this interconnections problem by placing several transistors on a single silicon surface about the size of a postage stamp.

The number of transistors per chip has grown from less than ten in the earliest chips to hundreds of thousands in the chips marketed today. In fact, the chip's capacity doubled every year from its creation until just a few years ago. But by putting more and more components on a chip's surface, designers have started to run into physical limits, and the interconnections problem has resurfaced.

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.

........When you want to talk price.

DRIVES

Atari
AA314 DS/DD Disk (ST)..........$199.00
AA354 SS/DD Disk (ST)............119.00
SHD204 20 Mb ST Hard Drive...569.00
Commodore
Amiga 1020..............................189.00
Amiga 1010 3½"......................219.00
1541C......................................179.00
1571.......................................239.00
1581 3½" External...................229.00
Indus
GT Disk Drive Atari XL/XE........ 179.00
Racore
Jr. Expansion Chassis.............299.00
Seagate
20 mb ST-225 Hard Drive Kit.....339.00
Supra
20 Meg Hard Drive (Amiga).......749.00
20 Meg Hard Drive (ST)............559.00
Xebec
20 mb (Amiga)..........................899.00

MONITORS

Amdek
Video 300 Amber Composite.....139.00
Commodore
Commodore 2002.....................319.00
Amiga 1080 Hi-Res Color..........299.00
Commodore 1802.....................199.00

Magnavox 8502 13" Composite **$169**

Magnavox
8505 RGB/Composite...............199.00
8562 RGB/Composite...............249.00
NEC
JC-1402P3A Multi-Sync................Call
Princeton Graphics
MAX-12 12" Amber TTL...........139.00
Taxan
Model 124 12" Amber..............119.00
Thomson
4120 RGB/Composite...............249.00
Zenith
ZVM 1220/1230 Composite.(ea.) 99.99

MODEMS

Anchor
Volksmodem 1200...................$89.99
6480 C64/128 1200 Baud..........119.00
VM520 ST520/1040 1200 Baud.129.00
Atari
SX212 (ST)...............................89.99
Best
1200 Baud External..................119.00
Commodore
Amiga 1680-1200 BPS.............169.00
CBM 1670 & C-128)....................99.99
Everex
Evercom 1200 Baud Internal.......99.99
Hayes
Smartmodem 300.....................139.99
Smartmodem 1200 External.....289.00
Novation
Parrot 1200...............................89.99
Practical Peripherals
Telecom Package....................129.00
Supra
MPP-1064 AD/AA C64................69.99
2400AT 2400 Baud Atari...........169.00

U.S. Robotics 1200 External **$99.99**

U.S. Robotics
2400 Baud Internal...................189.00

DISKETTES

Maxell
MD1-M SS/DD 5¼".....................8.49
MD2-DM DS/DD 5¼"...................9.49
MF1-DDM SS/DD 3½"...............12.49
MF2-DDM DS/DD 3½"...............18.49
Sony
MD1D SS/DD 5¼".......................6.99
MD2D DS/DD 5¼".......................7.99
MFD-1DD SS/DD 3½"...............11.99
MFD-2DD DS/DD 3½"...............16.99
Hewlett-Packard Calculators
28C Scientific Pro.....................199.99
18C Business Consultant..........139.95
12C Slim Financial.....................74.99

PRINTERS

Atari
1020 XL/XE Plotter...................$31.99
XDM121 Letter Quality.............159.00
XMM801 XL/XE Dot Matrix.......185.00
XMM804ST Dot Matrix.............179.00
Brother
M-1109 100 cps, 9 pin...............199.00
M-1409 180 cps, 9 pin...............319.00
Citizen
MSP-10 160 cps, 80-Column.....279.00
Premier 35 cps Daisywheel.......489.00
C.Itoh
315P 132 Column Prowriter......549.00

Epson EX-1000 300 cps 132 col. **$499**

Epson
LX-800 150 cps, 80-column...........Call
FX-86E 240 cps, 80-column...........Call
FX286E 240 cps, 132-column........Call
LQ-1000 24 Wire, 300 cps.........559.00
LQ-850 330 cps, 80-column...........Call
LQ-1050 330 cps, 132-column.......Call
Hewlett Packard
Thinkjet..................................379.00
NEC
Pinwriter 2200 24 Wire.............379.00
Pinwriter 660 24 Wire...............459.00
Pinwriter 760 24 Wire...............679.00
Okidata
Okimate 20 Color Printer..........129.00
ML-182 120 cps, 80-column......219.00
ML-192 + 200 cps, 80-column...329.00
ML-193 + 200 cps, 132-column.479.00
Panasonic
KX-1080i 144 cps, 80-column....179.00
KX-1091i 194 cps, 80-column....189.00
KX-P3131 22 cps Daisywheel....289.00
Star Micronics
NX-10 120 cps, 80-column........169.00
NX-10C 120 cps, C64 Interface.189.00
NX-15 120 cps, 132-column......319.00
Toshiba
P-321 SL 216 cps, 24-Pin..........539.00
P-351 II 300 cps, 24-Pin............889.00

In the U.S.A. and in Canada

Call toll-free: 1-800-233-8950

Outside the U.S.A. call 717-327-9575, Telex 5106017898, Fax 717-327-1217
Educational, Governmental and Corporate Organizations call toll-free 1-800-221-4283
CMO. 477 East Third Street, Dept. A2, Williamsport, PA 17701
ALL MAJOR CREDIT CARDS ACCEPTED.

POLICY: Add 3% (minimum $7.00) shipping and handling. Larger shipments may require additional charges. Personal and company checks require 3 weeks to clear. For faster delivery use your credit card or send cashier's check or bank money order. Pennsylvania residents add 6% sales tax. All prices are U.S.A. prices and are subject to change and all items are subject to availability. Defective software will be replaced with the same item only. Hardware will be replaced or repaired at our discretion within the terms and limits of the manufacturer's warranty. We cannot guarantee compatibility. All sales are final and returned shipments are subject to a restocking fee.

The State Of The Art

One way to trace the progress made in computer chip design is to look at the advances made in the random access memory, or RAM, of personal computers in the last few years.

RAM is measured in *bytes*, with one byte being roughly equivalent to one character of text. Not long ago, 4K (4 kilobytes) comprised the average memory for a home computer. Soon after, 16K became the standard. When the Commodore 64 came along, 64K was the yardstick, and then the IBM PC set 128K as a minimum.

Today, 512K is usually the lower limit for a PC, and computers using the Motorola 68000 processor are often considered underpowered if they have less than one megabyte (1024K) of memory. The recent emergence of machines that can address *gigabytes* (1024 *megabytes*) of RAM makes 64K seem small indeed.

It might seem to the casual observer that memory and processing power would simply continue to increase every year. Unfortunately, chip designers have encountered a brick wall of sorts—at least with traditional chip design. The problem is easy to understand. As more and more components are packed with ever increasing density on a chip's surface, interconnections can become so narrow that they don't allow enough electrons to pass through them to turn the chip's microscopic transistors on and off.

Exactly what *are* the limits of current chip technology? George Heilmeier, senior vice president and chief technical officer at Texas Instruments, Dallas, says that, with traditional transistor structures and current techniques, a density of about 20 million devices per chip is the limit. This may seem like more chip than anyone would ever need, but designers are already working on the billion-transistor chip they think future applications will demand.

The Next Generation

There are two ways to increase the number of components on a chip. One is to make the components smaller (and this is essentially what has been done for the last 25 years). The other is to leave the components the same size, but make the chip larger. Both methods are being explored.

To try to get more transistors on a chip, designers are experimenting with new ways of putting components on chips, using new materials, and redesigning the nature of chip structures.

One new way of putting components on chips is to stack them on top of each other. This makes the chip a three-dimensional object rather than the flat two-dimensional surface of the recent past. One new structure that may bear fruit is called a *superlattice*, which is capable of performing the same tasks as a transistor, but may be only a few atoms thick.

As for new materials, sapphire is one of the leading candidates, but the cost of sapphire is over four times that of silicon. Economic barriers are one of the principal obstacles in designing new chips.

Chips Or Wafers?

Among the most exciting areas of research involves mounting chips together on wafers instead of singly as is the current practice—essentially making bigger chips. Using this technique, the processing power of the biggest computer in the world—the Cray 2—could fit into a 12 × 8-inch box.

Mounting chips on wafers reduces the costly interconnections between chips. A wafer simply shrinks the distance a signal must travel (over a foot in a present-day system) to a few inches. This can yield tremendous increases in speed.

Another use for these wafer-scale devices is in external memory units—to replace a disk drive, for example. One problem with wafers, though, is the number of flaws, or defects, that can occur on each one during manufacture—as many as 2000 are not uncommon.

The structures on chips are so small that even a speck of dust encountered during the creation process can render one useless. When single chips are being made by the thousands, the few defective ones are simply discarded. Because of the size of wafers, completely eliminating flaws is impossible with present-day technology. *Every* wafer is going to have some flaws. Finding a way to deal with them is the problem.

Clive Sinclair, designer of the Timex/Sinclair and QL computers, has an answer. He is experimenting with wafers that contain logic and processing chips that find flaws and logically lock them out, so only the good areas of the wafer are accessible. This idea is similar to the one used by programs that find bad sectors on hard disks and isolate them from use.

Where Do We Go From Here?

What will the new superchips—whether wafers or super-dense single chips—mean to the average person? The dramatic increase in the power of personal computers we've witnessed in the last few years has been directly tied to the limits of chip design. According to Heilmeier, the new chips will make massively parallel processors available to the average user, with true image comprehension becoming a reality on home computers. Perhaps more importantly, the billion-transistor chips will change the user interface to what Mr. Heilmeier calls the transparent computer—a computer that adapts to the user, rather than the other way around.

—Clifton Karnes

Flat Screens: Computers That Go Anywhere

Not long ago, computers were huge affairs, filling entire rooms and needing round-the-clock air conditioning to cool their power-hungry engines. The invention of the microcomputer chip reduced the size and power requirements of computers. Computers became *personal*. A microcomputer could easily fit on a desk.

Now small computers are turning up on people's laps in airplanes, in courtrooms, and in just about every imaginable situation. Two technologies have made portable computers possible. The first is chip design, which every year produces chips that are smaller, more powerful, and more efficient. The second technology is flat screen, sometimes called flat-panel technology, which allows display screens to be small, light in weight, and energy efficient.

There are three types of flat screens: liquid crystal display (LCD), gas plasma, and electroluminescent (ELD).

LCDs

Almost everyone is familiar with LCD displays. We see them on watches, on calculators, on gas pumps, and recently on small laptop computers. The LCD became the most popular portable display early on because of its low power requirements and low cost. The LCD is so miserly when it comes to power because it uses reflected light rather than emitting light itself. But this is one of its problems, too. To be able to see an LCD, there must be light around for the screen to reflect it. Another drawback of the screen is the fact that the viewing angle is critical: From certain vantage points, the screen is unreadable.

There have been great improvements in LCD technology recently, and two variations are becoming more and more popular—especially in combination. Supertwist and backlit LCDs address many of the garden-variety LCD's negatives and do so with only a moderate penalty in cost and energy usage. The supertwisted backlit LCD is now the most popular design, and its screen is amazingly readable, even in low-light conditions. And LCDs are still the least expensive flat-panel screen with prices between $100 and $200 per display.

Gas Plasma

One flat-screen technology that outshines LCD is gas plasma. Gas plasma gets its name from the construction of the screen. It is neon gas between two panes of glass, and is, in effect, a dot-addressable neon light. Many top-of-the-line laptop makers have moved from LCD to gas plasma because of its outstanding readability. Gas plasma has some drawbacks, however.

Gas plasma screens are heavier than LCD, more fragile, less power efficient, and still much more expensive. The earliest screens were $3,000 each, and, although the price has dropped to the $500–$600 range, they still constitute a relatively power-hungry, high-priced alternative to LCDs. But since they are more readable, and can be much larger, the superior display quality is an important enough consideration for many users to be willing to pay more for it.

Electroluminescent Screens

ELDs have not enjoyed the popularity of either LCDs or gas plasma—yet. They are sturdier than gas plasma and produce a better display than either plasma or LCD, but they are the most expensive type of flat-panel display to produce, with prices between $800 and $1,000. They also require a lot of power, but less than plasma.

Why is the ELD so expensive? The screens of ELDs must be coated with a perfectly smooth, super-thin film. To paraphrase one reviewer, it's like trying to spread a perfect 0.001-inch layer of mayonnaise on a sandwich.

Into The Crystal Ball

The *cathode ray tube* or CRT which made TV possible has served computer users well as a display device. In fact, it's unbeatable in the areas of readability and resolution. Its only drawbacks are its size and power requirements. You can't carry one around easily, nor is it practical to run one on batteries. But the CRT still has one decided advantage over flat screens: color. None of the flat screens have color capability. Flat-screen advocates might argue that text is essentially a black-and-white medium, and most books impart all their information without using color, which leads us to one important potential for flat screens.

Because flat screens are small, light, and can run on batteries, a flat screen equipped with massive amounts of dedicated RAM could be used as a display device for books, magazines, and newspapers.

Instead of getting a newspaper each morning, you could download the information into your flat-panel screen and read it at your leisure. And if you needed to take a few books to work, these books (which would already be in electronic form) also could be transferred to your flat panel. Thus, in one device about the size of a traditional book, you could hold any type of text, and, if technology improves, color graphics.

—Clifton Karnes

Hypertext: Here, There, And Everywhere

Superconducting supermicrocomputers probing super CDs jammed with information. Computers so portable that they become inseparable from the user, always ready to access information the instant it's needed.

What information will these computers be working with? Maybe *all* of it.

The information revolution has accomplished many things, not least of which is the generation of more information. *Lots* more information. We're drowning in the stuff, with new volumes appearing every minute.

How do we sort through these universes of data, shaping their contents to our own needs?

Indexes In The Cards

Essentially, the sorting of information will be accomplished in the same way information has always been organized: via catalogs, concordances, and indexes.

Fortunately, the devices which have done so much to spur the information revolution to such heights are also excellent devices for indexing data. In fact, the traditional index in the back of a book is nothing compared to the types of indexing that computers make possible.

Glimmerings of such macroindexes began to appear near the close of World War II. Vannevar Bush, Director of the Office of Scientific Research and Development during Franklin Roosevelt's administration, oversaw the wartime activities of scientists in a vast number of fields, gaining firsthand experience with the immense explosion of knowledge and information that was even then taking place.

But Bush also saw a path through the tangle of data. In a visionary article entitled "As We May Think," Bush proposed that technology could solve the problems of indexing in ways previously unimaginable. Bush pointed out that traditional indexes were arranged artificially—with topics organized alphanumerically. Humans, though, tend to think in terms of associations and patterns, which may not be alphanumeric at all.

To remedy this conflict, Bush proposed a device he called the *memex*. Based on then-revolutionary microfilm technology, the memex would provide users with rapid access to millions of volumes of data. More importantly, Bush's hypothetical device would allow the user to build trails of associations through that data, in other words, tailoring an index to his or her particular needs.

Navigating Knowledge

Bush's idea caught the imagination of more than a few scholars and experimenters.

At Xerox PARC, Alan Kay and others applied themselves to the concept of a *dynabook*, in essence a dynamic book that configured itself to the user's needs and interests. Lately, Apple's CEO John Sculley has turned old speculations into new marketing, making a concept he calls the *Knowledge Navigator* one of the keys to his vision of the computerized future. Like memex and Dynabook, the Knowledge Navigator is intended to be a tool to help guide individuals interactively through the body of knowledge.

Today, with compact disc technology permitting the storage of greater and greater amounts of data in smaller and more rapidly accessible formats, all of these concepts are growing daily closer to reality. Whether memex, Dynabook, or Knowledge Navigator, the process on which they rest is hypertext.

Here Comes Hypertext

The term itself was coined by author Ted Nelson 20 or so years ago: *hyper* ("over") and *text* ("the body of words"). Together, *hypertext*, they make something else entirely.

Hypertext lets researchers—or just plain readers—look at information nonsequentially. In other words, with a hypertextual index to a body of knowledge, you would be able to proceed with an investigation based upon your own interests and areas of curiosity, rather than simply following the guides provided in a traditional index.

Because hypertext systems treat all of the information they contain as accessible in *any* order, the user is able to dictate the order according to need. One could pursue a subject by way of patterns and associations, combinations of perspective and information, rather than sequentially reading through a body of work in search of the far more specific body of material needed.

The larger the body of information available for hypertextual retrieval, the more complete the research can be. Additionally, side questions or issues that might otherwise go unaddressed can, with a sophisticated and powerful enough hypertext system, be answered easily and quickly.

Desperately Seeking Something

Imagine that from your computer you have access to the Library of Congress, all of its books and materials on fast-access CD ROM (or, more likely, a subsequent and even more powerful generation of storage media.) The whole of the Library has been indexed to form a superindex, a hypertext base through which you will make your way.

Key in the topic you wish to explore: Elvis Presley, say, or medieval economics, or anything you wish. Your screen fills with thousands or hundreds of thousands of references. Too many? You become more specific, narrowing your search, building patterns and associations that will guide you to the exact material you require.

As you make your search, though, new questions arise, side-issues are highlighted that may be worth exploring. Because the information is all there for you, it's a simple matter to widen or narrow the parameters of your research, gathering bits of information here, snippets of insight there, until you have constructed a view of a subject that is all yours.

Not incidentally, along the way to the completion of your project, you have been able to answer the sorts of questions that arise during any research or, for that matter, any reading. If you come across an unfamiliar word, a touch of a key or two is all that's required to obtain

definition, derivation, and any other lexical information desired. Footnotes which cite other, often hard or impossible to obtain sources are now guideposts which, at another keystroke or two, provide you with collateral material.

Nor is text the only material you will have access to. For your report on Elvis, say, you would have access not only to all printed materials about him, but also to his music and films, and the music of those who influenced him.

All of it without leaving your desk.

Tag, You're It

Such a base of knowledge does not yet exist in a hypertextual medium, but we're beginning already to build that base. *Grolier's Encyclopedia* is available on CD ROM, with an index that permits searches for far more patterns of associated material than could be accommodated by traditional indexing or "see also" subheads. Many of the professions are constructing huge data bases that will allow relational or associational searches for patterns of information.

Some see hypertext searches themselves as possessing high informational content. Let's go back to the hypertextual desk.

Suppose every piece of information you used in your project was tagged electronically when you accessed it. The tag might include the project at hand and its goals: a synthesis, say, of all historical data on the subject of hypertext.

After you retrieve each piece of information for your particular purposes, copying the data to a specific project file, your "tag" would remain in the original database to guide you should you need to revisit that particular subject or to contribute its own information to the next researcher who accesses the data.

As computers become more universally linked and able to intercommunicate, everything tagged by one user on one computer will be available to other researchers on other computers. The pathways through knowledge that we follow can, in turn, be followed by others, who may reach far different conclusions than ours.

Hype And Hypertext

There is money to be made from hypertext as well. Developer Neil Larsen, whose *Houdini* is a hypertextual idea processor, suggests that would-be millionaires apply themselves to building hypertext catalogs of the thousands of government rulebooks, codes, and regulations. Such catalogs are worth money to thousands of specialists.

Other commercial possibilities loom. It is not hard to envision scholarly books being advertised for the thoroughness of their indexes, as well as their contents; indeed, *Grolier's* is already emphasizing its index as a selling point. Apple Computer now includes a program titled *HyperCard* with each new Macintosh sold. *HyperCard* provides Mac owners with, among other things, the ability to link information throughout their hard disk, whatever software piece the data resides in. Word processors such as *Nota Bene* are incorporating text-base—essentially a database of words, accessible via hypertext-like searches—into programs. Outline processors including *ThinkTank* and *MaxThink* provide rudimentary hypertext capabilities. Databases and freeform filing systems such as *Notecard* and *3by5* provide users with the freedom to call up stored information in sophisticated nonsequential formats.

Consequences

Gradually at first, but with increasing speed, we will see the commercial potential of hypertext systems spawning conversions of information from ink on paper formats to digital formats.

Such an electronic universal library would permit closer and more immediate contact among researchers in all disciplines, allowing closer and more widespread focus upon topics and problems, reducing the delay time as materials are sought. If hypertextual paths are tagged, then researchers would have the benefit of viewing the paths followed by others, perhaps saving time otherwise wasted in duplicating research. More fancifully, those tagged paths might serve to illuminate "dark areas," paths as yet unexplored, an idea that has itself been explored by the science fiction writer Gordon R. Dickson, among others.

There are risks as well. Will we witness the transition of disciplines away from continuous bodies of knowledge and toward conglomerations of snippets, of patterns and associations rather than continuity and flow?

These articles themselves offer a good example. Of the roughly 6400 words here, several hundred individual words could be used as hypertextual nodes, keywords by which research could be pursued.

Suppose we make a hypertext search of the six elements that make up this entire feature article. A quick search of the total words used shows, as we might expect, that the word *computer* appears a couple of dozen times. That in itself tells us little. But if we use *computer* as a node, and from that node construct links to more specific areas of information, we begin to see different patterns. Four of the usages of *computer* coincide with the word *home,* for example; only a single usage occurs in concert with the word *hypertext.* Yet hypertext, even within these articles, is inextricably linked with the computer.

Would a reader get as much out of these articles reading them in snatches fetched by way of hypertext prompts determined by the reader's particular interests? Probably not. Would a student, preparing a paper on the popular press treatment of topics such as hypertext and home computers benefit from being able instantly to access these articles, and all the others that share those common themes? Possibly.

To be effective, any index must serve as a discriminating guide, an intelligent if not interactive interface between the user and a mass of information. The promise of hypertext is that of true interactivity, yet it carries the risk of being completely without discrimination.

As with any new or emerging technology, hypertext will require us to bring to it our own abilities, our own discrimination and intelligence. With those human tools, and these new technological tools, another level of the information revolution is already being shaped.

—Keith Ferrell

©

SPECTRUM HOLOBYTE PRESENTS...

THE F-16 FIGHTER SIMULATION

With FALCON you fly one of the fastest most maneuverable combat aircraft with a realism you'll find breathtaking. Filled with innovations in sound and graphics, FALCON even lets you dogfight against another player via computer link. This is the fighter simulation you've been waiting for!
Available on IBM and Macintosh.

A COLLECTION OF SOLITAIRE GAMES

SOLITAIRE ROYALE gives a new twist to one of the most popular and timeless card games—solitaire. Play alone or against others in a unique tournament mode that allows competitors the exact same deal of the cards. Choose from ten different card decks. Let the computer do the shuffling, dealing and scoring; never worry again about lost cards.
Available on IBM, Macintosh, Apple //e, //gs, and Tandy.

A STRATEGY MAZE GAME

SOKO-BAN is 50 levels of mazes that test your skills at maneuvering boxes to a designated area. Strategic planning is a must! Increasing levels of difficulty, competition mode, scoring, and option to create and save 49 of your own mazes make this a game you want to play again and again.
Available on IBM, C64, and Apple //e.

AN INTERACTIVE MYSTERY

In INTRIGUE! you find yourself chasing through the streets of D.C. hot on the trail of the villain behind a diabolical scheme involving release of a dangerous virus into the air of our nation's capital. Listen carefully and watch the expressions of the people you meet. With over 2,000 possible solutions, it's a new game every time you play!
Available on Apple // and C64.

SOMETHING FOR EVERYONE

PT-109

Maximize use of your speed, maneuverability, and full weapons arsenal as the enemy attacks from both air and sea in this PT-109 patrol boat simulation.
Available on IBM, Macintosh, and C64.

ORBITER

Experience space flight as you take the controls of this Space Shuttle Simulation from launch to landing. This is an educational and challenging experience.
Available on IBM, Macintosh, and Atari ST.

GATO

Command this WWII Submarine Simulation as you attempt to outsmart the enemy and complete your missions in the South Pacific. It's a classic!
Available on Apple //, IBM, Macintosh, Atari ST, and C64.

LUNAR EXPLORER

Explore the moon in this simulation of the lunar landing vehicle once you have mastered its piloting maneuvers.
Available on Apple // and IBM.

WILDERNESS

Sharpen your survival skills! You encounter harsh weather, rough terrain, and wild animals as you attempt to find a distant ranger outpost in this realistic and entertaining adventure.
Available on Apple // and IBM.

TELLSTAR

View constellations, planets, the sun, moon and stars as you discover the excitement of astronomy.
Available on Apple //, IBM, and Macintosh.

FLEET STREET PUBLISHER

Desktop publishing for the Atari ST is easy yet fully professional with this feature-loaded package.
Available on Atari ST.

BERMUDA PROJECT

This adventure takes you to an isolated pocket of civilization in the heart of that mysterious area known as the Bermuda Triangle.
Available on Macintosh, Atari ST, and Amiga.

THE ART STUDIO

Let your creativity come alive using this paint package to produce everything from business graphics to personal designs.
Available on IBM.

DONDRA—A NEW BEGINNING

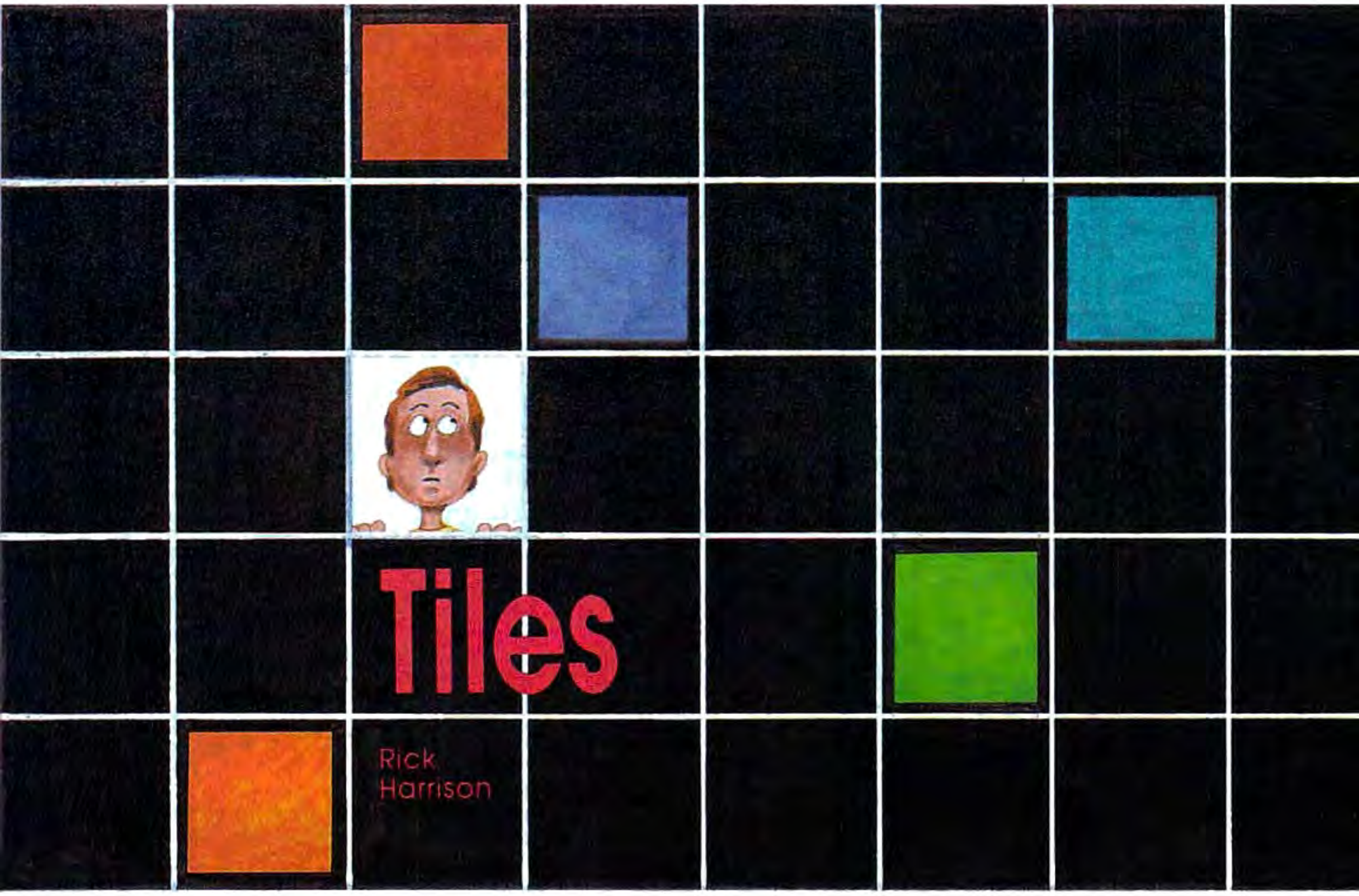
This fantasy quest for the Crystal Prism of Heheutotol is filled with animated sequences and superb graphics that will entertain and enchant you!
Available on Apple //, //gs, IBM and C64.

TETRIS

Fast thinking and quick reflexes are a must to score high on this "puzzle in motion."
Available on IBM.

Spectrum HoloByte™

2061 Challenger Drive Alameda, CA 94501 (415) 522-3584



Tiles

Rick Harrison

See if you can repeat the pattern of colored tiles with this challenging one- or two-player memory game. For the Commodore 64, Apple II, Atari, IBM PC/PCjr, Amiga, and ST. The 64 and Atari versions require a joystick. The IBM PC/PCjr version requires BASICA or GW-BASIC and a color/graphics adapter for the PC and compatibles, and Cartridge BASIC for the PCjr. The Amiga version requires 512K of RAM. The ST version requires GFA BASIC.

"Tiles" is a game of concentration and observation. Putting your memory to the test, Tiles offers amusement for game buffs of all ages. To gain points, players must memorize and reproduce different patterns of colored tiles. The number of tiles increases as the game proceeds, making tile memorization more and more difficult.

As a one-player game, Tiles offers a stimulating challenge. With two players, Tiles becomes a competitive memory test, as each player strives to attain a higher score than his or her opponent.

Playing The Game

When the game begins, a colored tile is randomly placed in a six by ten grid. The current player must memorize the position of the tile in as little time as possible. A player's score decreases by 20 points for approximately every second that the tile is displayed. The score shrinks by 40 points per second if it is between 2000 and 5000, and by 60 points if the score is above 5000. Here, time is not money; it's points.

When you're done viewing the tile pattern, a cursor appears in the grid (the Amiga and ST versions use the mouse pointer instead of a cursor). Moving the cursor, you must locate the position of the tile. If you're right, you gain 100 points, but if you're wrong, you lose 100 points. There is no time limit or order in which hidden tiles must be found.

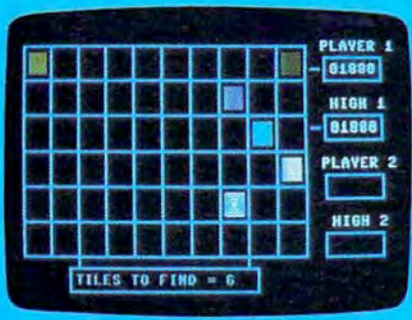
Every round, an extra tile is added to the pattern. The number of tiles that you must find is displayed on the screen. The game is over when your score drops to zero, or if you reach the objective, which is to successfully conquer a pattern of 30 tiles—no small task. A high score is kept for each player.

At the end of the game you are asked if you wish to play again. If you do, high scores are retained and transferred to the next game. Each version of Tiles is slightly different, so be sure to read the instructions for your computer.

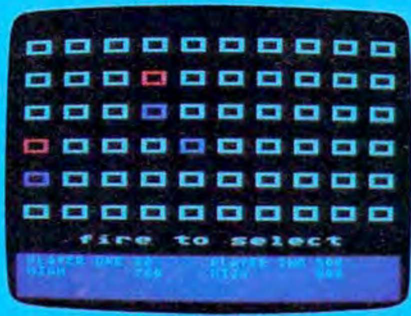
Apple II Version

Type in and save Program 1. If you use a color monitor, do not enter lines 921–926. The Apple II version of Tiles is written in BASIC and runs under DOS 3.3 or ProDOS. To play, simply load and run the program.

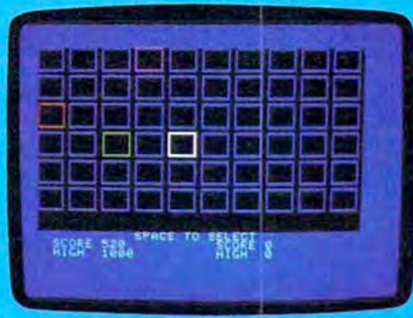
To view the random pattern of tiles, press the space bar. If you are using a black-and-white television or monochrome monitor, colored tiles show up with equal signs (=) in them. When you're done memorizing the tiles, press the space bar again. Once the tiles disappear, the



The Commodore 64 version of "Tiles"—a colorful and challenging memory game.



"Tiles" for Atari 400/800/XL/XE.



The Apple II version of "Tiles."

computer displays a cursor that resembles a large hyphen (—). Use the cursor keys to move the cursor. On the Apple II+, Ctrl-K and Ctrl-J move the cursor up and down, respectively. To select a square in the grid, move the cursor over it and press either Return or the space bar. In the two-player game, the computer prints either *First Player* or *Second Player* to show whose turn it is.

Commodore 64 Version

The Commodore 64 version of Tiles (Program 2) is written in BASIC. Type it in and save it to tape or disk. To play the game, plug a joystick into port 2 and load and run the program.

To view the random pattern of tiles, press the fire button. When you're done memorizing the tiles, press fire again. Once the tiles disappear, the computer displays a large white cursor. Use the joystick to move the cursor. To select a square in the grid, move the cursor over it and press the fire button. In the two-player game, the current player is indicated by the flashing words *Player 1* or *Player 2*.

Atari 400, 800, XL, And XE Version

Tiles for the Atari (Program 3) is written in BASIC. Type it in and save it to tape or disk. To play the game, plug a joystick into port 1 and load and run the program.

To view the random pattern of tiles, press the fire button. When you're finished memorizing the tiles, press fire again. Once the tiles disappear, the computer displays a red underline cursor. Use the joystick to move the cursor. To select a square in the grid, move the cursor over it and press the fire button. The computer prints either *First Player* or *Second Player* to show whose turn it is in the two-player game.

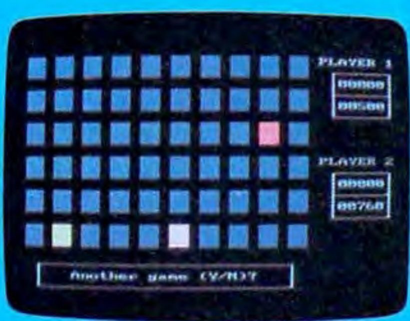
IBM PC/PCjr Version

The IBM PC/PCjr version (Program 4) is written in BASIC. Type it in and save it to disk. On the PC, Tiles requires a color/graphics adapter. To play, simply load and run the program.

To view the random pattern of tiles, press the space bar. When you're finished memorizing the tiles, press the space bar again. Once the tiles disappear, the computer displays a checkerboard cursor. Use the cursor keys to move the cursor. To select a square in the grid, move the cursor over it and press the space bar. In the two-player game, the computer prints either *Player 1* or *Player 2* to show whose turn it is.

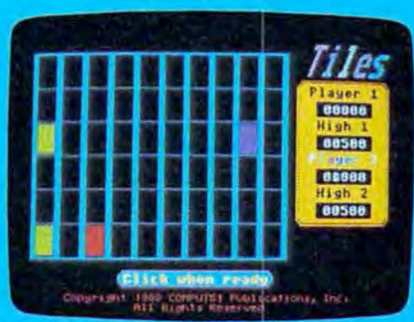
Amiga Version

Tiles for the Amiga (Program 5) is written in Amiga Basic. Type it in and save it to disk. This version



"Tiles" for the IBM PC/PCjr and compatibles.

The Amiga version of "Tiles."



"Tiles" for the Atari ST.

Make Any Computer Do Exactly What You Want With McGraw-Hill's

Contemporary Programming & Software Design Series

Make no mistake. Almost all books and courses on "programming" teach you only the final 5% of the total programming process—namely, how to code in a specific language . . . information of little value if you don't know how to reach the point in the programming process when you are ready to code.

With the Series, however, you'll learn to create your own programs from scratch, even modify off-the-shelf programs. You'll learn enough BASIC and machine language to get you started on the remaining 5% of the programming process.

From Writing Your Own Programs to Modifying Existing Software, Here's the New, Easy, and Low Cost Way to Unlock the Secrets of Your Computer

Whether you use computers for business, for personal applications, or for fun, off-the-shelf programs will never do everything you want them to do for you. That's because they were written by programmers to satisfy what they perceived as the needs of the greatest number of potential users—often missing some or many of your specific needs.

That's why McGraw-Hill's new Contemporary Programming and Software Design Series teaches you how to create your own software . . . either from scratch or by making key modifications to existing programs.

There is nothing magical about it. You learn the process of building a computer program step-by-step with McGraw-Hill *Concept Modules* sent to you one at a time, once a month. Each of the ten modules in the Series takes you through an important step in the development of the structure and detailed logic of a program, including testing, debugging, and documentation.

Unique Interactive Hands-On Instruction

Each module includes an easy-to-understand guide PLUS a 5¼" floppy disk containing typical programs and interactive instruction that you can run on Commodore 64 and 128 computers, IBM PCs and PC compatibles for hands-on experience.

In the first Module, for example, when your sample program (Declining Interest Loans) appears on your screen, you'll find errors on certain program lines. You'll also see that the program is only three-quarters completed.

Now comes the fun part. You'll discover how this program is built, and in the process you'll learn how to identify and correct errors. And by the end of Module 1, you'll actually have completed this program yourself.

But there's more. Special graphics on your screen work in conjunction with the accompanying guide to amplify, illustrate, and deepen your understanding of software design principles.

The Crucial 95%—Learn the Foundation of Computer Programming

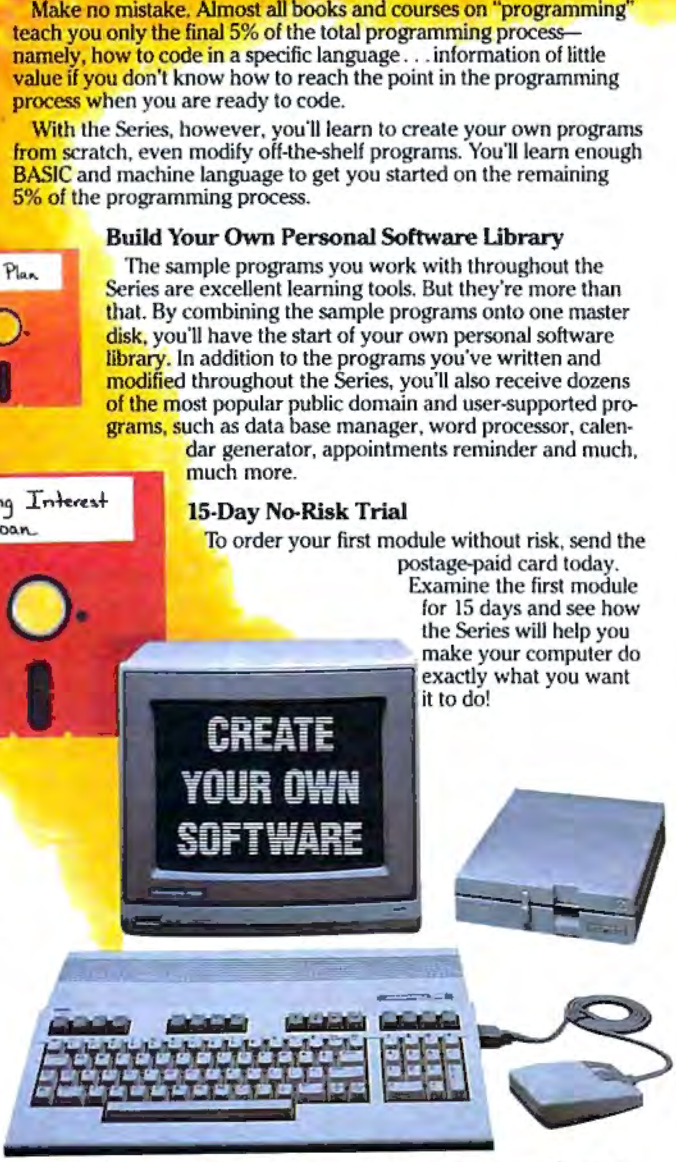
While the Series includes interactive disks that run on specific computers, everything you learn you can apply to any language or machine. Why is this possible? Because McGraw-Hill knows programming is far more than coding a program into the computer using a specific language. In the real world of computers, 95% of the programming process is carried out using design techniques that are independent of specific language or machine. It is this crucial 95% that you thoroughly understand and master in the Series.

Build Your Own Personal Software Library

The sample programs you work with throughout the Series are excellent learning tools. But they're more than that. By combining the sample programs onto one master disk, you'll have the start of your own personal software library. In addition to the programs you've written and modified throughout the Series, you'll also receive dozens of the most popular public domain and user-supported programs, such as data base manager, word processor, calendar generator, appointments reminder and much, much more.

15-Day No-Risk Trial

To order your first module without risk, send the postage-paid card today. Examine the first module for 15 days and see how the Series will help you make your computer do exactly what you want it to do!



If someone has beaten you to the card, write to us for ordering information about the Contemporary Programming and Software Design Series.

McGraw-Hill
Continuing Education Center
3939 Wisconsin Avenue
Washington, DC 20016

must be run on a 512K Amiga. To play the game, simply load and run the program.

The Amiga version uses the mouse. To view the random pattern of tiles, point to the words *Click to view* and press the left mouse button. When you're done memorizing the tiles, click in the same area again. To select a tile, use the mouse to point to the desired tile and press the left mouse button. The current player's score box is highlighted in the two-player version.

ST Version

The Atari ST version (Program 6) is written in *GFA BASIC*. You must own a copy of *GFA BASIC* to type in and run this program. Your ST must be in low-resolution mode. Before typing in Program 6, we recommend that you enter DEFLIST 0 from immediate mode. This mode automatically capitalizes BASIC commands and keywords as you type in the program. This should help insure accurate typing. Once you have entered Program 6, save a copy to disk.

To play the game, load and run the program. The ST version uses the mouse. To view the random pattern of tiles, click either mouse button. When you're finished memorizing the tiles, click again. To select a tile, use the mouse to point to the desired tile and press a mouse button. *Player 1* or *Player 2* flashes to indicate whose turn it is in the two-player game.

For instructions on entering these programs, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

Program 1: Tiles—Apple II Version

```
FA 100 REM COPYRIGHT 1988 COMPUTE! PUBLICATIONS, INC. ALL RIGHTS RESERVED
FC 110 HGR :LX = 28:LY = 24:DX = LX -.5:DY = LY - 4:CC = 6
E6 120 DIM HIGH(1),SC(1),TILE(1),LOSS(1),ROW(30),COL(30),C(30),SEL(59)
4A 130 HIGH(0) = 0:HIGH(1) = - 1:PL = 0
C3 140 HOME : VTAB 21: HTAB 8: PRINT "NUMBER OF PLAYERS (1/2) ?"
AA 150 VTAB 23: PRINT " COPYRIGHT 1988 COMPUTE! PUBLICATIONS": HTAB 11: PRINT "ALL RIGHTS RESERVED";
14 160 CA = INT ( RND (1) * 5) + 1: IF CA > 3 THEN CA = CA + 1
16 170 X = INT ( RND (1) * 10):Y = INT ( RND (1) * 6)
6B 180 X = X * LX:Y = Y * LY
3A 190 GOSUB 910:NUMPL = - 1
3D 200 SC(0) = 500:TILE(0) = - 1 : HCOLOR= CC
92 210 K = PEEK (49152)
8B 220 IF K = 177 THEN NUMPL = 0
4B 230 IF K = 178 THEN NUMPL = 1 :SC(1) = 500:TILE(1) = - 1
D3 240 IF NUMPL < 0 THEN 160
43 250 HOME : VTAB 22: PRINT " SCORE";: HTAB 23: PRINT " SCORE": HTAB 3: PRINT "HIGH  ";HIGH(0);: HTAB 23: PRINT "HIGH  ";HIGH(1);
81 260 PL = 1: GOSUB 970: IF NUMPL = 0 THEN PL = 0
47 270 CA = CC: FOR I = 0 TO 9: FOR J = 0 TO 5
6F 280 X = I * LX:Y = J * LY: GOSUB 910
2E 290 NEXT J: NEXT I
6E 300 PL = NUMPL - PL: GOSUB 970
E1 310 IF SC(PL) < = 0 THEN PL = NUMPL - PL: IF SC(PL) < = 0 THEN 860
D4 320 IF NUMPL = 0 THEN 350
78 330 VTAB 24: HTAB 14: IF PL = 0 THEN PRINT "FIRST PLAYER ";: GOTO 350
31 340 PRINT "SECOND PLAYER";
48 350 TILE(PL) = TILE(PL) + 1:N = TILE(PL): IF N > 29 THEN 860
88 360 FOR I = 0 TO 59:SEL(I) = 0: NEXT I
C8 370 LOSS(PL) = 40: IF SC(PL) > 6000 THEN LOSS(PL) = 60
E1 380 IF SC(PL) < 2000 THEN LOSS(PL) = 20
D8 390 FOR I = 0 TO N
A3 400 ROW(I) = INT ( RND (1) * 6) * LY
93 410 COL(I) = INT ( RND (1) * 10) * LX
2E 420 T = ROW(I) * 10 / LY + COL(I) / LX: IF SEL(T) > 0 THEN 400
C2 430 C(I) = INT ( RND (1) * 4) + 1: IF C(I) = 4 THEN C(I) = 5
CE 440 SEL(T) = I + 1
EA 450 NEXT I
8B 460 VTAB 21: HTAB 13: PRINT " SPACE TO VIEW ": VTAB 1
F4 470 GET A$: IF A$ = " " THEN 490
26 480 GOTO 470
44 490 FOR I = 0 TO N:X = COL(I):Y = ROW(I):CA = C(I): GOSUB 910: NEXT I
4A 500 VTAB 21: HTAB 13: PRINT " SPACE TO SELECT": VTAB 1
C7 510 K = PEEK (49152): IF K = 160 THEN 560
7B 520 SC(PL) = SC(PL) - LOSS(PL): IF SC(PL) < = 0 THEN SC(PL) = 0: GOTO 790
5B 530 GOSUB 970
C8 540 A = PEEK ( - 16336): FOR I = 1 TO 350: NEXT I
9B 550 GOTO 510
DD 560 GET A$
C8 570 CA = CC: FOR I = 0 TO N:X = COL(I):Y = ROW(I): GOSUB 910: NEXT I
4A 580 VTAB 21: HTAB 13: PRINT " TILES NEEDED ";TILE(PL) + 1;" ": VTAB 1
8B 590 CX = 0:CY = 0:HITS = 0
4A 600 HCOLOR= 3: GOSUB 940
68 610 GET K$:K = ASC (K$ + "@")
42 620 HCOLOR= 0: GOSUB 940
9D 630 IF K = 8 THEN CX = CX - 1: IF CX < 0 THEN CX = 9
87 640 IF K = 21 THEN CX = CX + 1: IF CX > 9 THEN CX = 0
C8 650 IF K = 11 THEN CY = CY - 1: IF CY < 0 THEN CY = 5
BD 660 IF K = 10 THEN CY = CY + 1: IF CY > 5 THEN CY = 0
C2 670 IF K = 32 OR K = 13 THEN 690
22 680 GOTO 600
16 690 A = CX + CY * 10
28 700 IF SEL(A) = 0 THEN SC(PL) = SC(PL) - 100: FOR I = 0 TO 5:A = PEEK ( - 16334): NEXT I: GOSUB 970: GOTO 730
83 710 SC(PL) = SC(PL) + 100: PRINT CHR$ (7): GOSUB 970:X = CX * LX:Y = CY * LY:CA = C(SEL(A) - 1):SEL(A) = 0: GOSUB 910
78 720 HITS = HITS + 1
27 730 IF SC(PL) < = 0 THEN SC(PL) = 0: GOTO 790
99 740 IF HITS < = TILE(PL) THEN 600
12 750 FOR I = 0 TO TILE(PL)
8B 760 X = COL(I):Y = ROW(I):CA = CC: GOSUB 910
F1 770 NEXT I
A1 780 GOTO 300
66 790 GOSUB 970
37 800 HCOLOR= 3: GOSUB 1000
CF 810 IF SC(1 - PL) = 0 THEN 860
AE 820 VTAB 21: HTAB 13: PRINT " RETURN TO CONT ";
19 830 GET A$: IF A$ < > CHR$ (13) THEN 830
33 840 HCOLOR= 0: GOSUB 1000
A3 850 GOTO 750
84 860 VTAB 21: HTAB 13: PRINT "   GAME OVER   "
ED 870 VTAB 24: HTAB 14: PRINT " PLAY AGAIN (Y/N)?";
1F 880 GET A$: IF A$ = "Y" OR A$ = "y" THEN HGR : GOTO 140
BD 890 IF A$ = "N" OR A$ = "n" THEN TEXT : END
1E 900 GOTO 880
C2 910 XA = X + DX:YA = Y + DY:YP = Y + 1: HCOLOR= CA
EC 920 HPLOT X,Y TO XA,Y TO XA,YA TO X,YA TO X,YP TO XA - 1,YP TO XA - 1,YA - 1 TO X + 1,YA - 1 TO X + 1,YP
3A 921 REM DELETE LINES 922-926 IF YOU HAVE A COLOR MONITOR
CC 922 IF CA = CC THEN HCOLOR= 0
6C 924 HPLOT X + 7,Y + 14 TO X + 14,Y + 14
86 926 HPLOT X + 7,Y + 6 TO X + 14,Y + 6
1F 930 RETURN
EE 940 XA = CX * LX + 8:YA = CY * LY + 9
4C 950 HPLOT XA,YA TO XA + 7,YA TO XA,YA + 1 TO XA + 7,YA + 1 TO XA,YA + 2 TO XA + 7,YA + 2
25 960 RETURN
15 970 VTAB 22:T = 9 + PL * 20: HTAB T: PRINT "     ";: HTAB T: PRINT SC(PL)
66 980 IF HIGH(PL) < SC(PL) THEN HIGH(PL) = SC(PL): HTAB T: PRINT "     ";: HTAB T: PRINT HIGH(PL);
3F 990 VTAB 1: RETURN
39 1000 FOR I = 0 TO N
85 1010 XA = COL(I) + 8:YA = ROW(I) + 9: GOSUB 950
73 1020 NEXT I
DD 1030 RETURN
```

Program 2: Tiles—Commodore 64 Version

```
FH 5 REM COPYRIGHT 1988 COMPUT
     E! PUBLICATIONS, INC.
     {3 SPACES}ALL RIGHTS RESE
     RVED.
MA 10 PRINT"{CLR}"CHR$(8)TAB(1
      5)"{7 DOWN}{CYN}TILES!"
MR 20 PRINTTAB(5)"{3 DOWN}COPY
      RIGHT 1988 COMPUTE! PUBL
      ."
JX 30 PRINTTAB(10)"{3 DOWN}ALL
      RIGHTS RESERVED"
FG 40 POKE53281,0:POKE53280,0:
      DIMT(60),Z(30),CG(30),M(
      30),CM(30):O$="00000"
PD 50 S=500:S2=500:HS=500:H1=5
      00:T=55337:T(X)=T:V=5324
      8:J=56320:SD=54272
HX 60 POKE2040,13:POKEV+39,1:G
      =1:P=1
PF 70 FORX=SDTOSD+24:POKEX,0:N
      EXT:POKESD+24,15
RB 80 FORX=0TO62:READQ:POKE832
      +X,Q:NEXT
AC 90 DATA,255,255,,128,1,,191
      ,253,,160,5,,175,245,
QX 100 DATA168,21,,171,213,,17
       1,213,,171,213,,171,213
       ,,168
HQ 110 DATA21,,175,245,,160,5,
       ,191,253,,128,1,,255,25
       5
XP 120 DATA,,,,,,,,,,,,,,
MJ 130 FORX=1TO59:T(X)=T(X-1)+
       3:IFA=9THENT(X)=T(X-1)+
       93:A=0:NEXT
SJ 140 A=A+1:NEXT
MM 150 G$="B{RVS}{BLK}
       {2 SPACES}{OFF}{CYN}B
       {RVS}{BLK}{2 SPACES}
       {OFF}{CYN}B{RVS}{BLK}
       {2 SPACES}{OFF}{CYN}B
       {RVS}{BLK}{2 SPACES}
       {OFF}{CYN}B{RVS}{BLK}
       {2 SPACES}{OFF}{CYN}B
       {RVS}{BLK}{2 SPACES}
       {OFF}{CYN}B{RVS}{BLK}
       {2 SPACES}{OFF}{CYN}B
       {RVS}{BLK}{2 SPACES}
       {OFF}{CYN}B{RVS}{BLK}
       {2 SPACES}{OFF}{CYN}B
       {RVS}{BLK}{2 SPACES}
       {OFF}{CYN}B"
ES 160 PRINT"{CLR}":GOSUB1420
ED 170 Y=60:FORX=0TO29:Y=Y-1
FD 180 C=RND(.)*14+1
HX 190 POKET(X),C:POKET(X)+1,C
       :POKET(X)+40,C:POKET(X)
       +41,C
MG 200 POKET(Y),C+1:POKET(Y)+1
       ,C+1:POKET(Y)+40,C+1:PO
       KET(Y)+41,C+1
PS 210 NEXT
PE 220 PRINTTAB(8)"{2 UP}{8}1
       {SPACE}OR 2 PLAYERS?"
FR 230 GOSUB300
RQ 240 IFK$="1"THENPL=1:GOTO290
QE 250 IFK$="2"THENPL=2:H2=500
       :GOTO290
PD 260 GOTO230
GJ 270 PRINT"{HOME}":IFP=2THEN
       FORX=1TO8:PRINT:NEXT:RE
       TURN
MJ 280 RETURN
JP 290 GOSUB1420:GOTO330
SR 300 GETK$:C=RND(.)*14+1:R=R
       ND(.)*60
RG 310 POKET(R),C:POKET(R)+1,C
       :POKET(R)+40,C:POKET(R)
       +41,C:IFK$<>""THENRETURN
KF 320 GOTO300
BF 330 PRINT"{HOME}"TAB(32)"
       {WHT}PLAYER 1":PRINTTAB
       (32)"{UP}{CYN}{A}CCCCC
       {S}":PRINTTAB(31)"CB
       {5 SPACES}B"
JD 340 PRINTTAB(32)"{Z}CCCCC
       {X}"
KX 350 PRINTTAB(33)"{DOWN}
       {WHT}HIGH 1 ":PRINTTAB(
       32)"{UP}{CYN}{A}CCCCC
       {S}":PRINTTAB(31)"CB
       {5 SPACES}B"
AE 360 PRINTTAB(32)"{Z}CCCCC
       {X}"
KS 370 PRINTTAB(32)"{DOWN}
       {WHT}PLAYER 2":PRINTTAB
       (32)"{UP}{CYN}{A}CCCCC
       {S}"
BK 380 PRINTTAB(31)"{BLK}C
       {CYN}B{5 SPACES}B"
EG 390 PRINTTAB(32)"{Z}CCCCC
       {X}"
AK 400 PRINTTAB(33)"{DOWN}
       {WHT}HIGH 2":PRINTTAB(3
       2)"{CYN}{A}CCCCC{S}"
AJ 410 PRINTTAB(31)"{BLK}C
       {CYN}B{5 SPACES}B"
GB 420 PRINTTAB(32)"{Z}CCCCC
       {X}{WHT}"
RK 430 SS=1:HS=H1:GOSUB270:GOS
       UB830:SS=0
GQ 440 IFPL=2THENSS=1:P=2:HS=H
       2:GOSUB270:GOSUB830:SS=
       0:P=1:HS=H1
DC 450 GOSUB1120:TL=1
QD 460 IFPEEK(J)=111THEN460
RE 470 C=INT(14*RND(.)+1)
KA 480 R=INT(60*RND(.)):Z(TL)=
       R
GK 490 IFTL=1THEN520
GG 500 FORX=1TOTL-1:IFR=Z(X)TH
       EN480
JB 510 NEXT
KR 520 M(TL)=T(R):CM(TL)=C:CG(
       TL)=T(R)
MF 530 POKET(R),C:POKET(R)+1,C
       :POKET(R)+40,C:POKET(R)
       +41,C
EQ 540 IFTL=GTHEN560
BR 550 TL=TL+1:GOTO470
EA 560 PRINTTAB(6)"{UP}
       {2 SPACES}FIRE WHEN REA
       DY {WHT}"
PH 570 VW=1:POKESD,14:POKESD+1
       ,67:POKESD+5,0:POKESD+6
       ,240:GOSUB270
QX 580 FORX=1TO150:IFPEEK(J)=1
       11THENVW=0:GOTO640
JH 590 NEXT
MR 600 DS=20:IFS>=2000THENDS=4
       0:IFS>=5000THENDS=60
CS 610 S=S-DS:IFS=0THENVW=0:GO
       TO830
BF 620 POKESD+4,33:POKESD+4,32
JB 630 GOSUB830:GOTO580
KJ 640 GOSUB1100
KE 650 PRINT"{HOME}":FORX=1TO1
       6:PRINT:NEXT:PRINTTAB(6
       )"{8}TILES TO FIND ="G"
       {WHT}"
GB 660 POKEV,24:POKEV+1,58:POK
       EV+21,1
BR 670 IFPEEK(J)=111THEN670
XR 680 IFPEEK(J)=119ANDPEEK(V)
       <>240THENPOKEV,PEEK(V)+
       24:T=T+3
CR 690 IFPEEK(J)=123ANDPEEK(V)
       <>24THENPOKEV,PEEK(V)-2
       4:T=T-3
FH 700 IFPEEK(J)=126ANDPEEK(V+
       1)<>58THENPOKEV+1,PEEK(
       V+1)-24:T=T-120
PA 710 IFPEEK(J)=125ANDPEEK(V+
       1)<>178THENPOKEV+1,PEEK
       (V+1)+24:T=T+120
PX 720 IFPEEK(J)=111THENPOKEV+
       21,0:GOTO740
HE 730 GOTO680
KA 740 GOSUB270:FORX=1TOTL
DA 750 IFT=M(X)THENM(X)=0:S=S+
       100:CK=CK+1:GOTO810
DB 760 NEXT
FG 770 VW=1:POKESD,14:POKESD+1
       ,67:POKESD+5,0:POKESD+6
       ,240
JR 780 FORX=1TO5:S=S-20:IFS=0T
       HENVW=0:GOTO830
JF 790 GOSUB830:POKESD+4,33:PO
       KESD+4,32:NEXT
QB 800 VW=0:POKEV+21,1:GOTO670
CP 810 POKET,CM(X):POKET+1,CM(
       X):POKET+40,CM(X):POKET
       +41,CM(X)
SE 820 GOSUB1020
BC 830 IFP=1THEN860
BK 840 CD=5:S2=S:IFS>H2THENH2=
       S:HS=S
PQ 850 GOTO870
XE 860 CD=4:S1=S:IFS>H1THENH1=
       S:HS=S
KC 870 S$=MID$(STR$(S),2)
XK 880 PRINTTAB(33)LEFT$(O$,5-
       LEN(S$))S$"{UP}":IFVWTH
       ENRETURN
EH 890 IFSSTHEN910
PP 900 IFS<HSTHEN930
JK 910 FORY=1TOCD:PRINT:NEXT:H
       S$=MID$(STR$(HS),2)
SK 920 PRINTTAB(33)LEFT$(O$,5-
       LEN(HS$))HS$"{UP}":IFSS
       THENRETURN
JS 930 IFS=0ANDPL=1THENGOSUB10
       20:GOTO1260
QG 940 IFS=0ANDPL=2THENGOSUB10
       20:GOSUB1220:GOSUB1090:
       GOSUB980:GOTO450
MJ 950 IFCK<>GTHENPOKEV+21,1:G
       OTO670
GQ 960 IFCK=30THENFORX=1TO2000
       :NEXT:PRINT"{CLR}":GOTO
       1340
XX 970 GOSUB980:GOSUB1090:GOTO
       450
AR 980 T=55337:IFPL=1ANDCK<>GT
       HENCK=0:RETURN
QP 990 IFPL=1THENG=G+1:CK=0:RE
       TURN
KH 1000 IFPL=2ANDP=1THENP=2:CK
        =0:S=S2:HS=H2:RETURN
KS 1010 P=1:G=G+1:CK=0:S=S1:HS
        =H1:RETURN
EB 1020 POKESD+5,15:POKESD+6,9
        :POKESD+15,150:POKESD+
        4,21
AQ 1030 IFS>0THENFORE=1TO150ST
        EP5:POKESD+1,E:NEXT:PO
        KESD+4,20:RETURN
XR 1040 FORX=1TOTL
CF 1050 IFCG(X)<>M(X)THEN1080
SH 1060 POKECG(X),CM(X):POKECG
        (X)+1,CM(X):POKECG(X)+
        40,CM(X):POKECG(X)+41,
        CM(X)
DQ 1070 FORE=1TO150STEP5:POKES
        D+1,E:NEXT:POKESD+4,20
GQ 1080 NEXT:RETURN
HE 1090 FORX=1TO2000:NEXT
CB 1100 FORX=1TOTL
DC 1110 POKECG(X),0:POKECG(X)+
        1,0:POKECG(X)+40,0:POK
```

```
          ECG(X)+41,0:NEXT:RETUR
          N
MB 1120 FL=55328:IFP=2THENFL=5
          5728:GOTO1140
QD 1130 POKE56007,0:POKE55807,
          0:POKE55607,3:POKE5540
          7,3:GOTO1150
MG 1140 POKE55407,0:POKE55607,
          0:POKE55807,3:POKE5600
          7,3
FM 1150 PRINT"{HOME}":FORX=1TO
          16:PRINT:NEXT:PRINTTAB
          (6)"{8}{3 SPACES}FIRE
          {SPACE}TO VIEW
          {4 SPACES}"
QM 1160 FC=1
PA 1170 FORX=FLTOFL+7:POKEX,FC
BJ 1180 IFPEEK(J)=111THENFORX=
          FLTOFL+7:POKEX,1:NEXT:
          RETURN
SR 1190 NEXT
MM 1200 IFFC=15THEN1160
QM 1210 FC=FC+1:GOTO1170
JP 1220 PL=1:PRINT"{HOME}":FOR
          X=1TO16:PRINT:NEXT
KE 1230 IFP=2THENP=1:OV=2:G=G+
          1:S=S1:HS=H1:GOTO1250
AG 1240 P=2:OV=1:S=S2:HS=H2
RE 1250 PRINTTAB(6)"{WHT}GAME
          {SPACE}OVER PLAYER"OV:
          RETURN
KX 1260 FORX=1TO3000:NEXT
KA 1270 PRINT"{HOME}":FORX=1TO
          16:PRINT:NEXT:PRINTTAB
          (6)"{WHT}{2 SPACES}PLA
          Y AGAIN Y/N?{2 SPACES}
          "
AR 1280 GOSUB300
RF 1290 IFK$="Y"THENPRINT"
          {CLR}":GOTO1320
GK 1300 IFK$="N"THENPRINT"
          {CLR}":END
KE 1310 GOTO1280
JM 1320 IFH2THENPL=2:S2=500
HF 1330 G=1:CK=0:P=1:S=500:T=5
          5337:GOTO290
JH 1340 PRINTTAB(9)"{5 DOWN}
          {CYN}WELL DONE! PLAYER
          "P
FJ 1350 PRINTTAB(11)"{3 DOWN}3
          0 TILES FOUND!"
CB 1360 PRINTTAB(4)"{3 DOWN}TR
          Y TO IMPROVE YOUR HIGH
           SCORE"
AQ 1370 PRINTTAB(11)"{3 DOWN}P
          LAY AGAIN? Y/N"
QC 1380 GETK$:IFK$=""THEN1380
JM 1390 IFK$="Y"THENPRINT"
          {CLR}":GOTO1320
XD 1400 IFK$="N"THENPRINT"
          {CLR}":END
KR 1410 GOTO1380
QB 1420 PRINT"{HOME}{CYN}{A}CC
          {R}CC{R}CC{R}CC{R}CC
          {R}CC{R}CC{R}CC{R}CC
          {R}CC{S}"
DA 1430 FORX=1TO5:PRINTG$:PRIN
          TG$
HD 1440 PRINT"{Q}CC+CC+CC+CC+C
          C+CC+CC+CC+CC+CC{W}"
DR 1450 NEXT:PRINTG$:PRINTG$
BS 1460 PRINT"{Z}CC{E}CC+CC{E}
          CC{E}CC{E}CC{E}CC{E}CC
          +CC{E}CC{X}"
GQ 1470 PRINTTAB(5)"{A}{E}CCCC
          CCCCCCCCCCCC{E}{S}"
DX 1480 PRINTTAB(5)"B
          {19 SPACES}B"
CP 1490 PRINTTAB(5)"{Z}CCCCCCC
          CCCCCCCCCCCC{X}":RETUR
          N
```

Program 3: Tiles—Atari 400, 800, XL, And XE Version

```
OB 10 REM COPYRIGHT 1988 COM
      PUTE! PUBLICATIONS, IN
      C.  ALL RIGHTS RESERVED
JJ 20 OPEN #1,4,0,"K:"
HO 30 GRAPHICS 1:POKE 756,22
      6:POKE 708,0:POKE 752,
      1
IE 40 ? CHR$(125):? #6;CHR$(
      125)
AN 50 ? "COPYRIGHT 1988 COMP
      UTE! PUBLICATIONS":? "
      {8 SPACES}ALL RIGHTS R
      ESERVED"
PD 60 DIM HIT(59),N(59),ROW(
      30),COL(30),C(30),A$(1
      2),B$(2),SCORE(1),TILE
      (1),HIGH(1),LOSS(1)
ML 70 HIGH(0)=500:HIGH(1)=50
      0:PL=0
AC 80 READ A,B:B$(1)=CHR$(A)
      :B$(2)=CHR$(B):A=1
GA 90 READ B:A$(A)=CHR$(B):A
      =A+1:IF A<13 THEN 90
CF 100 DATA 139,140,17,5,26,
       3,145,133,154,131,177
       ,165,186,163
ML 110 SCORE(0)=500:SCORE(1)
       =500:GOSUB 850:TILE(0
       )=-1:TILE(1)=-1
EI 120 PL=NUMPL-PL:GOSUB 109
       0:POKE 656,2:POKE 657
       ,13
DD 130 IF SCORE(PL)<=0 THEN
       PL=NUMPL-PL:IF SCORE(
       PL)<=0 THEN 700
OE 140 IF NUMPL=0 THEN 170
HL 150 IF PL=0 THEN ? "FIRST
        PLAYER ";:GOTO 170
NO 160 ? "SECOND PLAYER";
MF 170 TILE(PL)=TILE(PL)+1:N
       =TILE(PL):IF N>30 THE
       N 700
JI 180 FOR I=0 TO 59:N(I)=0:
       NEXT I
JP 190 LOSS(PL)=40:IF SCORE(
       PL)>6000 THEN LOSS(PL
       )=60
IE 200 IF SCORE(PL)<2000 THE
       N LOSS(PL)=20
CB 210 FOR I=0 TO N
MC 220 ROW(I)=INT(RND(1)*6)*
       3
ND 230 COL(I)=INT(RND(1)*10)
       *2
JH 240 T=ROW(I)*10/3+COL(I)/
       2:IF N(T)>0 THEN 220
GI 250 C(I)=5:IF RND(1)>0.5
       THEN C(I)=9
KF 260 N(T)=C(I)
CB 270 NEXT I
OE 280 GOSUB 1180
GN 290 POSITION 3,18:? #6;"
       fire to view "
MI 300 IF STRIG(PL)=1 THEN 3
       00
MJ 310 IF STRIG(PL)=0 THEN 3
       10
CD 320 FOR I=0 TO N
PD 330 C=C(I)
PK 340 POSITION COL(I),ROW(I
       ):? #6;A$(C,C+1);:POS
       ITION COL(I),ROW(I)+1
       :? #6;A$(C+2,C+3);
CA 350 NEXT I
OD 360 GOSUB 1180
GD 370 POSITION 3,18:? #6;"f
       ire to select":E=STRI
       G(PL)
KD 380 IF E=0 THEN 440
KJ 390 SCORE(PL)=SCORE(PL)-L
       OSS(PL):IF SCORE(PL)<
       0 THEN SCORE(PL)=0
ED 400 GOSUB 1090:IF SCORE(P
       L)=0 THEN 700
IB 410 SOUND 2,240,10,3:FOR
       I=1 TO 50:IF E=1 THEN
        E=STRIG(PL)
CH 420 NEXT I:SOUND 2,0,0,0
GL 430 GOTO 380
MB 440 GOSUB 1180:HITS=0
OA 450 GOSUB 1060
DE 460 POKE 656,3:POKE 657,1
       3:? "TILES NEEDED ";T
       ILE(PL)+1;
DC 470 X=0:Y=0:POKE 764,255
OG 480 GOSUB 1090
FA 490 POSITION X*2,Y*3+2:?
       #6;B$;:E=STICK(PL):FO
       R I=1 TO 9:NEXT I
IJ 500 POSITION X*2,Y*3+2:?
       #6;"  ";
BA 510 IF E=14 THEN Y=Y-1:IF
        Y<0 THEN Y=5
BA 520 IF E=13 THEN Y=Y+1:IF
        Y>5 THEN Y=0
AP 530 IF E=11 THEN X=X-1:IF
        X<0 THEN X=9
OF 540 IF E=7 THEN X=X+1:IF
       X>9 THEN X=0
NJ 550 IF STRIG(PL)=1 THEN 4
       90
JD 560 T=X+Y*10
FD 570 IF N(T)>0 THEN 620
LF 580 SOUND 2,50,12,12:FOR
       I=1 TO 10:NEXT I:SOUN
       D 2,0,0,0:SCORE(PL)=S
       CORE(PL)-100
CH 590 IF SCORE(PL)<=0 THEN
       SCORE(PL)=0:GOSUB 109
       0:GOSUB 780:GOTO 120
MN 600 IF STRIG(PL)=0 THEN 6
       00
GM 610 GOTO 480
AJ 620 FOR I=50 TO 200 STEP
       20:SOUND 2,I,10,I/15:
       NEXT I:SOUND 2,0,0,0
OJ 630 POSITION X*2,Y*3:A=N(
       T):N(T)=0
HA 640 ? #6;A$(A,A+1);:POSIT
       ION X*2,Y*3+1:? #6;A$
       (A+2,A+3);
GG 650 SCORE(PL)=SCORE(PL)+1
       00
AF 660 HITS=HITS+1:IF HITS<=
       TILE(PL) THEN 480
GE 670 GOSUB 1090:GOSUB 1060
       :POKE 656,3:POKE 657,
       13:? "{18 SPACES}";
NN 680 IF STRIG(PL)=0 THEN 6
       80
GL 690 GOTO 120
KE 700 POSITION 2,18:? #6;"
       {3 SPACES}game over
       {4 SPACES}";:C=5
GE 710 POKE 764,255:POKE 656
       ,3:POKE 657,3:? "DO Y
       OU WANT TO PLAY AGAIN
        (Y/N)?";
DC 720 E=PEEK(764)
BL 730 IF E<>43 THEN 760
JE 740 POKE 657,3:? "
       {33 SPACES}";:POKE 656
       ,2:POKE 657,2
GJ 750 POSITION 2,18:? #6;"
       {17 SPACES}";:GOTO 110
BL 760 IF E<>35 THEN 720
DA 770 GRAPHICS 0:END
CN 780 FOR I=0 TO N
NM 790 POSITION COL(I),ROW(I
       )+1:? #6;B$:NEXT I
AE 800 IF NUMPL=0 OR SCORE(1
       -PL)=0 THEN RETURN
MK 810 POSITION 3,18:? #6;"s
```

COMPUTER DIRECT

Is 1st In PRICE, SUPPORT, & GUARANTEE!

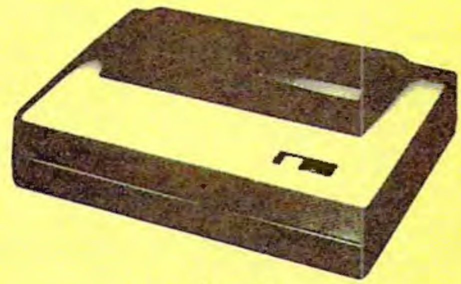
WE LOVE OUR CUSTOMERS! Prices Expire 2-29-88

Genuine IBM® 8½" Printer

8½" Letter Size • 80 Column • Limited Quantities

Sale $49.95

(Add $7.50 shipping.*) List $199



No One Sells This Printer For Less!

- Unbelievable Low Price
- Advanced Dot Matrix - Heat Transfer
- Upper & Lower Case (With True Lower Descenders)
- Underline • Enlarged
- Graphics With Commodore, Apple & Atari Interfaces
- Ready to Hook Up To Serial Port Of IBM® PCjr
- Low Cost Adapters For IBM®, Apple, Commodore, Laser, & Atari Computers

SAVE

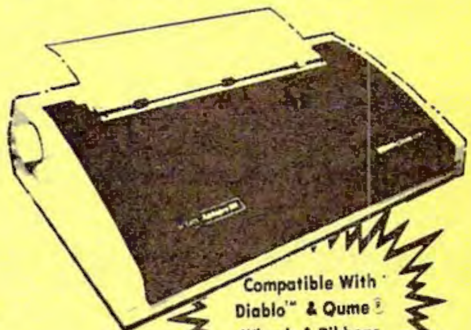
1st Come 1st Serve Basis

22 CPS Daisy Wheel Printer

13" Daisy Wheel Printer With True Letter Quality

Sale $99.95

(Add $10.00 shipping.*) List $499



No One Sells This Printer For Less!

Compatible With Diablo™ & Qume® Wheels & Ribbons

- Daisy Wheel Printing With True Letter Quality
- 22 CPS, AAA Text
- Below Wholesale Cost
- 13" Wide Carriage
- Both Daisy Print Wheel & Ribbons Compatible with Diablo™ & Qume®
- Single Sheet or Continuous Feed

Typewriter Quality

Sale

15 Day Free Trial • 90 Day Immediate Replacement Policy

Call **(312) 382-5050**

We Love Our Customers!

Mail

COMPUTER DIRECT
22292 N. Pepper Road
Barrington, IL. 60010

```
        pace to cont"
PF 820  GET #1,E:IF E<>32 THE
        N 820
DC 830  GOSUB 1060
HM 840  RETURN
JC 850  POKE 764,255:? "
        {5 SPACES}NUMBER OF P
        LAYERS (1/2)?"
LE 860  NUMPL=-1:C=5
DI 870  E=PEEK(764)
DN 880  IF E=30 THEN NUMPL=1
DO 890  IF E=31 THEN NUMPL=Ø
NL 900  GOSUB 1010
OP 910  IF NUMPL<Ø THEN 870
JB 920  POKE 656,Ø:POKE 657,Ø
        :? CHR$(125)
BB 930  FOR I=Ø TO 5
BH 940  FOR J=Ø TO 9
BI 950  POSITION J*2,I*3:? #6
        ;A$(1,2);
HJ 960  POSITION J*2,I*3+1:?
        #6;A$(3,4);
OL 970  NEXT J:NEXT I
HP 980  FOR PL=Ø TO 1:GOSUB 1
        Ø9Ø:GOSUB 115Ø:NEXT P
        L
KG 990  PL=1:IF NUMPL=Ø THEN
        PL=Ø
KB 1000 RETURN
KP 1010 C=C+4:IF C>=13 THEN
        C=1
DJ 1020 I=INT(RND(1)*1Ø):J=I
        NT(RND(1)*6)
IC 1030 POSITION I*2,J*3:? #
        6;A$(C,C+1)
DO 1040 POSITION I*2,J*3+1:?
        #6;A$(C+2,C+3)
KG 1050 RETURN
FF 1060 FOR I=Ø TO N
NC 1070 POSITION COL(I),ROW(
        I):? #6;A$(1,2);:POS
        ITION COL(I),ROW(I)+
        1:? #6;A$(3,4);
GL 1080 NEXT I:RETURN
FG 1090 IF PL=1 THEN 112Ø
LE 1100 POKE 656,Ø:POKE 657,
        1:? "PLAYER ONE
        {7 SPACES}";:POKE 65
        7,12:? SCORE(PL);
NB 1110 GOTO 113Ø
AA 1120 POKE 656,Ø:POKE 657,
        2Ø:? "PLAYER TWO
        {7 SPACES}";:POKE 65
        7,31:? SCORE(PL);
BF 1130 IF SCORE(PL)<HIGH(PL
        ) THEN RETURN
HJ 1140 HIGH(PL)=SCORE(PL)
HO 1150 T=1:IF PL=1 THEN T=2
        Ø
HB 1160 POKE 656,1:POKE 657,
        T:? "HIGH{14 SPACES}"
        ;:POKE 657,T+11:? HI
        GH(PL);
KJ 1170 RETURN
GP 1180 POSITION Ø,18:? #6;"
        {21 SPACES}":RETURN
```

Program 4: Tiles—IBM PC/PCjr Version

```
BO 10 'TILES
PD 20 'COPYRIGHT 1988
DF 30 'COMPUTE! PUBLICATIONS, IN
      C.
AC 40 'ALL RIGHTS RESERVED.
ED 50 CLS:PRINT"COPYRIGHT 1988 C
      OMPUTE! PUBL.":PRINT"   AL
      L RIGHTS RESERVED."
OE 60 FOR TT=1 TO 3ØØØ:NEXT TT
BC 70 KEY OFF:DEF SEG=Ø:DEFINT A
      -Z:POKE 1047,PEEK(1047) OR
       64:RANDOMIZE TIMER
FH 80 DIM HIT(59),NUMS(59),HROW(
      3Ø),HCOL(3Ø)
LF 90 WIDTH 4Ø:SCREEN Ø,1:COLOR
      8,Ø,Ø:CLS:GOSUB 39Ø:GOSUB
      42Ø
CC 100 PL=NUMPL-PL:IF SCORE(PL)=
       Ø THEN 1ØØ
OE 110 IF NUMPL=1 THEN LOCATE 3+
       9*PL,31:COLOR 23:PRINT"PL
       AYER"PL+1:COLOR 7:LOCATE
       3+9*(1-PL),31:PRINT"PLAYE
       R"2-PL
DE 120 LOCATE 22,3:PRINT SPC(6)"
       Space to view"SPC(6):WHIL
       E INKEY$<>CHR$(32):WEND
GB 130 GOSUB 61Ø:LOCATE 22,3:PRI
       NT SPC(5)"Space when read
       y":TTIME!=ABS(TIMER)
LB 140 DEC(PL)=4Ø:IF SCORE(PL)<2
       ØØØ THEN DEC(PL)=2Ø ELSE
       IF SCORE(PL)>5ØØØ THEN DE
       C(PL)=6Ø
PC 150 IF ABS(TIMER)>TTIME!+1 TH
       EN SCORE(PL)=SCORE(PL)-DE
       C(PL):SOUND 9ØØØ,.5:TTIME
       !=ABS(TIMER):IF SCORE(PL)
       <=Ø THEN SCORE(PL)=Ø:GOSU
       B 23Ø:GOTO 21Ø
AB 160 GOSUB 23Ø:IF INKEY$<>CHR$
       (32) AND SCORE(PL)>Ø THEN
        14Ø
AA 170 GOSUB 62Ø:LOCATE 22,5:PRI
       NT SPC(6)"Tiles ="HITNUM(
       PL)+1;SPC(4)
BJ 180 GOSUB 51Ø:IF SCORE(PL)>Ø
       THEN HNUM(PL)=HNUM(PL)+HI
       TNUM(PL):HITNUM(PL)=HITNU
       M(PL)+1 ELSE GOSUB 36Ø:SO
       UND 4Ø,5:IF SCORE(NUMPL-P
       L)=Ø THEN 21Ø ELSE LOCATE
        22,5:PRINT SPC(2)"Space
       to continue"SPC(2):WHILE
       INKEY$<>CHR$(32):WEND
HJ 190 IF SCORE(PL)>HIGH(PL) THE
       N P=7:GOSUB 23Ø:P=5
DI 200 GOSUB 62Ø:IF HITNUM(PL)<3
       Ø THEN 1ØØ
JG 210 LOCATE 22,3:PRINT SPC(3)"
       Another game (Y/N)?"SPC(3
       )
EF 220 K$="":WHILE K$<>"Y" AND K
       $<>"N":K$=INKEY$:WEND:IF
       K$="N" THEN CLS:END ELSE
       9Ø
PC 230 T$=STR$(SCORE(PL)):LOCATE
       P+9*PL,33:PRINT RIGHT$("
       ØØØØ"+RIGHT$(T$,LEN(T$)-1
       ),5):RETURN
IH 240 LOCATE ROW,32:PRINT CHR$(
       218)STRING$(5,196)CHR$(19
       1);
OP 250 LOCATE ROW+1,32:PRINT CHR
       $(179)"ØØ5ØØ"CHR$(179);
GM 260 LOCATE ROW+2,32:PRINT CHR
       $(195)STRING$(5,196)CHR$(
       18Ø)
CD 270 LOCATE ROW+3,32:PRINT CHR
       $(179)"ØØ5ØØ"CHR$(179);
KD 280 LOCATE ROW+4,32:PRINT CHR
       $(192)STRING$(5,196)CHR$(
       217);:RETURN
IL 290 LOCATE ROW*3+3,COL*3+1:CO
       LOR CI(COL):PRINT B$P$B$:
       COLOR 7:RETURN
NM 300 TILE=ROW*1Ø+COL:TROW=ROW*
       3+1:TCOL=COL*3+1:J=INT(HI
       T(PTILE)/2):LOCATE PROW+2
       ,PCOL:COLOR 8*(1-J)+CI(IN
       T((PCOL-1)/3))*J:PRINT B$
       P$B$
IK 310 J=INT(HIT(TILE)/2):LOCATE
        TROW+2,TCOL:COLOR 7,CI(C
       OL)*J:PRINT C$P$C$:PTILE=
       TILE:PROW=TROW:PCOL=TCOL:
       COLOR 7,Ø:RETURN
NC 320 FOR I=Ø TO 59:HIT(I)=Ø:NE
       XT
FP 330 FOR I=Ø TO 1Ø:R=INT(RND*1
       1):T=CI(R):CI(R)=CI(I):CI
       (I)=T:NEXT
DP 340 FOR I=Ø TO HITNUM(PL):R=I
       NT(RND*6Ø):T=NUMS(R):NUMS
       (R)=NUMS(I):NUMS(I)=T:NEX
       T
OG 350 FOR I=Ø TO HITNUM(PL):HIT
       (NUMS(I))=1:HROW(I)=INT(N
       UMS(I)/1Ø):HCOL(I)=NUMS(I
       ) MOD 1Ø:NEXT:RETURN
EL 360 FOR I=Ø TO HITNUM(PL):TIL
       E=NUMS(I)
GK 370 IF HIT(TILE)=1 THEN LOCAT
       E HROW(I)*3+3,HCOL(I)*3+1
       :COLOR CI(HCOL(I))+16:PRI
       NT B$P$B$
KP 380 NEXT:COLOR 7:RETURN
FD 390 B$=STRING$(2,219):C$=STRI
       NG$(2,176):P$=STRING$(2,2
       9)+CHR$(31):FOR I=Ø TO 59
       :NUMS(I)=I:NEXT:P=5
PH 400 FOR I=Ø TO 1:HITNUM(I)=Ø:
       HNUM(I)=Ø:SCORE(I)=5ØØ:HI
       GH(I)=5ØØ:NEXT:FOR I=Ø TO
        5:CI(I+5)=I+9:CI(I)=I+1:
       NEXT
MA 410 RETURN
NC 420 CLS:LOCATE 3,1,Ø:FOR I=Ø
       TO 5:FOR J=Ø TO 1:FOR K=Ø
        TO 9:PRINT B$CHR$(32);:N
       EXT:PRINT:NEXT:PRINT:NEXT
       :COLOR 7
KH 430 LOCATE 21,2:PRINT CHR$(21
       8)STRING$(25,196)CHR$(191
       )
MD 440 LOCATE 22,2:PRINT CHR$(17
       9)SPC(25)CHR$(179)
LN 450 LOCATE 23,2:PRINT CHR$(19
       2)STRING$(25,196)CHR$(217
       )
LC 460 LOCATE 22,3:PRINT"Number
       of players (1/2)?"
MD 470 K$="":WHILE K$<>"1" AND K
       $<>"2":K$=INKEY$:WEND:NUM
       PL=ASC(K$)-49
IC 480 LOCATE 3,31:PRINT"PLAYER
       1":ROW=4:GOSUB 24Ø
FH 490 IF NUMPL=1 THEN LOCATE 12
       ,31:PRINT"PLAYER 2":ROW=1
       3:GOSUB 24Ø
AD 500 PL=NUMPL:RETURN
PJ 510 ROW=Ø:COL=Ø:PROW=1:PCOL=1
       :GOSUB 3ØØ:HITS=Ø
PK 520 K$=RIGHT$(INKEY$,1):IF K$
       ="" THEN 52Ø ELSE K=ASC(K
       $)
FK 530 IF K<>32 THEN 56Ø
HA 540 IF HIT(TILE)=1 THEN HIT(T
       ILE)=2:GOSUB 29Ø:SCORE(PL
       )=SCORE(PL)+1ØØ:HITS=HITS
       +1 ELSE SOUND 9ØØØ,.5:SCO
       RE(PL)=SCORE(PL)-1ØØ:IF S
       CORE(PL)<=Ø THEN SCORE(PL
       )=Ø:COLOR 8:LOCATE PROW+2
       ,PCOL:PRINT B$P$B$:COLOR
       7
CA 550 GOSUB 23Ø:IF HITS>HITNUM(
       PL) OR SCORE(PL)=Ø THEN R
       ETURN
GI 560 IF K=72 THEN IF ROW>Ø THE
       N ROW=ROW-1
FN 570 IF K=8Ø THEN IF ROW<5 THE
       N ROW=ROW+1
DB 580 IF K=75 THEN IF COL>Ø THE
       N COL=COL-1
LM 590 IF K=77 THEN IF COL<9 THE
       N COL=COL+1
HI 600 GOSUB 3ØØ:GOTO 52Ø
CG 610 GOSUB 32Ø:FOR I=Ø TO HITN
       UM(PL):ROW=HROW(I):COL=HC
       OL(I):GOSUB 29Ø:NEXT:RETU
       RN
BJ 620 FOR I=Ø TO HITNUM(PL):LOC
       ATE HROW(I)*3+3,HCOL(I)*3
       +1:COLOR 8:PRINT B$P$B$:C
       OLOR 7:NEXT:RETURN
```

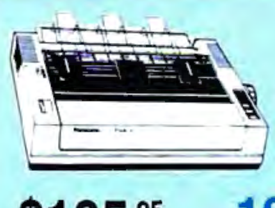
Program 5: Tiles—Amiga Version

```
'Tiles ◄
'Copyright 1988 ◄
'COMPUTE! Publications, Inc.◄
'All Rights Reserved.◄
◄
CLEAR ,25000:DEFINT a-z:RANDOMIZE TIMER◄
DIM hit(59),nums(59),hrow(30),hcoL(30),pat(1)◄
SCREEN 1,320,200,4,1:WINDOW 3,,(0,0)-(311,186),1
6,1:WINDOW OUTPUT 3◄
PRINT "Copyright 1988":PRINT"COMPUTE! Publicatio
ns, Inc.":PRINT"All Rights Reserved"◄
FOR tt=1 TO 3000:NEXT tt◄
RESTORE Colors:FOR i=0 TO 15:READ r,g,b:PALETTE
i,r/10,g/10,b/10:NEXT◄
Colors:◄
DATA 1,0,3,6,6,6,2,2,6,7,7,0◄
DATA 5,0,0,8,0,0,0,8,0,4,4,4◄
DATA 8,8,0,8,0,8,0,8,8,5,0,0◄
DATA 0,5,0,5,5,0,5,0,5,0,5,5◄
FOR i=0 TO 59:nums(i)=i:NEXT:ON TIMER(1) GOSUB D
ecScore◄
FOR i=0 TO 10:ci(i)=i+5:NEXT:pat(1)=&HFFFF◄
NewGame:◄
FOR i=0 TO 1:hitnum(i)=0:hnum(i)=0:score(i)=500:
high(i)=500:NEXT:GOSUB Board◄
Start:◄
pL=numpL-pL:IF score(pL)=0 THEN Start◄
IF numpL=1 THEN◄
i=1-pL◄
LINE (247,31+64*i)-(311,72+64*i),0,b◄
LINE (249,33+64*i)-(309,70+64*i),0,b◄
LINE (247,31+64*pL)-(311,72+64*pL),7,b◄
LINE (249,33+64*pL)-(309,70+64*pL),7,b◄
END IF ◄
LINE(17,173)-(227,185),4,bf:COLOR 3,4◄
LOCATE 23,4:PRINT SPACE$(5)"Click to view"SPACE$
(6):k=0◄
WHILE k<>3 AND k<>4:WHILE MOUSE(0)<1:WEND:k=POIN
T(MOUSE(1),MOUSE(2)):WEND◄
GOSUB ShowTiLes:LINE(17,173)-(227,185),4,bf◄
LOCATE 23,8:PRINT "Click when ready":COLOR 3,0:T
IMER ON:k=0◄
WHILE k<>3 AND k<>4 AND score(pL)>0◄
IF MOUSE(0)>0 THEN k=POINT(MOUSE(1),MOUSE(2))◄
WEND◄
TIMER OFF:LINE(17,173)-(227,185),0,bf◄
GOSUB HideTiLes:LOCATE 23,8:PRINT SPACE$(4)"Tile
s ="hitnum(pL)+1 SPACE$(3)◄
IF score(pL)>0 THEN GOSUB DoSearch◄
IF score(pL)>0 THEN◄
hnum(pL)=hnum(pL)+hitnum(pL):hitnum(pL)=hitnum(p
L)+1◄
ELSE◄
GOSUB UncoverTiLes◄
IF score(numpL-pL)=0 THEN◄
GOTO Endgame◄
ELSE◄
LINE(17,173)-(227,185),4,bf:COLOR 3,4◄
LOCATE 23,4:PRINT SPACE$(4)"Click to continue"SP
ACE$(3):k=0◄
WHILE k<>3 AND k<>4:WHILE MOUSE(0)<1:WEND:k=POIN
T(MOUSE(1),MOUSE(2)):WEND◄
END IF ◄
END IF◄
IF score(pL)>high(pL) THEN high(pL)=score(pL):p=
1:GOSUB PrintScore:p=0◄
GOSUB HideTiLes:IF hitnum(pL)=30 THEN Endgame EL
SE Start◄
◄
DecScore:◄
dec(pL)=40◄
IF score(pL)<2000 THEN dec(pL)=20 ELSE IF score>
5000 THEN dec(pL)=60◄
score(pL)=score(pL)-dec(pL):SOUND 9000,.5◄
IF score(pL)<=0 THEN score(pL)=0◄
◄
PrintScore:◄
t$=STR$(score(pL)):LOCATE pL*8+p*2+6,34◄
PRINT RIGHT$(SPACE$(4)+RIGHT$(t$,LEN(t$)-1),5):R
ETURN◄
◄
ShowTiLe:◄
x=coL*24+6:y=row*24+16:LINE(x,y)-(x+16,y+16),ci(
coL),bf:RETURN◄
◄
Square:◄
x=coL*24+6:y=row*24+16:LINE(x,y)-(x+16,y+16),2,b
f:RETURN◄
◄
Board:◄
CLS:LINE(0,10)-(245,159),1,b◄
FOR row=0 TO 5:FOR coL=0 TO 9:GOSUB Square:NEXT
coL,row◄
LINE(16,172)-(228,186),1,b◄
LOCATE 23,4:COLOR 3:PRINT"Number of players (1/2
)?"◄
k$="":WHILE k$<>"1" AND k$<>"2":k$=INKEY$:WEND:n
umpL=ASC(k$)-49◄
LOCATE 4,32:PRINT"PLAYER 1":LINE (248,32)-(310,7
1),1,b◄
IF numpL=1 THEN LOCATE 12,32:PRINT"PLAYER 2":LIN
E (248,96)-(310,135),1,b◄
FOR pL=0 TO numpL:p=0:GOSUB PrintScore:p=1:GOSUB
 PrintScore:NEXT◄
pL=numpL:p=0◄
RETURN◄
◄
DoSearch:◄
hits=0◄
GetMouse:WHILE MOUSE(0)<1:WEND◄
x=MOUSE(3):y=MOUSE(4):IF POINT(x,y)<>2 THEN GetM
ouse◄
row=INT((y-16)/24):coL=INT((x-6)/24):tiLe=row*10
+coL◄
IF hit(tiLe)=1 THEN◄
GOSUB ShowTiLe:score(pL)=score(pL)+100:hits=hits
+1:hit(tiLe)=2◄
ELSE◄
SOUND 9000,.5:score(pL)=score(pL)-100:IF score(p
L)<=0 THEN score(pL)=0◄
END IF◄
GOSUB PrintScore:IF hits>hitnum(pL) OR score(pL)
=0 THEN RETURN ELSE GetMouse◄
◄
ShowTiLes:◄
FOR i=0 TO 59:hit(i)=0:NEXT◄
FOR i=0 TO 10:r=INT(RND*11):t=ci(r):ci(r)=ci(i):
ci(i)=t:NEXT◄
FOR i=0 TO hitnum(pL):r=INT(RND*60):t=nums(r):nu
ms(r)=nums(i):nums(i)=t:NEXT◄
FOR i=0 TO hitnum(pL):hit(nums(i))=1:hrow(i)=INT
(nums(i)/10)◄
hcoL(i)=nums(i) MOD 10:row=hrow(i):coL=hcoL(i):G
OSUB ShowTiLe:NEXT◄
RETURN◄
◄
UncoverTiLes:◄
FOR i=0 TO hitnum(pL):tiLe=nums(i)◄
IF hit(tiLe)=1 THEN◄
pat(0)=&HAAAA:pat(1)=&H5555:PATTERN ,pat◄
PAINT (hcoL(i)*24+6,hrow(i)*24+16),ci(hcoL(i)),0
◄
pat(0)=&HFFFF:pat(1)=&HFFFF:PATTERN ,pat◄
END IF◄
NEXT:RETURN◄
◄
HideTiLes:◄
FOR i=0 TO hitnum(pL):row=hrow(i):coL=hcoL(i):GO
SUB Square:NEXT:RETURN◄
◄
Endgame:◄
LOCATE 23,7:PRINT"Play Again (Y/N)?"◄
k$="":WHILE k$<>"Y" AND k$<>"N":k$=UCASE$(INKEY$
):WEND◄
IF k$="Y" THEN GOTO NewGame ELSE WINDOW CLOSE 3:
SCREEN CLOSE 1:END◄
◄
```

Price Guarantee

Since 1981, we have led the industry by continuing to offer the lowest national prices while providing quality service. Many companies have come and gone trying to imitate our quality and service. If by some oversight we do not have the lowest prices advertised on the products you desire, then we would appreciate the opportunity to rectify this oversight.

New 1091i Model II



NEW!

Panasonic 1091i Model II

$185.95

Monitors

Thomson:

230 Amber TTL/12"	$85
450 Amber TTL/15"	$129.95
4120 CGA	$199.95
4160 CGA	$259.95
4460 EGA	$319.95
4375 UltraScan	$459.95
GB 100 EGA Card	$129.95
GB 200 Super Card	$219.95

Blue Chip:

BCM 12" Green TTL	$75
BCM 12" Amber TTL	$79
BCM 14" Color	$239

NEC:

Multisync II	$559

Save $100 over NEC Multisync with Thomson 4375 UltraScan $459.95

THOMSON 4120 Monitor

- 14" RGBI/video composite/analog
- Compatible with IBM and Commodore
- RGB data cable included

$199.95

Modems

Avatex:

1200e	$75.00
1200i PC Card	$75.00
1200hc Modem	$95.00
2400	$179.95
2400i PC Card	$169.95

Hayes:

Smartmodem 300	$149.95
Smartmodem 1200	$285.95
Smartmodem 2400	$425.95

Special Purchase

star micronics inc

NP-10

$134.95

- 100 cps draft
- 25 NLQ
- EZ Front Panel Selection
- Auto Single Sheet Feed
- Adjustable Tractor Feed

EXCELLENT VALUE

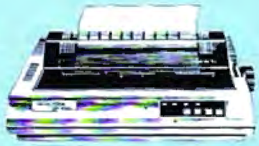
w/purchase of 2 ribbons

Great Performance Great Price

SEIKOSHA SP 180Ai

- 100 cps draft
- 20 cps NLQ
- direct connect for Commodore

$129.95



PRINTERS

SEIKOSHA

SP 180Ai	$129.95
SP 180VC	$129.95
SP 1000VC	$139.95
SP 1200VC	$155.95
SP 1200Ai	$165.95
SP 1200AS RS232	$165.95
SL 80Ai	$299.95
MP1300Ai	$269.95
MP5300Ai	$375.95
MP5420Ai	$879.95
SP Series Ribbon	$7.95
SK3000 Ai	$349.95
SK3005 Ai	$429.95
800 CPS	$CALL!

BROTHER

M1109	$195
M1409	$299
M1509	$365
M1709	$475
Twinwriter 6 Dot & Daisy	$899
M1724L	$599
HR20	$339
HR40	$569
HR60	$709.95

star micronics inc

NP-10	$134.95
NX-10	$149.95
NX-10C w/interface	$165.95
NL-10 w/o Cart	$169.95
NX-1000	$165.95
NX-1000C	$175.95
NX-1000 Color	$225.95
NX-1000C Color	$229.95
NX-15	$295.95
ND-10	$265.95
ND-15	$379.95
NR-15	$425.95
NB-15 24 Pin	$699.95
NB24-10 24 Pin	$425.95
NB24-15 24 Pin	$559.95

CITIZEN

120 D	$149.95
180 D	$169.95
MSP-10	$259.95
MSP-40	$289.95
MSP-15	$324.95
MSP-50	$389.95
MSP-45	$425.95
MSP-55	$489.95
Premiere 35	$464.95
Tribute 224	$624.95

EPSON

LX800	$175.95
FX86E	$289.95
FX286E	$429.95
EX800	$374.95
EX1000	$509.95
LQ500	$CALL
LQ800	$369.95
LQ1000	$549.95
LQ2500	$849.95
GQ3500	SLOW
LQ850	$499.95
LQ1050	$679.95

OKIDATA

Okimate 20	$119
Okimate 20 w/cart	$179.95
120	$189.95
180	$219.95
182	$199.95
192+	$309.95
193+	$449.95
292 w/interface	$449.95
293 w/interface	$585.95
294 w/interface	$819.95
393	$955.95

Panasonic

1080i Model II	$165.95
1091i Model II	$185.95
1092i	$295.95
1592	$399.95
1595	$449.95
3131	$269.95
3151	$419.95
KXP 4450 Laser	$CALL
1524 24 Pin	$559.95

Toshiba

321SL	$489
341 SL	$659
P351 Model II	$899

DIABLO

D25	$499.95
635	$779.95

Join the thousands who shop Lyco and Save!

We stock interfacing for Atari, Commodore, Apple and IBM.

Lyco Means Total Service.

ATARI ATARI

Mark "Mac" Bowser, Sales Manager

I would personally like to thank all of our past customers for helping to make Lyco Computer one of the largest mail order companies and a leader in the industry. Also, I would like to extend my personal invitation to all computer enthusiasts who have not experienced the services that we provide. Please call our trained sales staff at our toll free number to inquire about our diverse product line and weekly specials.

Call Lyco

First and foremost our philosophy is to keep abreast of the changing market so that we can provide you with not only factory-fresh merchandise but also the newest models offered by the manufacturers at the absolute best possible prices. And we offer the widest selection of computer hardware, software and accessories.

Feel free to call Lyco if you want to know more about a particular item. I can't stress enough that our toll-free number is not just for orders. Many companies have a toll-free number for ordering, but if you just want to ask a question about a product, you have to make a toll call. Not at Lyco. Our trained sales staff is knowledgeable about all the products we stock and is happy to answer any questions you may have. We will do our best to make sure that the product you select will fit your application. We also have Saturday hours — one more reason to call us for all your computer needs.

Order Now

Once you've placed your order with Lyco, we don't forget about you. Our friendly, professional customer service representatives will find answers to your questions about the status of an order, warranties, product availability, or prices.

Lyco Computer stocks a multimillion dollar inventory of factory-fresh merchandise. Chances are we have exactly what you want right in our warehouse. And that means you'll get it fast. In fact, orders are normally shipped within 24 hours. Free shipping on prepaid cash orders over $50, and there is no deposit required on C.O.D. orders. Air freight or UPS Blue/Red Label shipping is available, too. And all products carry the full manufacturers' warranties.

Here's How

I can't see why anyone would shop anywhere else. Selection from our huge in-stock inventory, best price, service that can't be beat — we've got it all here at Lyco Computer.

TO ORDER, CALL TOLL-FREE: 1-800-233-8760
New PA Wats: 1-800-233-8760
Outside Continental US Call: **1-717-494-1030**

Hours: 9AM to 8PM, Mon. - Thurs.
9AM to 6PM, Friday — 10AM to 6PM, Saturday

VISA

For Customer Service, call 1-717-494-1670,
9AM to 5PM, Mon. - Fri.
Or write: Lyco Computer, Inc.
P.O. Box 5088, Jersey Shore, PA 17740

MasterCard

Risk-Free Policy: • full manufacturers' warranties • no sales tax outside PA • prices show 4% cash discount; add 4% for credit cards • APO, FPO, international: add $5 plus 3% for priority • 4-week clearance on personal checks • we check for credit card theft • compatability not guaranteed • return authorization required • price/availability subject to change • prepaid orders under $50 in Continental US, add $3.00

Attention Educational Institutions:
If you are not currently using our educational service program, please call our representatives for details.

Access:
- Triple Pack $11.95
- Leader Board Pack $14.95

Activision:
- Hitch Hikers $13.95
- Leather Goddesses $19.95
- Music Studio $19.95

Broderbund:
- Print Shop $25.49
- Print Shop Compan. $22.95
- Graphic Lib. I, II, III $13.49
- Bank St. Writer $27.95

Electronic Arts:
- Pinball Con Set $8.95
- Lords of Conquest $8.95
- Starfleet I $32.95
- Chess Master 2000 ..... $25.95
- Music Con Set $8.95
- Super Boulderdash $8.95
- One on One $8.95

Firebird:
- The Pawn $22.95

Microleague:
- Microleag. Baseball $22.95
- General Manager $16.95
- Stat Disk $13.95

Microprose:
- Conflict in Vietnam $22.95
- F-15 Strike Eagle $19.95
- Kennedy Approach $13.95
- Silent Service $19.95
- Top Gunner $13.95

Optimized Systems:
- Action $44.95
- Action Tool Kit $16.95
- Basic XE $44.95
- Basic XL $33.95
- Basic XL Tool Kit $16.95
- Personal Pascal 2 .... $64.95

Strategic Simulations:
- Battle of Antietem $28.95
- Battlecruiser $33.95
- Nam $22.95
- Phantasie $22.95
- Wargame Construc. $16.95
- Warship $33.95
- Wizards Crown $22.95

Sublogic:
- Flight Simulator II $31.49
- Night Mission Pinball ... $18.95
- Scenery #1-#6 ea. $12.95
- Scenery #7 .... $14.95

ATARI ST

Access:
- Leader Board $22.95
- Tournament #1 .... $11.95
- 10th Frame $22.95

Activision:
- Champion. Baseball $22.95
- Champion. Basketball . $22.95
- Championship Golf .... $New
- GFL Football $22.95
- Leather Goddesses $22.95
- Music Studio $27.95
- Paint Works $22.95
- Shanghai $22.95
- Bureaucracy $22.95
- Top Fuel $19.95

Electronic Arts:
- Arctic Fox $25.95
- Empire $32.95
- Starfleet I $32.95
- Chess Master 2000 ..... $25.95
- Gridiron $32.95

Epyx:
- Sub Battle Simu
- World Games ..
- Wrestling ....
- Winter Games .

Firebird:
- Pawn ....
- Starglider ....
- Golden Path ....
- Guild of Thieves
- Tracker ....

Microleague:
- Microleague Bas
- General Manage
- Wrestling ....

Microprose:
- Silent Service ..
- F-15 Strike Eagl

Optimized Syst
- Personal Pascal
- Personal Prolog

Strategic Simul
- Phantasie ....
- Phantasie II .....
- Road War 2000
- Colonial Conque

Sublogic:
- Flight Simulator
- Scenery Disk ...

Unison World:
- Art Gallery 1 or
- Print Master .....
- Fonts & Borders

COMM

Access:
- Echelon ....
- Mach 5 ....
- Mach – 128 ....
- 10th Frame .....
- Triple Pack .....
- Wld. Cl. Leader
- Famous Course
- Famous Course
- Leader Board P

Action Soft:
- Up Periscope ...
- Thunderchoppe

Activision:
- Champion. Bask
- Music Studio ....
- Leather Goddes
- Top Fuel Elimin
- Beyond Zork ....
- GFL Football ...
- Gee Bee Air Ra
- Last Ninja ....
- Might & Majic ..
- Nord & Bert .....

Berkeley Softw
- Deskpak I ....
- Fontpak I ....
- Geodex ....
- Geofile ....
- Geo Calc ....
- Geos 64 ....
- Geos 128 ....
- Geowrite ....

Broderbund:
- Bank St. Writer
- Carmen San Di
- Graphic Lib. I, I
- Print Shop ......
- Print Shop Com
- Print Shop Pape
- Cauldron ....
- Superbike Chall

Free ship

ARI ST / COMMODORE / COMMODORE / AMIGA / IBM

tor ... $22.95
... $22.95
... $22.95
... $22.95

... $25.95
... $25.95
... $25.95
... $25.95
... $25.95

ball .. $33.95
... $16.95
... $25.95

... $22.95
... $24.95

ms:
... $42.95
... $51.95

tions:
... $22.95
... $22.95
... $22.95
st ... $22.95

... $31.49
... $14.95

... $14.95
... $19.95
... $17.95

ODORE

... $25.95
... $19.95
... $28.95
... $22.95
... $11.95
3rd. ... $22.95
#1 ... $11.95
#2 ... $11.95
ck ... $14.95

... $18.95
... $18.95

tball . $19.95
... $19.95
es ... $22.95
tor ... $15.95
... $25.95
... $19.95
y ... $16.95
... $19.95
... $19.95
... $19.95

rks:
... $20.95
... $17.95
... $23.95
... $29.95
... $29.95
... $35.95
... $39.95
... $29.95

... $27.95
go ... $19.95
III ... $13.95
... $25.49
an. ... $22.95
... $12.95
... $16.95
nge ... $11.95

Electronic Arts:
Bard's Tale ... $25.95
Legacy of The Ancients ... $20.95
Marble Madness ... $20.95
Starfleet I ... $25.95
Yeager's AFT ... $22.95

Epyx:
Destroyer ... $22.95
Fastload ... $19.95
Sub Battle ... $22.95
Winter Games ... $22.95
California Games ... $22.95
Graphics Scrapbook ... $11.95
Str. Sports Basketball . $22.95
Summer Games II ... $22.95
World Games ... $22.95

Firebird:
Elite ... $18.95
Golden Path ... $22.95
Guild of Thieves ... $22.95
Pawn ... $22.95
Tracker ... $22.95
Starglider ... $18.95
Sentry ... $22.95

Microleague:
Microleag. Baseball ... $22.95
General Manager ... $16.95
Stat Disk ... $13.95
Microleag. Wrestling ... $16.95
'87 Team Disk ... $11.95

Microprose:
Airborne Ranger ... $22.95
F-15 Strike Eagle ... $19.95
Gunship ... $19.95
Kennedy Approach ... $13.95
Silent Service ... $19.95
Solo Flight ... $13.95
Top Gunner ... $13.95
Pirates ... $22.95
Stealth Fighter ... $22.95

Origin:
Autoduel ... $28.95
Ultima III ... $22.95
Ultima IV ... $33.95
Moebius ... $22.95

Software Simulations:
Pure Stat Baseball ... $22.95
Football ... $17.95
Pure Stat College Basketball ... $22.95

Springboard:
Newsroom ... $29.95
Certificate Maker ... $29.95
Clip Art Vol. #1 ... $17.95
Clip Art Vol. #2 ... $23.95
Clip Art Vol. #3 ... $17.95
Graphics Expander ... $21.95

Strategic Simulations:
Gettysburg ... $33.95
Phantasie II ... $22.95
Phantasie III ... $22.95
Ring of Zelfin ... $22.95
Road War 2000 ... $22.95
Shard of Spring ... $22.95
Wizards Crown ... $22.95
Wargame Constr. ... $16.95
Battlecruiser ... $33.95
Battle of Antetiem ... $28.95
Eternal Dagger ... $22.95
Shiloh ... $22.95

Sublogic:
Flight Simulator II ... $31.49
Jet Simulator ... $24.95
Night Mission Pinball ... $18.95
Scenery Disk 1-6 ... $12.95

Timeworks:
Partner C64 ... $24.95
Partner 128 ... $39.95
Swift Calc 128 ... $29.95

Wordwriter 128 ... $29.95

Unison World:
Art Gallery 1 or 2 ... $14.95
Print Master ... $17.95

apple

Access:
Triple Pack ... $11.95
World Class Leader Board ... $23.95
Famous Courses ... $11.95

Activision:
Zork Trilogy ... $39.95
Game Maker ... $29.95
Leather Goddesses ... $22.95
Term Paper ... $29.95
Champ. Baseball ... $22.95
Champ. Basketball ... $22.95
GFL Football ... $22.95
Top Fuel Eliminator ... $19.95

Broderbund:
Airheart ... $22.95
Print Shop ... $26.95
Print Shop Comp. ... $22.95
Carmen S. Diego (USA) . $22.95
On Balance ... $56.95
Bank St. Writer + ... $44.95

Electronic Arts:
Bards Tale II ... $32.95
Legacy of the Ancients ... $25.95

Epyx:
Destroyer ... $22.95
Movie Monster ... $11.95
St. Sports Basketball ... $22.95
Sub Battle Simulator ... $22.95
Winter Games ... $22.95
World Games ... $22.95
Wrestling ... $22.95

Firebird:
Colossus IV Chess ... $22.95
Elite ... $19.95
The Pawn ... $25.95
Starglider ... $25.95

Microleague:
Microleag. Baseball ... $22.95
General Manager ... $16.95
Stat Disk ... $11.95
'86 Team Disk ... $11.95

Microprose:
Crusade in Europe ... $22.95
Decision in Desert ... $22.95
F-15 Strike Eagle ... $19.95
Silent Service ... $19.95

Strategic Simulations:
Battlecruiser ... $33.95
Colonial Conquest ... $22.95
Gettysburg ... $33.95
Phantasie III ... $22.95
Realms of Darkness ... $22.95
Shard of Spring ... $22.95
Wizards Crown ... $22.95
Eternal Dagger ... $22.95
Shiloh ... $NEW

Sublogic:
Flight Simulator II ... $30.95
Jet Simulator ... $24.95
Night Mission Pinball ... $21.95
Scenery #1-# ea. ... $12.95

Unison World:
Art Gallery 2 ... $14.95
Print Master ... $19.95

Access:
Leader Board ... $22.95
Tournament #1 ... $11.95

Activision:
Champ. Basketball ... $25.95
Championship Golf ... $22.95
GFL Football ... $25.95
Music Studio ... $27.95

Electronic Arts:
Deluxe Print II ... $48.95
Rockford ... $25.95
Empire ... $31.95
Scrabble ... $25.95

Epyx:
Apshai Trilogy ... $11.95
Rogue ... $22.95
Winter Games ... $22.95
World Games ... $22.95

Firebird:
Guild of Thieves ... $25.95
Pawn ... $25.95
Starglider ... $25.95

Microprose:
Silent Service ... $22.95

Sublogic:
Flight Simulator II ... $31.49
Jet Simulator ... $31.49
Scenery Disk ... $CALL

Unison World:
Print Master ... $19.95
Art Gallery 1 or 2 ... $14.95
Fonts & Borders ... $17.95

IBM

Access:
Wld. Cl. Leader Board . $27.95
10th Frame ... $27.95

Activision:
Champ. Baseball ... $22.95
Champ. Basketball ... $22.95
Zork Trilogy ... $39.95
Leather Goddesses ... $22.95
Beyond Zork ... $27.95

Broderbund:
Ancient Art of War ... $25.95
Print Shop ... $32.95
Print Shop Compan. ... $31.95
Graphic Lib. I or II ... $19.95
Karateka ... $19.95
Toy Shop ... $22.95
Bank St. Writer + ... $44.95
Ancient Art of War at Sea ... $25.95

Electronic Arts:
Pegasus ... $25.95
Weaver Baseball ... $25.95
Rockford ... $25.95

Epyx:
Apshai Trilogy ... $11.95
Create A Calendar ... $15.95
Destroyer ... $22.95
Movie Monster ... $11.95
Rogue ... $16.95
St. Sports Basketball ... $22.95
Sub Battle Simulator ... $22.95
Winter Games ... $22.95
World Games ... $22.95

Firebird:
Starglider ... $25.95
Guild of Thieves ... $25.95

Microleague:
Microleag. Baseball ... $22.95
General Manager ... $16.95
Stat Disk ... $13.95

Microprose:
Conflict in Vietnam ... $22.95
Crusade in Europe ... $22.95

Decision in Desert ... $22.95
F-15 Strike Eagle ... $22.95
Silent Service ... $22.95
Gunship ... $27.95

Origin:
Ultima I ... $22.95
Ultima III ... $22.95
Ultima IV ... $33.95
Moebius ... $33.95
Ogre ... $16.95

Strategic Simulations:
Road War 2000 ... $22.95
Gettysburg ... $33.95
Wizards Crown ... $22.95

Sublogic:
Jet Simulator ... $31.49
Scenery Japan ... $17.95
Scenery San Fran. ... $17.95
Scenery #1-#6 ... $64.95
Flight Simulator ... $34.95

Timeworks:
Swiftcalc ... $39.95
Wordwriter ... $34.95

Unison World:
Art Gallery 2 ... $14.95
News Master ... $49.95
Print Master (+) ... $29.95
Fonts & Borders ... $17.95

Diskettes

5-1/4

Maxell:
SSDD ... $7.95
DSDD ... $8.95

Bonus:
SSDD ... $5.95
DSDD ... $6.95

SKC:
DSDD ... $6.95
DSHD ... $13.95
Generic DSDD ... $4.95

Verbatim:
SSDD ... $8.99
DSDD ... $11.50

3.5

Maxell:
SSDD ... $11.50
DSDD ... $17.95

Verbatim:
SSDD ... $14.50
DSDD ... $20.95

SKC:
SSDD ... $11.95
DSDD ... $13.99
Generic SSDD ... $11.50
Generic DSDD ... $12.95

Joysticks

Tac 3 ... $8.95
Tac 2 ... $8.95
Tac 5 ... $11.95
Tac 1 + IBM/AP ... $17.95
Economy ... $4.49
Slik Stick ... $4.95
Black Max ... $7.95
Boss ... $10.95
3-Way ... $17.95
Bathandle ... $14.95
500XJ ... $10.95
500XJ-Apple/PC ... $21.95
Winner 909 ... $19.95
Wico IBM/AP ... $19.95
Contriver Joystick Bd .. $29.95

ping on Prepaid cash orders over $50 in the Continental US.

Program 6: Tiles—Atari ST Version

```
' Copyright 1988 COMPUTE! Publications, Inc.←
' All Rights Reserved←
rez%=XBIOS(4)←
IF rez%<>Ø←
  alrt$="Please switch to|Low Resolution."←
  ALERT 3,alrt$,1,"OK",b←
  END←
ENDIF←
GRAPHMODE 2←
DIM hitnum(1),hnum(1),high(1),score(1),hit(59)
,hrow(3Ø),hcol(3Ø)←
DIM c(17),t(6),dec(1),bc(3Ø)←
HIDEM←
@save_palette←
new_game:←
@blank_screen←
FOR i=1 TO 16←
  @colors←
NEXT i←
FOR i=Ø TO 1←
  hitnum(i)=Ø←
  hnum(i)=Ø←
  score(i)=5ØØ←
  high(i)=5ØØ←
NEXT i←
@board←
@set_palette←
f!=TRUE←
start:←
pl=numpl-pl←
IF score(pl)=Ø←
  GOTO start←
ENDIF←
PUT 75,169,erase$←
DEFTEXT 4,1,Ø,6←
TEXT 9Ø,178,"Click to view"←
WHILE MOUSEK=Ø←
  row=RANDOM(6)←
  col=RANDOM(1Ø)←
  @player(pl)←
  @random_num←
  IF f!=TRUE←
    @square←
  ENDIF←
  @tiles←
WEND←
f!=FALSE←
@show_tiles←
PUT 75,169,erase$←
DEFTEXT 4,1,Ø,6←
TEXT 79,178,"Click when ready"←
WHILE MOUSEK=Ø AND score(pl)>Ø←
  @tiles←
  @timer←
  @player(pl)←
WEND←
SOUND 1,Ø←
@hide_tiles←
SHOWM←
DEFMOUSE 3←
PUT 75,169,erase$←
DEFTEXT 4,1,Ø,6←
TEXT 1Ø5,178,"Tiles = "+STR$((hitnum(pl)+1))←
PAUSE 5Ø←
@do_search←
HIDEM←
IF score(pl)>Ø←
  hnum(pl)=hnum(pl)+hitnum(pl)←
  hitnum(pl)=hitnum(pl)+1←
ENDIF←
IF score(numpl-pl)=Ø←
  GOTO end_game←
ENDIF←
IF score(pl)>high(pl)←
  high(pl)=score(pl)←
  @print_score←
ENDIF←
IF hitnum(pl)=>3Ø←
  @thirty←
ENDIF←
GOTO start←
end_game:←
IF numpl=1 AND score(Ø)=Ø AND score(1)=Ø←
  GOTO end_it←
ELSE←
  IF numpl=Ø AND score(Ø)=Ø←
    GOTO end_it←
  ENDIF←
ENDIF←
GOTO start←
end_it:←
WAVE Ø,Ø←
SOUND 1,Ø←
SOUND 2,Ø←
PUT 75,169,erase$←
DEFTEXT 4,1,Ø,6←
TEXT 81,178,"Play Again (Y/N)"←
WHILE k$<>"Y" AND k$<>"N"←
  @tiles←
  k$=UPPER$(INKEY$)←
WEND←
IF k$="N"←
  CLS←
  @restore_palette←
  END←
ENDIF←
GOTO new_game←
PROCEDURE print_score←
  t$=STR$(score(pl))←
  FOR i=LEN(t$) TO 4←
    t$="Ø"+t$←
  NEXT i←
  DEFTEXT 16,Ø,Ø,6←
  PUT 25Ø,45+25*pl*2,black$←
  TEXT 256,53+25*pl*2,t$←
  t$=STR$(high(pl))←
  FOR i=LEN(t$) TO 4←
    t$="Ø"+t$←
  NEXT i←
  PUT 25Ø,7Ø+25*pl*2,black$←
  TEXT 256,78+25*pl*2,t$←
RETURN←
PROCEDURE random_num←
  REPEAT←
    c=RANDOM(15)+1←
  UNTIL c<>9←
RETURN←
PROCEDURE tiles←
  t$="Tiles"←
  FOR i=1 TO 5←
    t(i)=t(i+1)←
  NEXT i←
  t(5)=RANDOM(15)+1←
  FOR i=1 TO 5←
    DEFTEXT t(i),4,Ø,32←
    TEXT 22Ø+15*i,25,MID$(t$,i,1)←
  NEXT i←
RETURN←
PROCEDURE square←
  DEFFILL c←
  x1=4+22*col←
  y1=12+25*row←
  x2=2Ø+22*col←
  y2=33+25*row←
  PBOX x1,y1,x2,y2←
  COLOR Ø←
  DEFLINE 1,2←
  BOX x1,y1-1,x2,y2+1←
RETURN←
PROCEDURE sound←
  n=RANDOM(12)←
  SOUND 1,15,n,6,5←
RETURN←
PROCEDURE restore_palette←
  LOCAL i%←
  FOR i%=Ø TO 15←
    SETCOLOR i%,palette%(i%)←
  NEXT i%←
RETURN←
PROCEDURE save_palette←
```

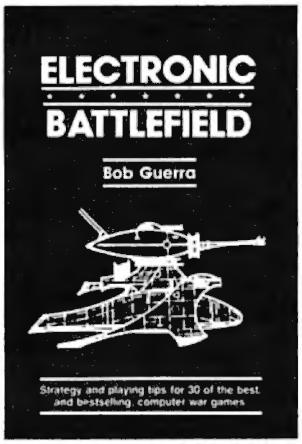
```
  LOCAL i%←
  DIM palette%(15)←
  FOR i%=Ø TO 15←
    palette%(i%)=XBIOS(7,W:i%,W:-1)←
  NEXT i%←
RETURN←
PROCEDURE set_palette←
  LOCAL i%←
  RESTORE←
  FOR i%=Ø TO 15←
    READ r%,g%,b%←
    SETCOLOR i%,r%,g%,b%←
  NEXT i%←
RETURN←
PROCEDURE blank_screen←
  LOCAL i%←
  FOR i%=Ø TO 15←
    SETCOLOR i%,Ø,Ø,Ø←
  NEXT i%←
  DATA 1,1,Ø,7,Ø,Ø,7,3,Ø,7,5,Ø,7,7,Ø,4,7,Ø,Ø,7
,Ø,Ø,7,5,Ø,7,7,Ø,5,7←
  DATA Ø,2,7,Ø,Ø,7,5,Ø,7,7,Ø,7,7,Ø,4,7,7,7←
RETURN←
PROCEDURE show_tiles←
  @clear_tiles←
  FOR i=Ø TO 29←
    hrow(i)=-1←
    hcol(i)=-1←
  NEXT i←
  FOR i=Ø TO hitnum(pl)←
    get_coord:←
    er=Ø←
    hrow(i)=RANDOM(6)←
    hcol(i)=RANDOM(1Ø)←
    IF i>=1←
      FOR x=Ø TO i-1←
        IF hrow(x)=hrow(i) AND hcol(x)=hcol(i)
        ←
          er=1←
        ENDIF←
      NEXT x←
    ENDIF←
    IF er=1←
      GOTO get_coord←
    ENDIF←
    REPEAT←
      bc(i)=RANDOM(16)←
    UNTIL bc(i)<>9 AND bc(i)<>Ø←
    row=hrow(i)←
    col=hcol(i)←
    c=bc(i)←
    @square←
  NEXT i←
RETURN←
PROCEDURE player(pl)←
  IF pl=Ø←
    t1$="Player 2"←
    t2$="Player 1"←
    y1=4Ø←
    y2=9Ø←
  ELSE←
    t1$="Player 1"←
    t2$="Player 2"←
    y1=9Ø←
    y2=4Ø←
  ENDIF←
  DEFTEXT Ø,Ø,Ø,6←
  TEXT 243,y2,t1$←
  @colors←
  FOR i=1 TO 8←
    DEFTEXT c(i)←
    TEXT 235+8*i,y1,MID$(t2$,i,1)←
  NEXT i←
RETURN←
PROCEDURE timer←
  z=INT((TIMER)/2)/1ØØ←
  IF z>=htime+1←
    @dec_score←
    htime=z←
  ENDIF←
RETURN←
PROCEDURE dec_score←
  dec(pl)=4Ø←
  IF score(pl)<2ØØØ←
    dec(pl)=2Ø←
  ELSE←
    IF score>5ØØØ←
      dec(pl)=6Ø←
    ENDIF←
  ENDIF←
  score(pl)=score(pl)-dec(pl)←
  IF score(pl)<=Ø←
    score(pl)=Ø←
  ENDIF←
  @print_score←
  SOUND 1,15,1,5,1←
  SOUND 1,Ø←
RETURN←
PROCEDURE hide_tiles←
  c=Ø←
  FOR i=Ø TO hitnum(pl)←
    row=hrow(i)←
    col=hcol(i)←
    @square←

  NEXT i←
RETURN←
PROCEDURE do_search←
  SHOWM←
  hits=Ø←
  search:←
  WHILE MOUSEK=Ø←
    c1=INT(MOUSEX/22+1)-1←
    r=INT(MOUSEY/24+Ø.4)-1←
  WEND←
  PAUSE 15←
  FOR i=Ø TO hitnum(pl)←
    IF c1=hcol(i) AND r=hrow(i)←
      row=hrow(i)←
      col=hcol(i)←
      c=bc(i)←
      @square←
      hrow(i)=-1←
      hcol(i)=-1←
      hit=1←
    ENDIF←
  NEXT i←
  IF hit=1←
    @hit_sound(TRUE)←
    score(pl)=score(pl)+1ØØ←
    hits=hits+1←
    hit=Ø←
  ELSE←
    @hit_sound(FALSE)←
    score(pl)=score(pl)-1ØØ←
    IF score(pl)<Ø←
      score(pl)=Ø←
    ENDIF←
  ENDIF←
  @print_score←
  IF hits>hitnum(pl) OR score(pl)=Ø←
    GOTO end_search←
  ELSE←
    GOTO search←
  ENDIF←
  end_search:←
  FOR i=Ø TO hitnum(pl)←
    PAUSE 15←
    c=bc(i)←
    row=hrow(i)←
    col=hcol(i)←
    IF hrow(i)<>-1←
      c=bc(i)←
      @square←
    ENDIF←
  NEXT i←
  PAUSE 1ØØ←
  @clear_tiles←
RETURN←
PROCEDURE hit_sound(flag!)←
  IF flag!=TRUE←
    oct=6←
  ELSE←
    oct=2←
```

```
  ENDIF←
  SOUND 1,15,4,oct←
  SOUND 2,15,7,oct←
  WAVE 3,3,0,65535,10←
RETURN←
PROCEDURE colors←
  FOR i=1 TO 16←
    c(i)=c(i+1)←
  NEXT i←
  REPEAT←
    c(17)=RANDOM(15)+1←
  UNTIL c(17)<>6 AND c(17)<>9←
RETURN←
PROCEDURE thirty←
  PUT 75,169,erase$←
  FOR t=1 TO 100←
    @colors←
    t$="30 Tiles Found!"←
    FOR i=1 TO 15←
      DEFTEXT c(i),1,0,6←
      TEXT 81+8*i,178,MID$(t$,i,1)←
    NEXT i←
    @tiles←
    @song←
  NEXT t←
RETURN←
PROCEDURE song←
  READ n1←
  IF n1=999←
    RESTORE sdat←
    READ n1←
  ENDIF←
  SOUND 1,15,n1,6,2←
  sdat:←
  DATA 3,5,5,7,1,3,5,7,6,999←
RETURN←
PROCEDURE clear_tiles←
  c=0←
  FOR col=0 TO 9←
    FOR row=0 TO 5←
      @square←
    NEXT row←
  NEXT col←
RETURN←
PROCEDURE board←
  CLS←
  DEFLINE 1,3←
COLOR 9←
FOR x=0 TO 6←
  LINE 0,x*25+10,223,x*25+10←
NEXT x←
FOR x=0 TO 10←
  LINE x*22+1,10,x*22+1,160←
NEXT x←
FOR row=0 TO 5←
  FOR col=0 TO 9←
    @random_num←
    @square←
  NEXT col←
NEXT row←
COLOR 13←
RBOX 75,169,211,181←
DEFFILL 9←
FILL 85,175←
GET 75,169,211,181,erase$←
COLOR 2←
RBOX 233,30,315,135←
COLOR 4←
RBOX 232,29,316,136←
DEFFILL 6←
FILL 300,75←
DEFTEXT 0,0,0,6←
TEXT 243,40,"Player 1"←
TEXT 250,65,"High 1"←
TEXT 243,90,"Player 2"←
TEXT 250,115,"High 2"←
DEFFILL 0←
FOR x=0 TO 3←
  PBOX 250,45+25*x,300,55+25*x←
NEXT x←
GET 250,45,300,55,black$←
DEFTEXT 4,1,0,6←
TEXT 84,178,"1 or 2 Players?"←
DEFTEXT 3,0,0,4←
TEXT 30,190,"Copyright 1988 COMPUTE! Publica
tions, Inc."←
TEXT 90,197,"All Rights Reserved"←
@set_palette←
k$=""←
WHILE k$<>"1" AND k$<>"2"←
  k$=INKEY$←
  row=RANDOM(6)←
  col=RANDOM(10)←
  @random_num←
  @square←
  @sound←
  @tiles←
WEND←
SOUND 1,0←
numpl=ASC(k$)-49←
  FOR pl=0 TO numpl←
    @print_score←
  NEXT pl←
  pl=numpl←
RETURN←
```

New from COMPUTE!

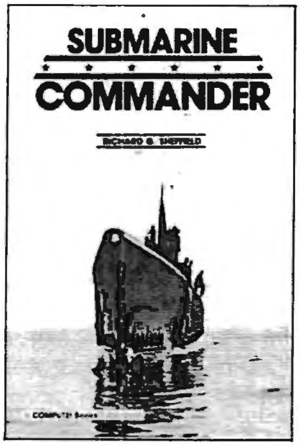
War game simulations come alive with two new books from COMPUTE!. Whether you're dodging torpedoes, bombing targets, or planning to defend against an enemy invasion, these books contain the hints and strategies you need for a successful defense—and offense. There's historical information on actual battles, technical information about your war machines, plus many tips on using hidden features of the software.



The Electronic Battlefield
Bob Guerra
ISBN 0-87455-117-X
$12.95

This comprehensive reference book on computer war games—how to play them and what strategies to use—will find a place on thousands of computer owners' desks. Whether acting as fierce opponent or impartial referee, the computer is perfect for war game play. *The Electronic Battlefield* offers general and specific hints, tips, and play techniques for all the most popular computer games. Filled with detailed information on such games as *Silent Service, Roadwar 2000, Field of Fire, Arctic Fox,* and *Balance of Power,* this book will appeal to players of all ages and levels of experience. Also included in this book are undocumented features of games like *F-15 Strike Eagle,* command reference charts for many of the games, and even tips from the game designers. The perfect resource for any electronic war gamer.

Sub Commander: Tactics and Strategy for WWII Submarine Simulations
Richard G. Sheffield
ISBN 0-87455-127-7
$12.95



Run silent, run deep. This hands-on approach to popular submarine simulators like *Silent Service, Gato, Sub Battle,* and *Up Periscope* puts you at the periscope of an American boat in World War II. This collection of undersea warfare tactics and strategies will appeal to every simulator captain. Information about actual World War II submarines, crews, and captains provides an excellent introduction. Detailed accounts of real submarine battles illustrate ways you can pilot your own electronic sub. The heart of the book outlines offensive and defensive tactics and maneuvers that bring you into torpedo range and get you out again safely. Learn how to decimate heavily escorted convoys, how to shoot "down the throat" at a charging destroyer, and how to weather determined-depth charge attacks. You'll be able to immediately apply these techniques and tricks to your own submarine software.

Note: The books do not include any of the software products mentioned.

To order COMPUTE! books, call toll free 1-800-346-6767 (in NY 212-887-8525) or write COMPUTE! Books, Customer Service, P.O. Box 5038, F.D.R. Station, New York, NY 10150. Customer Service hours are 10:00 a.m. to 12:30 p.m. and 1:30 p.m. to 3:30 p.m. EST Monday through Friday.

Please add $2.00 per book shipping and handling. NC residents add 5 percent sales tax and NY residents add 8.25 percent sales tax. Please allow 4–6 weeks for delivery.

COMPUTE! books are available outside the United States from subsidiaries of McGraw-Hill International Book Company.

COMPUTE! Publications, Inc. abc
A Capital Cities/ABC, Inc. Company

EOS: Earth Orbit Stations

Russell H. Fisher

Requirements: Commodore 64, Apple II.

Earth Orbit Stations, or *EOS*, is a game of space development for one to four players. Although several variations and scenarios are included, one would be well advised to start with the scenario designated "Research Mission" to become familiar with the game. Depending on the mission selected, playing time to completion can vary from a couple of hours to much, much longer (the package says 2–40 hours). Some game scenarios can be extremely complicated. If you like action-oriented games, you should look elsewhere; on the other hand, if you like a mental challenge, *EOS* could be the one for you.

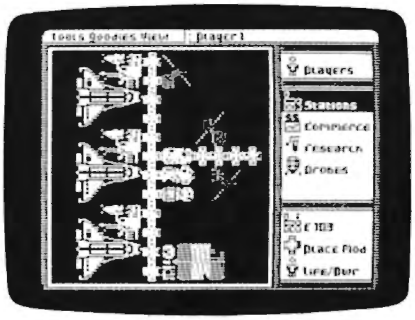
Tackling The Final Frontier

EOS begins in the spring of 1996 and continues thereafter in annual quarters. During each player's turn, several actions may be performed. Some of them are quite clever, including an online newspaper filled with information that pertains to the scenario being played. I chose to read the *EOS News* at the beginning of each quarter. Sometimes the news can be astounding: I was surprised at the beginning of one quarter to read that I had received an emergency bank loan during that period. The news is often filled with mundane announcements about price increases and more interesting announcements about Space Shuttle events; these announcements, however mundane, can affect your scenario, so pay attention. The news also contains little tidbits attributed to unnamed sources, so don't believe all that you read.

In the beginning games, Research and Drydock, one must construct space stations using modular parts. All stations must have a Command Module, Logistics Module, Life Module, Galley and Gym Module, a power source, and assorted connectors to link the modules together. Since you pay for each addition to your station, consideration of economic factors is a must. Other modules may then be added, and these are used for research or commerce. In early stages of station development, it's usually best to use these modules to develop a source of income. As the game develops, the player can select what modules are to be used for research or commerce.

Among the many kinds of modules available are solar power units, biology labs, computer labs, energy platforms, space telescopes, and weather centers.

During each turn, you can examine usage of the commercial modules during the previous quarter and adjust your prices accordingly. Other players' pricing can affect your income. You can advertise, which will increase your income and decrease other players' income.



Earth Orbit Stations *puts you in charge of the development of an ongoing space-station/space-commerce facility.*

Research Materials And Venture Capital

In addition to the daily newspaper, you may visit one of several libraries to learn about station or module characteristics, or to get information about any of the planets or moons in our solar system. You also may visit the bank to borrow money, or to deposit/withdraw funds from savings.

Longer versions of the game require space development by building Cargo Liners and Jupiter Explorers. This level of ambition takes money as well as constantly advancing technologies, making the balance between commerce and R&D an ongoing challenge.

There are a couple of minor quibbles I have with this Electronic Arts release. I reviewed the game on a 64. Although I've never before had difficulty loading an EA product, my copy of this game was hard to load. For some reason, the game loads more predictably when my printer interface is disconnected from the cassette port, although there is no mention of special conditions or requirements in the manual.

The game requires that you make a mission disk before play begins. If you follow the directions, several disk swaps are required. I wish EA had included recognition of a second drive to help automate this procedure. In any event, I found that the mission disk was not copy protected, and a copy can be made before the game is loaded.

On the whole, though, *EOS* offers a level of challenge unusual in space-related software. To succeed at this game requires careful thought. The computer can be selected as one or more competitors, and competitors can be added to play at any time. Of the seven scenarios, I have played only the first two to conclusion. One of the features of the game is that after a winner is declared, you can continue playing, apparently for as many quarters as you like. *EOS* is an excellent introduction to the many problems man must overcome to succeed in space. In that regard this game could be an attractive choice for younger minds, as well as older minds with youthful outlooks.

EOS: Earth Orbit Stations
Electronic Arts
1820 Gateway Dr.
San Mateo, CA 94404
$29.95 Commodore version
$34.95 Apple II-series version

Earl Weaver Baseball

Neil Randall

Requirements: Amiga with a minimum of 512K (reviewed here), and IBM PC/Tandy and compatibles with a minimum of 256K, to be released in late 1987.

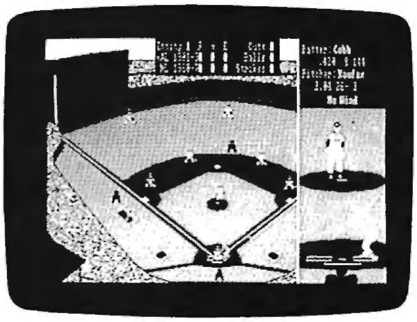
Baseball simulations have been popular ever since the introduction of the home computer. Intellivision's baseball game perhaps started the trend, and the baseball game for the old Odyssey was certainly that machine's strongest offering. SSI's *Computer Baseball* ruled the statistical games for a couple years, while Gamestar's *Star League Baseball* held us in thrall with its animation and graphics when it was introduced for the Commodore 64 and the Atari 800. Probably because of the game's unique relationship between the individual player and his or her favorite team, and the classic one-on-one showdown between pitcher and batter, baseball lends itself extremely well to the computer, overcoming in fact many of the limitations of computer sports.

For the most part, though, even the best baseball offerings have emphasized either the statistical game or the arcade game. *Computer Baseball* and *Star League Baseball* represent the earliest best of each world, while subLogic's *Baseball* and Accolade's *HardBall!* demonstrate the advances made in each of these separate categories. Beginning with Epyx's *World's Greatest Baseball Game*, designers began to attempt a merging of the two game styles. Epyx's game offered you the choice of managing the team or operating the players on the field (or both), and except for its difficult arcade mechanics, and some questionable statistical results, it worked very well.

With the coming of the more powerful computers, though, it was inevitable that sports games would increase in sophistication. Gamestar (now under Activision's umbrella) has released *Championship Baseball*, with its play/manage mode, and *Micro League Baseball* has appeared for these computers as well. The best of them all, though, is Electronic Arts' *Earl Weaver Baseball*. It is, without question, the closest we have to the ideal computer baseball game, and as a product and a program it is stunning to behold.

Earl's Input

Earl Weaver Baseball has been designed under the strategic guidance of (not surprisingly) Earl Weaver, the venerable former manager of the Baltimore Orioles. Combining both statistical and action games, it allows you to draft teams, create players, trade players among teams, keep track of stats throughout a season, create your own leagues, and even design your own ballparks. On the original disk, are 8 teams of all-time greats and 32 well-known parks, and you can play with these until you are ready to go out on your own.



Earl Weaver Baseball *is a feature-packed baseball simulation that offers, among other things, comprehensive statistical detail and extraordinary graphics and animation.*

Additionally, Electronic Arts offers data disks to supplement the game. Each contains all teams from a particular season (the 1986 season is available now) and all the appropriate parks. Nice as these are to have, they are hardly necessary. Stats creation is extremely easy (if time-consuming); all you need is the complete statistics page from your newspaper, and a few hours of time. To create leagues, teams, and parks, you access the Commissioner menu in the game and follow the instructions. Creating players is simply a matter of filling in the stats as they appear, one by one. You can even use split stats, for switch-hitters, and stats for secondary fielding positions.

But creating statistics is hardly what this game is about, any more than reading the box scores is what real baseball is all about. *Earl Weaver Baseball* is played, and managed, on the field, and here lie the game's greatest strengths.

First of all, on some versions, we have an announcer. Making use of the Amiga's speech, the game announces the pitcher's name at the start of the game, and each batter's name as he/she steps to the plate. (I say *he/she* because, even though the teams that come with the disk use only male players, there is absolutely no reason not to create female players. You can rename any player on the disk, and the program even allows you to play with the phonemes so the announcer can get the pronunciation right.) The speech synthesizer's other functions are to shout *Play Ball!* at the start of the game and to take the role of the home-plate umpire by telling you, upon request, the location of the last pitch.

Split-Screen Action

During the pitch, the screen is divided into two parts. The left two-thirds show the playing field from over home plate. The right third shows the pitcher-batter detail, from the vantage point of the home-plate umpire. When the ball is hit, the right third disappears, and the entire park comes into view.

You can use either the mouse, a joystick, or the keyboard to control the game's activities. The interface is well-designed and straightforward, and, for the first time, pitching seems to feel like pitching (even though it lacks the actual detail of Accolade's *Hardball!*). One of the few problems with the game is the difficulty, as batter, in seeing the pitch's height, and I hope the designers see fit to improve this in a later edition.

At any point during the pitching, you can call up the manager's menu and order your players into such actions as Guard the Lines, Charge from First, Shift the Outfield back and to the right, Bunt, Hit and Run or Run and Hit (the game makes the distinction), Squeeze Bunt, and so on. Actually, many of these are available only in Manage-only mode, since in Manage-and-Play mode you perform some of these on your own.

Advice From The Expert

If you aren't sure which play to call, you can ask Earl. When you do, the screen fills with advice from the great manager. Most of it is useful. If your pitcher is doing suspicious things (like giving up home runs), you can choose Conference. Here, the manager walks to the mound (actually, he bolts), and a screen appears showing the pitcher lying about his arm being fine and the catcher agreeing so that he doesn't get punched in the clubhouse later.

The management game offers a host of options, as many as exist in real baseball, but even here the game doesn't reach its peak. Despite all the strategic strengths, the action game is the strong point. What happens on the field is the closest computer approximation so far of the events on a real baseball field.

The timing seems perfect. Throwing out a fast runner from third base means a very close call at first, and on a slow bounce the runner will cross. On a single to the outfield, a slow runner will not make it from second to home, unless the outfielder's throwing arm is weak or unless he misses the cut-off man. Yes, in this game you throw to the cut-off man, then relay to the plate—unless the fielder's arm is good enough and he is close enough to throw it directly. Stolen bases are difficult, unless Maury Wills or Tim Raines is the runner, and even they get caught occasionally. A sacrifice fly will score a runner from third only if the runner is fast enough and only if the outfielder is deep enough.

Then there's what everyone else does. If the throw to the plate misses the cut-off man, the runner from first will usually make it to second (again, depending on his speed). On a ground ball to short, the catcher runs to cover first, and this is important because the game includes wild throws and fielding errors. It also includes wild pitches and past balls, and you can (at rare times) nail the runner trying to advance from third to home on such plays. On a grounder to first, the pitcher will cover first; while, on a bunt, the second baseman will cover first. In other words, all fielders take up their positions, on all plays, exactly as they do in real baseball. Given the importance of the cut-off man, and the fact that not all throws go where they're supposed to, this positioning quickly shows its importance.

There are umpires at each base, and the manager even comes out to argue close calls. Fly balls and ground balls are hit all over the field, but not in preselected places as in many games. Wind factors, stadium design, and pitcher's fatigue all figure in, as do injuries in a long season. The game even has a radar gun for tracking the speed of a pitch.

Play It Again, Earl

The most fun of all, though, is the instant replay feature. You can get a normal-speed replay or a slow-motion replay of any play, and you can even slow the replay down to a frame-by-frame breakdown. Furthermore, the replay is shown from a lower perspective, and watching the play develop is as realistic as anything you've ever seen in a computer sports game.

If you are a baseball fan, you will want this game. Period.

Earl Weaver Baseball
Electronic Arts
1820 Gateway Dr.
San Mateo, CA 94404
$49.95 Amiga version
$39.95 IBM PC/Tandy and compatibles version

Test Drive

Troy Tucker

Requirements: Commodore 64; Amiga with 512K minimum of memory; IBM PC/XT/AT, Tandy 1000, 1200, 3000, and compatibles with 256K minimum of memory with CGA/EGA or Hercules™; Atari ST. Joystick required.

Strap yourself in and get ready for the ride of your life. Accolade's latest simulation, *Test Drive*, puts you behind the wheels of the five fastest, most exotic cars in the world. You may choose from the Lamborghini Countach, Ferrari Testarossa, Porsche 911 Turbo, Lotus Turbo Esprit, and America's own Chevrolet Corvette. This high-speed simulation is guaranteed to give you sweaty palms.

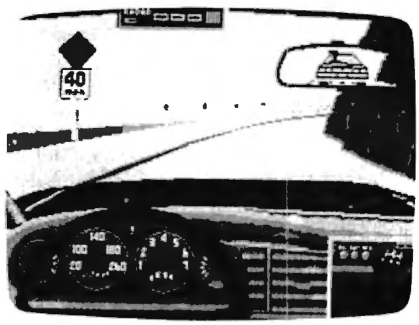
After booting up, *Test Drive* provides you with a beautifully detailed photo of each car and its corresponding specification sheet. Here, you are provided with all of the vital statistics relating to the car's performance and handling. For example, you can view the top speed, torque, acceleration, braking distance, and a few other items that separate these cars from the ones in their rear-view mirrors.

Drivers, Start Your Engines!

Once you have decided which car is right for you, a press of the joystick button puts the keys in the ignition and you in the driver's seat. The cockpit view is very authentic, giving you the feel of actually being in the car. The fully detailed instrument panel contains a speedometer, tachometer, and an oil pressure gauge. A working shifter is displayed at the lower part of the screen. Shift gears by pressing the joystick up or down and hitting the fire button. As an added touch, *Test Drive* provides you with the option to mimick the gear patterns of each car with the joystick.

You are also provided with a radar detector, which can be found in the upper left portion of the windshield, attached to the sun visor. The radar detector comes in very handy along the way as you encounter the radar traps. *Test Drive*'s fuzzbuster works just like the real thing, with five red lights on its face and a speaker to alert you of the danger ahead. When radar is first detected, you are alerted with an audible tone and a flashing light. As you get closer to the radar, more of the red lights flash. When all of the lights are on, you are being clocked by the Highway Patrol.

After getting the feel of the controls, you're ready to go. Your goal is to reach the top of the mountain as fast as you can and in one piece. The trip is divided into segments by a series of checkpoints. Only five crashes are allowed between each checkpoint, so you must drive carefully. You rev the engine into the red line and slam it into first, and you're off. Driving through the winding mountain road at breakneck speed is no easy task. Water slicks, potholes, and passing motorists will either slow your progress or cause you unfortunate mishaps. Occasionally you may even have a bug or two splatter on your windshield, just to make matters worse.



Trying to outrun Smokey in Test Drive.

Smokey Alert

If the terrain doesn't get you, the Smokies will. They patiently wait for speeding motorists driving flashy sportscars. And once they clock you breaking the speed limit, the chase is on. You can outrun him if you're good, or you can just pull over and take your medicine. If, however, you fail to get away, he'll force you off the road and write you a citation.

The superbly detailed graphics and realistic sound effects make *Test Drive* one of a kind. One drawback to playability is that you are limited to a single driving course. In addition to the mountain course, a nice drive through a cityscape or other terrain would have made the game seem more complete. Even at 140 mph, a mountain road can get boring after a while.

Overall, though, the game seems designed with the player in mind. Simulations in the past have always overwhelmed players with a vast array of controls and gauges to master, often taking the fun out of playing. *Test Drive* takes away the drudgery and adds playability without sacrificing realism. Simulation enthusiasts as well as arcade gamers and speed demons will want to check this one out. You wouldn't drive like this on real roads, so it's fun to be able to cut loose with *Test Drive.*

Test Drive
Accolade
20813 Stevens Creek Blvd.
Cupertino, CA 95014
$29.95 Commodore 64 version
$39.95 IBM and compatibles version
$44.95 Amiga and Atari versions

Project: Stealth Fighter

Ervin Bobo

Requirements: Commodore 64; conversions are planned for IBM PC and compatibles, and Atari ST.

There probably are not enough superlatives to describe this one, so I'll hedge by saying that *Project: Stealth Fighter* is a superlative flight simulator, from the opening title screens to the final mission assessment.

To understand *Stealth*, it will be helpful to understand that "stealth" technology is concerned with building aircraft that are invisible to radar. It is probably the most secret project of our armed forces, and from that, the obvious question arises: How can they simulate something nobody knows about?

Quite simple, really. Once those two parameters are known—"aircraft" and "radar invisible"—those involved with aircraft design can extrapolate the rest. While they may not be 100 percent correct in their assessments, they're probably very close. Since *Project: Stealth Fighter* does not have to concern itself with building materials, it concentrates on speed, altitude, and aircraft configuration. All the parameters used here seem to hang together logically.

The result is an aircraft of tomorrow flying the missions of tomorrow. (And given the current situation in the Middle East, some of these missions are being flown today.) This makes *Stealth* one of the best flight simulators ever.

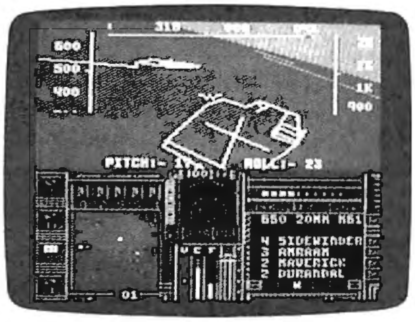
Meet The Enemy

In form, *Stealth* borrows heavily from the routines of the wildly successful *Gunship*, also from Microprose. The animation of the title screens is more sophisticated, showing what another year of experience can produce, and the mission planning is very similar. There are five scenarios from which to choose, one of them a training mission, but even these have three different levels: Cold War, Limited War, and Conventional War. These are further subdivided by pitting you against Green Opponents, Regular Opponents, or Expert Opponents, which are divided even further into Air or Ground targets.

Following these choices, you're given the details of your mission and your primary target. Maps of the four areas are included with the program, and a map will appear on your control console as well. The console map also functions as an automatic navigator, just as it did in *F-15 Strike Eagle*, making it easy for you to set course for the target, even if you understand nothing about map coordinates.

Now you arm the plane. There are four weapons bays—because hanging the weapons on conventional pylons would increase the radar profile—and a great variety of weapons. Since your mission could be air-to-air or a ground strike, it's important to have the right mix of weapons for the job. Fortunately, the default choices made by the program are usually enough, though you may vary the mix if you want.

All of this has taken place on side 1 of the disk. Now you'll flip it to side 2 to actually begin your mission. Perhaps we should note that, unlike *Gunship*, *Stealth* will load and run with extra disk drives or a printer attached to your computer, making the loading and playing of the game more convenient. A trade-off is that if you wish to save your pilot and his record, it must be done on a separate disk—certainly no hardship.



Project: Stealth Fighter *is a flight simulator for state-of-the-art aircraft technology.*

In flying a mission, you'll rely heavily on the keyboard overlay. It seems to be a rule that as flight simulators become more sophisticated, the controls become more complex. Given all that will be happening in a mission, it is an understandable complexity. For example, before you can fire on the enemy, you must open the bay doors, then arm one of the weapons bays by pressing the appropriate function key. Only then will the fire button launch a weapon. Then, to minimize your radar profile, press RETURN to shut down the weapons and close the bay doors.

Similarly, when flying, one key will be used for each of these moves: Engine On/Off, Flaps Extended/Retracted, Throttle Increase, Throttle Decrease, Landing Gear Up/Down, and so on. If this seems daunting, it is so only in the telling. The keyboard overlay helps keep things sorted out and simple.

Heads Up!

Stealth's heads-up display (HUD) is one of the best I've seen. At the top is your compass heading and your autonavigation mark. Running down the left side is a speed indicator, while on the right is an altitude reading. In the center is the horizon marker, which changes to a gunsight if you arm your cannon. Below that are pitch and roll indicators. All this information is delivered without cluttering your view through the windscreen.

On the control panel are indicators for throttle, fuel efficiency, positions of landing gear and speed brakes, vertical climb indicator, color CRT with satellite map, and a radarscope that will display the enemy and his missiles. Once again, the display manages this without clutter and with a very high degree of readability.

You may find that the most important displays on the console are those having to do with Electro-Magnetic Visibility—your radar profile. A row of LED lights shows your degree of visibility, and you must take steps to lower it: fly higher or lower, turn off targeting systems, close weapons bays, and so on. The F-19 Stealth does not have the high speed of some of your enemies, so it must rely on concealment in order to survive.

Your Mission, Should You Decide to Accept It. . .

When a mission has ended—either by your returning to base or aircraft carrier, or by being shot down and perhaps killed or captured—there will be a sequence of captioned still frames that summarize your mission. You may have carried out your mission flawlessly, or created an embarrassing international incident, or achieved one of several degrees between.

In any event, you'll be scored on several levels: how well you flew your mission, whether you accomplished your objective, whether you managed to return to base, and so on. All this boils down to a number called Mission Score. Yet another number indicates how well you maintained radar invisibility.

These two scores are combined for a total mission score, which determines what decorations or promotions in rank you may earn. Mission scores are cumulative, and all this information can be saved on a pilot's disk. Whether you wish to cheat is up to you, but it's possible to save only the good missions and ignore those that have you killed in action.

In any combat simulator, the quality of the missions is a big factor. *Stealth* has both quality and challenge. Depending upon your choices, a mission may be as simple as flying to the coast of Libya and shooting down a single airplane, or as complex as finding a lone aircraft over the vast waste of the North Cape, or striking well-protected ground targets in Europe during a conventional war.

Sights And Sounds

Whichever you choose, be prepared for excellent 3-D graphics on the ground and in the air. The graphics are of the "wire-frame" type, which has come to be the 64's best combination for representing objects while keeping the speed of screen updating high enough that it does not get boring. During the many hours I've spent with *Stealth*, I've often thought of *F-15 Strike Eagle*. While there are similarities in the games, the difference is in three years of learning new programming techniques, and this difference shows very well in *Stealth*.

Sound has also improved, from the rising whine of a newly started engine and the whistling roar at high altitudes, to the sounds of air and ground explosions.

As always, the documentation from Microprose is as deserving of superlatives as the simulation itself. Because they're now producing games in more computer formats, the manual serves as a general guide, while an insert gives the specifics for your machine. From the manual, you'll learn as much as is known and probably most of what can be surmised as to the eventual nature of the Stealth aircraft. You'll also learn of weapons systems and be given a very good grounding in air combat techniques.

While I've admitted I can't say enough good things about *Project: Stealth Fighter*, I can say this: If you're to have only one flight simulator in your library, let it be this one. On the other hand, if you've tried them all while looking for the best, your search may well end here. On the title screen, below the Microprose logo, are the words *Proudly Presents*. Rightly so.

After all the fun I've had with *F-15 Strike Eagle*, *Silent Service*, and *Gunship*, I now realize that these were merely a prologue. *Project: Stealth Fighter* is Microprose's best.

Project: Stealth Fighter
Microprose
120 Lakefront Dr.
Hunt Valley, MD 21030
$39.95

Guitar Wizard

Art Hunkins

Requirements: Commodore 64, Atari 800/XL/XE, Apple II series, or Macintosh.

Guitar Wizard from Baudville is an educational program designed to teach beginning and intermediate pop-guitarists about chords and scales. It does this by relating chords and scales to the guitar fret board. I reviewed the Commodore 64 version, which comes with an Atari version on the flip side of the disk. The program also is available for Macintosh and Apple II.

The main menu accesses four program options: Chord Wizard, Scale Wizard, Fret-Board Wizard, and Improvization Wizard. The instructional value of the program is augmented by a screen-dump option that prints three or four hi-res screens on a page. This permits hard-copy customization to fit individual needs. You can customize *Guitar Wizard* to default to your choice of several printers.

Fret-Board Choices

Guitar Wizard is easy to use. Available options are indicated on the screen at all times, so even a computer neophyte should experience few, if any, difficulties. Choices are made from menu via cursor controls. A minor problem is that the direction of cursor movement is opposite from what you might expect. The only other inconvenience I encountered is that you have to load the main menu before you can move between programs. This takes time and cuts down somewhat on the attraction of the material's presentation.

Chord Wizard places chords on a displayed fret board. You choose among 17 roots, 32 chord types, and many positions on the fret board. You can display note names, chord degree numbers, or pitches. The number of choices is substantial, and the music terminology used in the program is standard.

Scale Wizard is similar to Chord Wizard, but offers a choice of 19 scales instead of chords, including some that are rarely used.

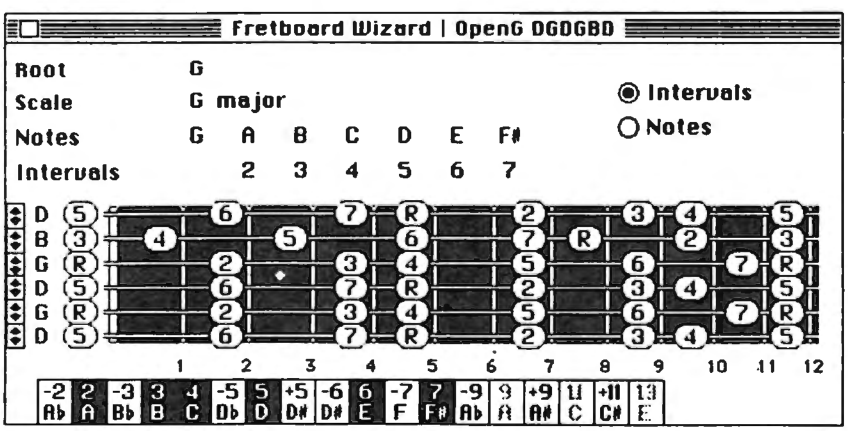
Fret-Board Wizard resembles Scale Wizard, but it allows you to create your own scale or chord by specifying either pitches or intervals. Fret-Board Wizard offers several other fascinating options: You can change instrument tuning, and you can load and save special fret boards of up to six strings. Four special tunings not mentioned in the documentation are on the disk: bass guitar, guitar open-G tuning, open-D tuning, and mandolin. These tunings make *Guitar Wizard* useful for a variety of fretted instruments.

The fourth program is Improvization Wizard. After you select a root and chord type, the computer provides one to nine scales for improvising with that particular chord. In a sense, this program serves to integrate the others—it brings chords and scales together and encourages creativity.

Guitar Wizard includes a tutorial and program instruction booklet, which includes a short primer on fret-board harmony, useful for beginners. In addition to these booklets, the supplemental book, *Guitar Wizard Study Guide*, is available at $12.95.

Two questions came to mind as I evaluated *Guitar Wizard*. First, what is the difference between *Guitar Wizard* and written material dealing with the same subject? My answer is *nothing*, except for the mode of presentation. *Guitar Wizard* is an alternate method of learning scales and chords, perhaps a slightly more interactive method. I see it as a matter of personal preference rather than which method is better.

The second question: Couldn't sound be used to demonstrate different chords and scales? Yes, but there is an obvious problem playing a four-to-six note chord with even the three-voice Commodore 64 SID chip. There also is the problem of differing sound capabili-



ties of various computers. Nonetheless, these difficulties could be overcome by playing four-to-six voice chords in arpeggio, one note at a time, like a slow guitar strum. I think the incorporation of selected play options would enhance the educational value of *Guitar Wizard* significantly. Music is sound, and sound always makes dusty theory become relevent.

Guitar Wizard is a well-produced software package. I recommend it to beginning and intermediate pop-guitarists who wish to cover chords and scales in a comprehensive manner, and who are looking for an alternative to written methods.

Guitar Wizard
Baudville
1001 Medical Park Dr., S.E.
Grand Rapids, MI 49506
$24.95 Commodore 64/Atari version
$29.95 Apple II–series version
$34.95 Macintosh version

Boulderdash Construction Kit

Rhett Anderson

Requirements: Atari, Commodore 64, IBM PC and compatibles, Apple II series, and Atari ST. Atari, ST, and 64 versions require joystick(s). PC version requires CGA. ST version requires color monitor.

I remember the original *Boulderdash.* For a month after I bought it, my computer forgot how to process words. It neglected to calculate my budget, and it flatly refused to run any other games. The first *Boulderdash* was among the most addictive of computer games.

With the release of the *Boulderdash Construction Kit,* Rockford, the star of *Boulderdash,* leaps once more onto the stage. This time, he'll battle the slime and the killer butterflies on all the major home computers, with the regrettable exception of the Amiga.

Boulderdash was originally available from First Star Software—a company started by Fernando Herrara, the winner of the first Atari Star programming contest. The game was programmed by Peter Liepa with Chris Grey. The various versions of the new *Boulderdash Construction Kit* have been created by different programmers, but they're all faithful to the original.

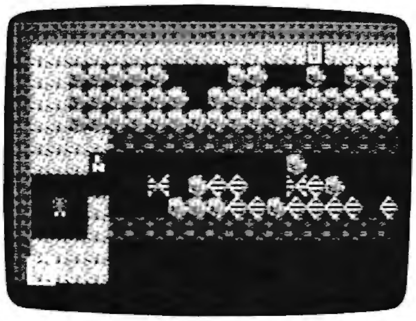
Rockford's Environment

Boulderdash is set in a diamond mine. This is no ordinary mine, though. A variety of deadly inhabitants circulate through its tunnels.

Each game screen is four times the size of the actual computer screen. The background scrolls behind you as you run across the screen.

Each game screen is unique. Some screens require dexterity, some require patience, some require planning, and some require a little of all three. Variety is what makes *Boulderdash* a great game. Being able to design your own screens makes *Boulderdash Construction Kit* even better.

With the game's editor, you design your own screens and games. The editor is easy to use. It's similar to the paint programs found on the Macintosh, ST, and Amiga. You select a brush (like a boulder, butterfly, firefly, or slime) and draw with it. There's even an option that lets you test the screen you're working on. The editor lets you control all aspects of the game, from the number of diamonds that need to be gathered before exiting, to the length of time allotted. When you have the screen just the way you like it, you can save it to disk.



Boulderdash Construction Kit *lets you design your own challenging mine mazes.*

Not For Beginners, But. . .

Boulderdash Construction Kit comes with one sample game that consists of 15 different challenges. This game is much more difficult than the original. Players unfamiliar with *Boulderdash* probably won't stand much of a chance; experts will be delighted. Beginners are better off designing their own games with easier challenges before trying to tackle the sample game.

Boulderdash Construction Kit is a new release in the new MAXX OUT! software series from Epyx. I'm not sure what this means, but if future games in this series are as good, we're all in for a treat.

Boulderdash Construction Kit
Epyx
600 Galveston Dr.
P.O.Box 8020
Redwood City, CA 94063
$24.95

David D. Thornburg, Associate Editor

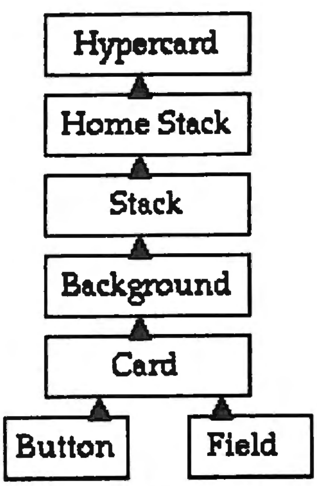
The Power of *HyperCard*, Part 2

Last month we started our exploration of *HyperCard*, the new programming language being shipped with Macintosh computers. This month I want to lift the hood and peer inside this product since it embodies many of the features I think should be a normal part of high-level programming languages. As I said last month, *HyperCard*-like products will probably appear on other computers soon, so if you don't have a Macintosh you may still find something of interest here this month.

HyperCard And Hypertext

HyperCard's name is derived from the word *hypertext*, a concept by which any word or object in one document can be linked to another word or object in another document. Suppose you are reading a biography of the poet Shelley and you come across Lord Byron's name. In a hypertext environment, you could click your mouse on Byron and find yourself in his biography, from which you might then click on the name of his daughter, Ada, which would take you to a document on her and her connection with Babbage and the invention of the digital computer.

Figure 1



Last month's sample program used this technique to link various parts of a fish's anatomy to cards that described each part of the fish in more detail. If *HyperCard* only performed this type of linking function, it would be a valuable product. In fact, *HyperCard*'s underlying metaphor goes far beyond hypertext. To make this point, this month's column will *not* use any hypertext features of the product. Instead we will explore *HyperCard* as an example of an object-oriented programming language.

Object-Oriented Programming

Object-oriented programming is quite different from traditional programming. Programs in languages like Forth, C, LISP, or BASIC consist of documents built out of subroutines or procedures that are (usually) accompanied by a main procedure that is executed when the program is run. The main procedure is always in control, and it uses the subroutines as they are needed.

In object-oriented programming, the user constructs a program by defining classes and instances of objects (we will see some of these later), each of which has the capacity to perform a task in response to messages sent by other objects or by the user. The program consists of a collection of definitions for the various message handlers of each object. Each of these definitions is called a *script* and, in our case, scripts are written in a high-level English-like language called *HyperTalk*.

If this all sounds like computational anarchy and gobbledygook, stay with me. I think you'll see the power and beauty of these languages once we're done.

HyperCard's Objects

HyperCard's objects are built-in. They are based on the metaphor of a stack of index cards blended with some traditional Macintosh objects, like buttons and text fields. The five objects under the user's control are the stack of cards itself, the background image for a series of cards, individual cards, buttons, and text fields. These objects exist in a hierarchy (Figure 1) that includes a special stack called the Home stack and *HyperCard* itself.

To see how the hierarchy works, suppose we are looking at a card with a button on it. When we click the mouse and let go of the mouse button, the message *mouseUp* is sent to the button. If the button contains a handler that does something when this button is clicked, *HyperCard* then executes the handler's script and waits for the next message. If the button doesn't have a script for this mouseUp, the message is passed to the underlying card, then to the stack, then to the home stack, and finally to *HyperCard* itself until it finds a handler that does something with the message. This automatic passing of messages up a hierarchical chain of objects is an interesting feature of *HyperCard*.

Messages

Almost any event can be a message, and *HyperCard* comes with the ability to detect lots of messages: the opening of a card, the entry of the mouse inside a button, the deletion of a card, the entry of the cursor into a text field, the dialing of a telephone, and so on. Furthermore, the user can define new messages and extend the communicative ability of *HyperCard*'s objects.

Why Bother?

At this point, you might be wondering why anyone would go to such extremes to define a new way of creating programs. The proof, as they say, is in the pudding. To illustrate the ease with which a real Macintosh application can be created, I've designed a program that takes a text-based document and codes its message using the following rule: Each letter in the message is followed by a letter chosen at random from the message. For example, the word *this* might be turned into *tihsihst*. (Obvously, this gets more interesting for longer messages.) The program will also have the capacity to decode messages coded in this manner. The user interface for this program consists of a scrolling text window and two buttons labeled Encode and Decode.

If you have ever created Macintosh programs from scratch, you know that a program can take quite a while to write. Using *HyperCard*, I spent only about 20 minutes to bring the program to the level you will see here.

Writing The Code

The first step in creating the application was to define a new background. (Note, since this stack only has one card in it, the *HyperCard* terminology may sound a bit funny here.) Starting with a completely blank screen, I painted the background gray with one of the graphics tools. Next I created a text field, complete with a scroll bar. *HyperCard* lets you choose any of a number of text field designs and lets you choose the size, font, style, and layout of the text in each field. *HyperCard*'s button tools were then used to define the Encode and the Decode buttons.

At this stage of the process (which took about five minutes), we defined the visual appearance of our program (Figure 2). And, even at this stage, the text field supports the ability to accept and edit text just like a word processor.

So far the buttons don't do anything when they are clicked on. To fix this problem, we need to create two scripts, one for each button. The *encode* script looks for the mouseUp message and then encodes the text in the text field. The *decode* script looks for the same message and then decodes the text in the text field. The scripts for the mouseUp message for each button are shown in Figures 3 and 4.

If you have done much programming at all, you will probably be able to read these scripts with ease. For example, the fifth line of the *encode* button's script instructs *HyperCard* to fetch any character (at random) from the text field and concatenate it with the character whose position in the field is stored in the local variable Number. The next line puts this two-character pair into the text field in place of the original character. This process is repeated for all the characters in the text field.

The decode script simply deletes every other character in the text field. (Note to programmers: The reason 1 is added to Number instead of 2 is because once a character is deleted, the rest of the text moves to the left by one character, thus putting the index just one character from its next destination.)

The result of this program's encryption process is shown in Figure 5.

Figure 2

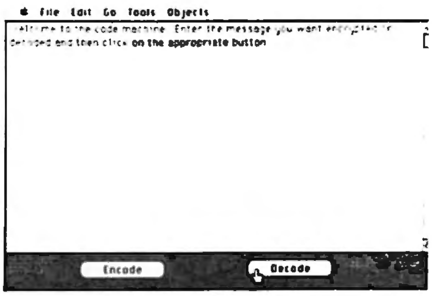


Figure 3

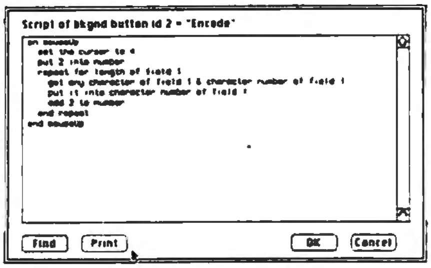


Figure 4

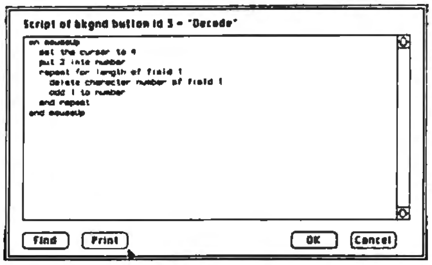
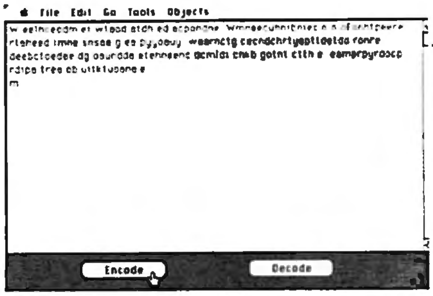


Figure 5



While this brief program isn't going to revolutionize the way you use your Macintosh, it serves to illustrate an important point. *HyperCard* is far more than a hypertext tool; it is a complete programming environment that can be used to build a wide range of computer programs.

Pros And Cons

One of the beauties of *HyperCard*'s object-oriented programming environment is that it makes debugging very easy. By distributing the program among the various objects in the workspace, the user can quickly find the program segment that needs work. Another advantage is that *HyperCard* programs can be easily extended to offer new features without worrying that these new features might interact with existing code. For example, a new button can be added whose mouseUp script loads or saves the text field's contents into a file that can then be read with your word processor.

One feature that I'd like to see is the ability to find all the scripts in a stack and to print them out. At this point you have to print them out object by object. However, since HyperTalk lets you examine scripts, I'm sure that someone will create a button to perform this task.

A major question concerning *HyperCard* is whether it will turn us into a nation of programmers. My feeling at this time is that *HyperCard* lowers the barrier to creating applications for the Mac by quite a bit, but it still requires the discipline and planning required for any programming task. My guess is that most *HyperCard* users will start with existing applications created by others and that they will then modify them to meet their own needs.

In many ways *HyperCard* suggests that the personal computer revolution has just begun in earnest.

Dr. Thornburg welcomes letters from readers and can be reached at P.O. Box 1317, Los Altos, CA 94023. ©

The World Inside the Computer

Fred D'Ignazio, Associate Editor

Skiing Down The *HyperCard* Iceberg

A new toy had just arrived, and I was overjoyed. I threw open the front door and let the UPS man into the house. In he came, lugging three hefty boxes, each with a little rainbow-colored Apple on the corner.

After the UPS man left, I grabbed a giant butcher knife and attacked the boxes. Fifteen minutes later, in my tiny study just off the kitchen, was a platinum-colored Macintosh SE running Apple's snazziest new software hotrod, *HyperCard*.

I fiddled and fussed with *HyperCard* for the next three days—hopping from card to card like Robin Hood cavorting from stone to stone across a forest creek. Tigers and babies popped on the screen and surprised me; and I marvelled at all the prerecorded example cards that were there for me to borrow and personalize for my own tasks. Now I could really get my life in order. Using *HyperCard*, I could throw a dinner party and know what wine to serve, or I could leave my babysitter a personalized file card on the computer screen. I could plot my family tree or print a fancy menu for our family dinner. It was so neat working in an environment where the objects on the screen looked like things from the real world.

A Joker In A Box

At the end of the three days I proudly called my wife, Janet, and daughter, Catie, in for a demonstration of my first customized "stack" of hypercards. As they looked at the screen I gazed into their eyes, sure that I'd see the appropriate awe and respect. But all I saw was bewilderment. They tried to be kind: "Gee, Dad," said Catie, struggling for something nice to say, "I like the picture of the joker in a box." "And the little man wearing sunglasses," remarked Janet. "What a nice touch."

As the two of them left my study, I felt strangely let down. Something was missing. Here I was, a grown man, playing with a multi-thousand-dollar computer, staying up late three nights in a row, and all I'd come up with was a joker in a box and a little man wearing sunglasses.

Peeking At The Iceberg's Tip

Then it hit me. No wonder Catie and Janet hadn't been impressed. What looked like an ice cube to them was really an iceberg. But most of it was hidden. Looking at the computer screen was like peeking at the iceberg's tip.

HyperCard is built like an iceberg. It has five levels—each of which is successively more powerful and more elaborate. The "tip" of *HyperCard* is the Browsing level, where you can point the mouse at different buttons and jump across cards like an eight-year-old on a pogo ball. At the next level down, Typing, you can visit a card and enter in your own information. Here, ready-made *HyperCard* applications, like Rolodex cards and a desk calendar, are available to you. One more level down, and you are at Painting, where you can scribble with a powerful new *MacPaint*. Now the fun begins. If you journey still deeper into *HyperCard* to the Authoring level, you can make copies of sample hypercards and create new applications. Still deeper, and you arrive at the base of the *HyperCard* "iceberg"—the Scripting level. At this level, new passageways and rooms open to you, and you can use the HyperTalk language to write your own programs.

Blazing New Trails

As big as *HyperCard* is, it is really just the tip of an even bigger iceberg—a vast, free-floating iceberg known as *memex*. Memex (short for memory expander) was first conceived by Vannevar Bush in an article entitled, "As We May Think," which appeared in the July 1945 issue of *Atlantic Monthly*. Bush had been director of the U. S. Office of Science and Research during World War II. Now he challenged scientists to stop creating inventions for war and turn their genius to a new kind of computer. Memex would be a personal computer, a guide that would help an individual blaze a trail through the mountainous stacks of human knowledge. Bush felt that Memex should have unlimited storage and lightning-fast retrieval. But it needed something even more important: the ability to connect ideas, facts, and information by association, just like the human mind.

Now, more than 40 years later, memex is still nothing more than a dream. But it's a dream which has spawned new dreams by some of our most brilliant computer scientists. It is Alan Kay's *Dynabook*; Andy Van Dam's *Electronic Book*; Ted Nelson's *HyperText*; and now, Bill Atkinson's *HyperCard*.

Most of the memex iceberg is still waiting to be discovered. So I'm going to say goodbye to my joker and my little man with glasses, and schuss boom my way into *HyperCard*. Who knows where I'll turn up next? ©

Microscope

Sheldon Leemon

High-speed 24-pin dot-matrix printers have already had a significant impact on the daisywheel printer market. Using 24 fine wires instead of the standard 9 pins, they print fully formed characters which look like they came from a typewriter at speeds of up to 70 or 80 characters per second. They're also capable of high-density graphics printing—almost a necessity for producing graphs and charts.

Daisywheel printers never get much higher than 55 characters per second, can't print any graphics, and are about 50 percent more expensive than the 24-pin printers. As a result, a recent PC magazine comparison which included hundreds of printers could only find a handful of daisywheel printers to test.

The price advantage of 9-pin printers has steadily eroded, as 24-pin printers have steadily dropped in price—from $1,500 to $1,000, and now finally to the under-$500 price range. NEC was the first to come out with a 24-pin printer for under $500, when it introduced the P2200 a few months ago. Printer giant Epson, not to be outdone, introduced the $495 LQ-500 shortly thereafter.

Not only are these printers priced right, they have features galore. The P2200 comes with six standard fonts, two slots for additional font cartridges, and special paper-handling capabilities for both single sheets and tractor paper, including an optional cut-sheet feeder. With prices and features like this, the days of dotty-looking term papers may soon be just a memory.

****

The quest to clone IBM's PS/2 line of computers seems to be right on course. Phoenix, the company which produced the compatibility software that launched a thousand clones, has announced BIOS software for all PS/2 models. To prove the point, Phoenix showed actual IBM computers running with their operating system software. Meanwhile, on the hardware side, Western Digital has joined Chips and Technology in announcing chip sets for PS/2 look-alikes and a clone of the Micro Channel bus.

Meanwhile, IBM's plans for the new OS/2 multitasking Operating System are a little ahead of schedule. Shipments of the text-based version 1.0 have already started. The graphics-based version 1.1 which includes the Presentation Manager is still not expected until late 1988.

****

Laptops were all the rage at the Fall COMDEX show. One of particular interest came from Amstrad, the British company whose low-priced desktop machines have done well both in Europe and in the U.S. The Amstrad machine weighs under 12 pounds, has a full-size keyboard, a half-size supertwist LCD screen, a built-in modem, and 3½-inch drives. A single-drive 512K model will sell for as low as $700, while the two-drive 640K model with modem will cost only $1,100. Unlike most laptops, which come with rechargeable batteries, the Amstrad runs on ten ordinary C cells.

Laptops were not the smallest PCs at COMDEX, however. A few vendors were showing hand-held PCs, the size of pocket calculators. Since they can't include disk drives, software must be downloaded through a serial port or included in ROM. Though not cheap, these little guys take MS-DOS applications where none have gone before. What's next, MS-DOS wristwatches?

****

Atari has unveiled the Abaq, a $5,000 workstation built around the Inmos transputer, a parallel-processing RISC (Reduced Instruction Set Computer) chip. The base unit will come with four megabytes of main memory, one meg of display RAM, an ultra-high resolution 1280 × 960 color display, a super-fast blitter chip that also supports a Local Area Network, and a single processor. Multiple processors can be added to increase the computing power past mainframe levels. The workstation will use an ST computer as an I/O processor to handle keyboard input and disk functions.

The operating system for the Abaq will be Helios, a multitasking, multiuser, Unix-like operating system with a graphics interface, possibly X-Windows. Development of Helios is being headed by Dr. Tim King, formerly of MetaComCo. ST owners may recognize MetaComCo as the British firm responsible for ST BASIC, the much-criticized language which comes packaged with the ST. MetaComCo also created AmigaDOS, the portion of the Amiga Operating System which so irks its users that a grass-roots movement known as the AmigaDOS Replacement Project has emerged to replace it.

****

Although Atari insists that the ST is its flagship line, and Commodore swears that its future depends on the Amiga, it's interesting to note that both companies are broadening their PC-compatible lines. Commodore recently showed the PC-10-3, a new small-footprint 10 MHz turbo PC with built-in EGA adapter, along with the PC-40, an AT clone, and the PC-60, an 80386 model. The slotless Atari PC shown earlier in the year has now become the PC-1, and it is joined by the PC-2, a four-slot model with a bigger power supply; the PC-4, a 5-slot 286 machine with VGA graphics; and the PC-5, a 16-20 MHz 80386 powerhouse. ©

Telecomputing Today

Arlan R. Levitan

The Las Vegas Connection

The first week of November 1987 saw more chips than usual stack up in the casinos of Lost Wages, Nevada, as more than 100,000 attendees descended on the desert mecca for the winter Computer Dealers Exposition (COMDEX). At the show the modem market was as busy, albeit decidedly less volatile (due to battery-backed-up CMOS), than the Dow Jones average.

Novation demonstrated production models of its diminutive Parrot 1200 modem. Although announced earlier in 1987, card-counters had noted that delivery of the audio-cassette-sized modem had been delayed until November by snags in the availability of proprietary VLSI chips that account for the unit's low chip count and list price ($119).

Supra, a well-known supplier of hard disk drives and other accessories for Atari, Commodore, and Apple computers, rolled seven come eleven with the SupraModem 2400, a stand-alone 2400 bps modem tagged at a surprising $179. According to Supra president John Wiley, the new Supra unit will also be available in bundled packages that include all required cables and full-featured telecomputing software for $219.

About a year ago, Avatex was one of the first manufacturers to break the $200 price barrier with a Hayes Compatible 1200 bps external modem. Not content to rest on its laurels, the firm has hit and stands pat with the Avatex 1200e, a smaller (5 × 6 × 1 inch) version of its big brother that carries a suggested list price of only $99.

Super Fast Modems

Hayes Microcomputer did not show any new products, but managed to up the ante in the high-stakes, high-speed game by announcing that CompuServe and Western Union Easylink will offer 9600-bps dial-up access for Hayes V-series Smartmodem 9600 owners. CompuServe will be first to the post, with Atlanta, Boston, Chicago, Columbus (Ohio), Dallas, Detroit, Los Angeles, New York, San Francisco, and Washington D.C. off and running by March 1988.

Western Union will place second, deploying Hayes 9600s in most major U.S. markets by mid-1988. As we go to press, neither commercial service has announced connect-time charges, but reliable sources indicate that current plans call for from two to two-and-a-half times the current 2400 bps rates.

Elsewhere on the 9600 front, U.S. Robotics continues to hold its own with the computer-based bulletin board crowd. According to USR representatives, over a thousand BBS system operators have used the firm's special SYSOP purchase program to acquire Courier 9600 HST modems. Automatic data compression/decompression has been added to the 9600 HST with no change in the unit's suggested list of $995. Other new features include automatic data rate fall-back and step-forward capability, which lowers or raises the modem's speed to 9600, 7200, or 4800 bps in response to changing line conditions. Owners of older Courier 9600s may elect to make the squeeze play for a $50 upgrade charge.

For lower rollers, USR drew three to fill out its budget-priced Sportster line of modems with four-of-a-kind, priced from $139 (300/1200 bps) to $249 (300/1200/2400 bps). Both internal and external Sporsters also come with TelPac, a PC-compatible terminal program. IBM PS/2 owners who want to play the slots can opt for the new Courier 2400e/PS internal modem ($699).

Even Jimmy The Greek was thrown for a loop by the news that U.S. Robotics has settled its lawsuit with Hayes Microcomputer. Early last year, Robotics had named Hayes as a party to a lawsuit against modem manufacturer Bizcomp. While USR appears to be prevailing in that action, Robotics has dropped Hayes from the suit in exchange for an agreement that allows both USR and Hayes to use each other's existing patents and those issued on new inventions over the next three years.

Does this mean that the present confusion over 9600-bps standards may soon come to end? While nothing is in the immediate offing, highly placed sources at both Hayes and U.S. Robotics have intimated that you can make book on the two firms' high-speed modems being on speaking terms in about two years.

Parrot 1200
Novation
21345 Lassen St.
Chatsworth, CA 91311

SupraModem 2400
Supra
1133 Commercial Way
Albany, OR 97321

Avatex 1200e
Elec & Eltek (U.S.A. Corp.)
1230 Oakmead Pkwy., Suite 310
Sunnyvale, CA 94086

Smartmodem 9600
Hayes Microcomputer Products
705 Westech Dr.
Norcross, Georgia 30092

Courier 9600 HST, 2400e/PS
Sportster Modems
U.S. Robotics
8100 N. McCormick Blvd.
Skokie, IL 60076

ST Outlook

Philip I. Nelson

Atari Does Business At COMDEX

It's November as I write this column, which means that it's time for another COMDEX (Computer Dealer's Expo), where microcomputer manufacturers strut their stuff. Atari used this COMDEX to make some strong moves in the direction of the business market.

Megas And Clones

"New" is perhaps not the best word to describe Atari's line of PC-compatible machines, since the Atari community has been hearing and talking about these for many months now, and the basic MS-DOS technology is the stuff of yesterday's headlines. But at least the Atari PC lineup looks reasonably complete, with an 80386-based machine in the works in addition to more conventional clones.

In terms of price and features, Atari's entries in the PC-compatible sweepstakes are middle-of-the road material—not as affordable as the least expensive clones, but less costly than an IBM.

Another not-exactly new product is the Mega ST, which has been getting press for what seems like an eternity. What's newsworthy about these machines is that they're finally on the shelves, complete with a blitter chip for faster graphics and a revised operating system that fixes assorted TOS bugs.

The Mega ST models have a more businesslike configuration than the 520ST and 1040ST, with a detached keyboard and enough room in the CPU box for an internal hard disk drive. Whether or not they're bought by businesses, as Atari is betting, it's certain that the Megas will become the new machine of choice for ST developers and power users.

New From The Batcave

To many mortals, Atari's newly announced Abaq Transputer will sound like something straight out of the Batcave. A RISC-based computing engine with one megabyte of RAM *for screen memory alone,* which can take as many as 13 parallel processors and requires a four-megabyte Mega ST just to communicate with the outside world. Holy Megaflops, Batman—it must be an Abaq!

What's an Abaq good for? The answer, again, is networks. Satisfying the demands of a multiuser network takes a lot more churning power than any desktop computer can provide. What you want here is a big, fat box crammed full of processors and RAM, and that's exactly what the Abaq provides. Compared to what you pay for an ST, the Abaq's $5,000 price tag seems outrageous, but that's a fraction of what you have to pay for other devices of this type.

RISC, by the way, stands for *Reduced Instruction Set Computer,* and with parallel processing, it's one of the three great techie buzzwords of this season. The third one is *MIPS,* which signifies Millions of Instructions Per Second. With those three words alone, you should be able to survive any social situtation involving computer nerds for the next six to nine months. Just remember that a RISC can do many MIPS, and a parallel-processing RISC can do the most MIPS of all. Lard your conversation with these terms, and your listeners will either slump in slack-jawed amazement or beat you to death on the spot.

Whither CD-ROM?

One of the few consumer-oriented products that Atari announced at COMDEX is a double-duty CD-ROM player that can play music CDs and also function as a CD-ROM reader for an ST computer. If the wheels behind the CD-ROM industry have their way, this product will help open the door to a brave, new multimedia world for microcomputers, where the computer is intimately linked with the other household electronics, and every gadget in the house reads tons of video, audio, and text data from CD-ROM disks.

That's a noble vision, all right, and if CDs can obtain some cost-effective read/write capability before they're replaced entirely by some other medium, then it may happen exactly as dreamed. In the meantime, there are exactly two CD-ROM applications for micros—the American Heritage encyclopedia and Microsoft Bookshelf—both for the IBM PC.

More On *pc-ditto*

Not long ago, a reader sent me a long, impassioned letter in defense of *pc-ditto,* the IBM PC emulator that I reviewed in a previous column. The letter-writer asked, among other things, what software I tried with *pc-ditto,* and whether I had taken advantage of the two utility programs in the *pc-ditto* package that speed up the screen display and keyboard repeat rate of the emulated PC.

The software that I tried out was best-selling stuff along the lines of *WordPerfect* (the IBM PC version) and *Microsoft Word,* with some popular shareware and public domain programs thrown in for good measure. In answer to the second question, I did use the two speed-up utilities. While they help a bit, the difference isn't enough to make me want to use *pc-ditto* on a regular basis.

To reiterate my advice from the previous column, the only way to evaluate a piece of software is to get your hands on a copy and give it a thrashing. If you're curious about *pc-ditto,* find a reputable dealer and ask for a chance to try it under real-life circumstances. ©

AmigaView

Sheldon Leemon

New Amiga Products

Although the fall COMDEX show came less than a month after AmiExpo, it brought many surprises and delights for Amiga users. Commodore showed some of the Amiga peripherals we've been speculating about for many months. One of the 2000s at the booth was using an accelerator card, which features a 68020 processor running at 14 MHz, a math coprocessor, an MMU (Memory Management Unit), and two megabytes of 32-bit RAM. It fits into the 86-pin coprocessor slot on the 2000, effectively replacing the Amiga's 68000 processor with the faster and more powerful 32-bit 68020. This allows most software to run three to four times as fast (great for multitasking). Nearly all current software runs under the faster processor without modification, since the Amiga has always been designed with upward compatibility in mind. The optional 68881 coprocessor chip allows math-intensive programs to perform floating-point operations many times faster than normal. In one case, a tester reported that drawing time for a very complex 3-D picture was cut from two days to less than two hours. The math chip is clocked separately and can be run at 7, 14, or 21 MHz (the faster chips perform better, but cost more).

The reason for the optional MMU (memory management unit) is less clear, since Amiga's multitasking operating system doesn't use one. However, a Commodore engineer suggested that it could be used to move the 256K of Kickstart code into the 32-bit memory, allowing for even greater speedups. This chip might also be used for the version of Unix under development at Commodore. This software requires a 2000 with the accelerator card and hard disk, but it is a full version 4.3 Berkeley Unix. Reportedly, it will have a windowing user interface. And, there's a possibility that using the *Mock* software developed at Carnegie-Mellon Institute, the 2000 will be able to run software written for Sun workstations.

Higher Resolution

For workstation applications like CAD and desktop publishing, a greater display resolution is required. Commodore's response to this challenge is the A2024 high resolution black-and-white monitor. This display shows 1008 x 800 pixels using two memory bits per pixel (for four shades of gray). Best of all, it requires no additional hardware and plugs right into the RGB port of any Amiga. The expanded display is managed entirely with modified Kickstart software which is loaded into RAM (running the monitor requires at least one megabyte and possibly more). The software sends out one section of the bitmap at a time through the RGB port, and the monitor assembles these pieces into a unified display. A display rate of 10 complete screens per second makes the monitor unsuitable for animation, but fine for other purposes.

Since the modified display software takes the place of the normal ROM routines and provides an expanded Workbench screen, almost any software that uses the Workbench screen can take advantage of the expanded resolution without modification (Commodore was showing the stock version of the City Desk publishing software running on the monitor). Other software will have to make only minor changes to take advantage of the new display. Perhaps the biggest change will be the need to adjust the size of the character fonts. Using the system display font on a 1008 × 800 display, you get 100 rows of 126 tiny characters each. Even software that doesn't make these changes can benefit from this display, however, since it can also show normal 640 × 400 Amiga screens without the flicker. The monitor hardware automatically deinterlaces the display, adding a new dimension to existing CAD and publishing software.

Commodore also showed two video products, both for the 2000. The first is an internal Genlock card, which goes in the video slot. The card overlays Amiga graphics on live video, producing a broadcast-quality combined video signal. The other product is the Professional Video Adapter, a combination Genlock and realtime digitizer that I've described in previous columns. The base unit will take up at least two slots, the video slot and one of the 100-pin slots. The optional field-store board will take up another 100-pin slot. This unit provides true RS-170A video and features software control over brightness, contrast, color, and hue of both the incoming video signal and the overlay. Commodore stated it would release a software interface guide to developers.

When? And How Much?

Many of the products that were shown are within six months of shipment. Among these are an 80286 version of the Bridge card (for AT compatibility), the A2058 8-meg memory board, the A2080 hi-persistence phosphor monitor, and the Genlock for the 2000. The A2024 hi-res monitor, the 68020 accelerator card, and PVA shouldn't be very far behind. Cost is less certain. The PVA should be around $800, and the accelerator should be under $2,000. The hi-res monitor may be anywhere from $500–$700; the AT bridge card, about $1,000.

Next month, we'll talk about third-party products at COMDEX. ©

IBM Personal Computing

Donald B. Trivette

Pop Quiz

Have you been reading this column closely for the past few months? Good, because you're in for a surprise. Today there's going to be a pop quiz, courtesy of *Crossword Power,* a crossword-puzzle generator for the IBM PC and PCjr.

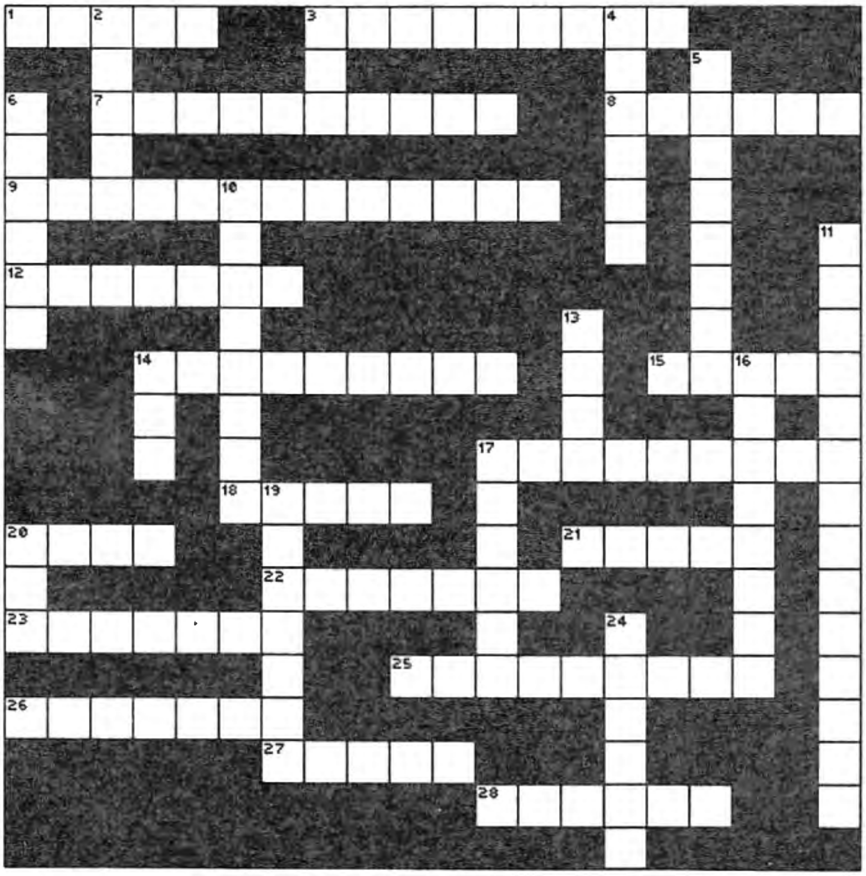
I went through six months of my previous COMPUTE! columns and selected key words, software names, computer terminology, and general trivia that I hoped would challenge you. Then, I entered the answers along with the definitions in *Crossword Power* and told it to generate as many 20 x 20 puzzles as it could find. After it constructed about 50 (each puzzle takes less than four seconds to appear on the screen), I pressed the Esc key and looked at the best ones—the ones that used the most words. The puzzle below uses 32 of the 35 words in my word list and was the twenty-seventh puzzle generated.

With the puzzle completed, I used the menu to tell the program to print the grid, the definitions, and the answer, which will appear in a future column. *Crossword Power* supports eight printers: Epson MX, Epson RX, Epson FX (special and regular), Brother HR-15S, Brother HR-15#, Okidata, and a user-defined printer. I was unable to get my NEC 3550 to print the puzzle, even though the manual said the program would work with letter-quality printers. I user-defined the NEC, but that made no difference, so I borrowed an Epson FX, which printed the puzzle without problems.

Crossword Power is not very professional in some areas: The menu has some rough edges and the manual is not the easiest to understand, but with a little experimenting, you'll have it working. Even though it was designed for teachers to create vocabulary drills, it's a great product for puzzle fans.

Send your completed puzzle to IBM Crossword, c/o COMPUTE! Magazine, P.O. Box 5406, Greensboro, NC, 27403. I'll publish the names of the first three winners based on the earliest postmarks.

Crossword Power requires 256K of memory, one disk drive, DOS 2.0 or later, and a supported printer. It's available for $69 plus $2 shipping from WISCO Computing, 2821 Sampson St., Wisconsin Rapids, WI 54494.



Across

1. Popular database program
3. Year after the freshman year
7. The Lascaux1000 is one
8. Grandslam _______ is a game
9. Type of menu selection
12. Denver basketball team
14. 201, 202, and 212 are examples (2 words)
15. The computer's telephone
17. Pen name for Samuel Clemens (2 words)
18. A black suit
20. The IBM PC's little brother
21. A musical instrument
22. Denver football team
23. Name of a famous numerical constant
25. Used in playing computer games
26. Telephone dialer recently reviewed
27. Rolls _______
28. The queen of _______

Down

2. Common way to represent characters
3. All Sass _______
4. You get 700 points for this in bridge
5. Number of cards in a deck
6. _______ Instructor II is an educational program
10. Name of a .BAT file
11. Type of software used to dial other computers
13. Number of players in a bridge game
14. Higher than a king
16. One of the four suits
17. Place where computers store information
19. The _______ Sea Adventure is part of Typing Instructor
20. Something you can do to corn
24. Developer of King's Quest

Bill Wilkinson

Adding Power To BASIC

Last month we took a look at data types and how they're used successfully and profitably in computer languages such as Pascal. There are real and discernible advantages to using a language that handles structured data, and I hope I convinced you of that. Of course, most of those languages offer other significant reasons to use them, such as faster execution speed. Still, none of them do one thing as well as good old BASIC does. The interpretive environment of BASIC makes program development exceptionally easy.

When I travel to user-group meetings and show off one or the other of the OSS advanced BASICs, I inevitably write a program that I make up on the spot, in the meeting. And I usually manage to convert a few nonprogrammers into at least thinking about taking up BASIC as a hobby. I'm not sure I could do that with most compiler environments. So, like it or not, I do understand why most people want to learn to program in BASIC first. (And did you notice that I didn't even mention the usual reason? BASIC comes with the machine, so you don't have to pay extra to use it.)

So, given that most of you, my readers, will program in BASIC, the least I can do is show you some techniques to make such programming easier. To me, that implies showing you how to use techniques from other languages in BASIC. In turn, that means learning some tricks that will make BASIC more powerful.

Sorting Things Out

Type in and try Program 1. When you run it, give it any numbers you like, including perhaps several occurrences of the same value. When you finally enter a 0 value, the program will print out the list of your numbers in sorted order. Congratulations—you've just used a technique known as an *insertion sort.*

The name makes sense, doesn't it? As each new number is entered (line 25), we find where it belongs (that is, after which current number; lines 35 and 40), and then "insert" it into the appropriate spot in the list of numbers (lines 45 through 60). In some situations, this is a pretty good sorting method. For example, when you have to wait several seconds between user input, what's a quarter second or so to insert a number? A lot of the efficiency of an insertion sort depends on the speed with which the actual insertion is made. In this little BASIC program, we used a FOR/NEXT loop (lines 45 through 55) to do the insertion. (Note that this would be way too slow if we were trying to do a couple of thousand insertions.) Luckily, in most languages, there are faster methods. But, in any case, the sorting method is not the important part of this month's discussion.

Now suppose, that instead of inserting a single number (as we were doing here), we were working with an entire set of information. Consider a typical mailing list, where we would be shuffling around a name, address, city, state, zip code, phone number, and various other bits and pieces. Can we, using BASIC, manipulate this information as easily as we sorted those numbers? Not quite, but we can come close.

Setting The Record

Take a look at this example of a Pascal record as I presented it last month:

```
TYPE
   Cust_Rec = RECORD
      Name : String[30];
      Addr : String[30];
      City : String[15];
      State : String[2];
      Zip : 0. .99999;
      Credit : ( OK,Avg,Bad );
      END;
VAR
   Mail_List : ARRAY [1. .100]
                      OF Cust_Rec ;
```

Our variable **Mail_List** is an array of records, and each record holds several pieces of information about a given customer. Using the information in these records is almost easy. For example, we could find the zip code of customer number 17 by simply coding

```
Write( Mail_List[ 17 ].Zip )
```

The conversion from a number-sorting program to a record-sorting program is almost a trivial exercise in Pascal. While we can't duplicate the feat as easily in BASIC, we can at least simulate this convenient grouping of related pieces of information into a record. Again, look at Program 2. This is actually the same program as Program 1, but it uses strings to simulate records. If you look at the code from line 300 to line 410, you should be able to find a direct correlation to the statements of lines 30 to 60 of the first listing. True, the lack of string arrays in Atari BASIC has forced us to use some pretty strange looking assignment statements because we are now moving around substrings instead of simple numbers. I've tried to make these movements as clear as possible, but don't feel bad if it takes you some time to understand what is going on. I encourage you to print out the various strings (such as MAILLIST$ and RECORD$) at several points to see what is happening.

You may have noticed that these records are sorted based on the name of the person. Try this puzzle before reading on: Can you suggest ways of insuring that the sort is by zip code, instead?

And, if you have ST BASIC, Atari Microsoft BASIC, or OSS's BASIC XL or BASIC XE, you might try converting this program to use string arrays. I think you'll find that the only real savings in coding complexity occurs in the actual insertion loop (lines 300 to 410). For the rest of the program, good old Atari BASIC doesn't suffer too much in comparison.

What have we accomplished? I hope you can see how, by isolating the record build/retrieve in subroutines such as those at lines 800 and 900 in this example, pseudo records are quite possible in BASIC. But we have also seen that the manipulation of these records can be tedious. And certainly moving all that string data around is not the fastest set of operations in the world. How could we improve things? Time to borrow some more concepts from structured languages such as Pascal: pointers and linked lists. But, for a look at those topics, we'll have to wait until next month.

Thought I forgot the answer to my little puzzle? Nope. Two ways to sort by zip code: Rearrange the order of the data in the RECORD$ string so that the zip code comes first; or, change the master record comparison in line 340 so that only the zip code portions of the strings are compared. For example:

```
340 IF RECORD$(78,82) > MAILLIST
    $(RECPTR + 78,RECPTR + 82)
    THEN NEXT RECNUM:STOP
```

Program 1: Simple Numeric Insertion Sort

```
CO 10 DIM NPOS(20)
LK 15 FOR TOP=0 TO 19
EA 20 PRINT "GIVE ME A NUMBER BIGGER THAN 0 ";
IA 25 INPUT NPOS:IF NPOS<=0 THEN 80
JO 30 IF TOP=0 THEN CHK=1:GOTO 60
CJ 35 FOR CHK=1 TO TOP
PF 40 IF NPOS>NPOS(CHK) THEN NEXT CHK:GOTO 60
JC 45 FOR MV=TOP+1 TO CHK STEP -1
GI 50 NPOS(MV)=NPOS(MV-1)
EM 55 NEXT MV
EK 60 NPOS(CHK)=NPOS
JN 65 NEXT TOP
ME 70 REM IF 20 NUMBERS, FALL THROUGH
FE 80 REM TO HERE WHEN 0 OR LESS ENTERED
DN 85 FOR CNT=1 TO TOP
GM 90 PRINT NPOS(CNT)
JC 95 NEXT CNT
```

Program 2: Insertion Sort of Pseudo-Records

```
AI 100 REM DATA DECLARATIONS
NE 110 DIM NAME$(30),ADDR$(30),CITY$(15)
OL 120 DIM STATE$(2),ZIP$(5),CREDIT$(1)
EJ 130 RECSIZE=30+30+15+2+5+1:MAXREC=100
LI 140 DIM RECORD$(RECSIZE)
EO 150 DIM MAILLIST$(RECSIZE*MAXREC)
MH 160 DIM SPACE$(RECSIZE),YESNO$(1)
FK 170 SPACE$=" ":SPACE$(RECSIZE)=" "
LK 180 SPACE$(2,RECSIZE)=SPACE$
FO 190 MAILLIST$=SPACE$:TOPREC=0
CC 200 REM DATA ENTRY
BC 210 FOR TOPREC=TOPREC TO MAXREC-1
KE 220 GRAPHICS 0:PRINT TOPREC;" CUSTOMERS IN FILE"
MJ 230 PRINT "ENTER ANOTHER CUSTOMER (Y/N) ";
OG 240 INPUT YESNO$:IF YESNO$="N" THEN 500
AF 250 GRAPHICS 0:GOSUB 700:REM ENTER A RECORD
NO 260 GRAPHICS 0:GOSUB 600:REM SHOW THAT SAME RECORD
HE 270 PRINT :PRINT "IS THIS OKAY";
DE 280 INPUT YESNO$:IF YESNO$<>"Y" THEN 250
BO 290 GOSUB 900:REM CONVERT TO RECORD FORMAT
PH 300 REM FIND INSERT POINT
EI 310 IF TOPREC=0 THEN RPTR=0:GOTO 400
DA 320 FOR CHK=1 TO TOPREC
NP 330 RPTR=(CHK-1)*RECSIZE
AF 340 IF RECORD$>MAILLIST$(RPTR+1,RPTR+RECSIZE) THEN NEXT CHK:RPTR=RPTR+RECSIZE:GOTO 400
BA 350 REM INSERT RECORD
DP 360 FOR R=TOPREC TO CHK STEP -1
FI 370 TEMP2=R*RECSIZE:TEMP1=TEMP2-RECSIZE
NL 380 MAILLIST$(TEMP2+1,TEMP2+RECSIZE)=MAILLIST$(TEMP1+1)
CN 390 NEXT R
DA 400 MAILLIST$(RPTR+1,RPTR+RECSIZE)=RECORD$
KB 410 NEXT TOPREC
HJ 500 REM
KA 510 REM DONE
ID 520 FOR RECNUM=0 TO TOPREC-1
CG 530 RPTR=RECNUM*RECSIZE
DB 540 RECORD$=MAILLIST$(RPTR+1)
LN 550 GRAPHICS 0:GOSUB 800:GOSUB 600
BF 560 PRINT :PRINT "HIT RETURN TO SHOW NEXT RECORD";
NO 570 INPUT YESNO$
KG 580 NEXT RECNUM
GJ 590 GOTO 200
IK 600 REM SUBROUTINE
LM 610 REM SHOW A RECORD
HO 620 PRINT "NAME ::";NAME$
HH 625 PRINT "ADDR ::";ADDR$
KP 630 PRINT "CITY ::";CITY$
EE 635 PRINT "STATE::";STATE$
CE 640 PRINT "ZIP  ::";ZIP$
PJ 645 PRINT "CREDIT RATING (A TO F) ::";CREDIT$
HP 690 RETURN
IL 700 REM SUBROUTINE
AM 710 REM INPUT A RECORD
NJ 720 PRINT "NAME >";:INPUT #16,NAME$
MC 725 PRINT "ADDR >";:INPUT #16,ADDR$
PK 730 PRINT "CITY >";:INPUT #16,CITY$
IP 735 PRINT "STATE>";:INPUT #16,STATE$
EP 740 PRINT "ZIP  >";:INPUT #16,ZIP$
OF 745 PRINT "CREDIT RATING (A TO F) >";
MB 750 INPUT #16,CREDIT$
IA 790 RETURN
IM 800 REM SUBROUTINE
BK 810 REM TAKE APART A RECORD
AE 825 NAME$=RECORD$
KP 830 ADDR$=RECORD$(31)
MF 835 CITY$=RECORD$(61)
MP 840 STATE$=RECORD$(76)
JI 845 ZIP$=RECORD$(78)
FI 850 CREDIT$=RECORD$(83)
IB 890 RETURN
IN 900 REM SUBROUTINE
OO 910 REM BUILD A RECORD
EL 920 RECORD$=SPACE$
BG 925 RECORD$(1,30)=NAME$
EC 930 RECORD$(31,60)=ADDR$
GO 935 RECORD$(61,75)=CITY$
LK 940 RECORD$(76,77)=STATE$
CP 945 RECORD$(78,82)=ZIP$
PA 950 RECORD$(83,83)=CREDIT$
IC 990 RETURN
```

©

CAPUTE!

INSIGHT: Atari

The code that appears in the October 1987 "INSIGHT: Atari" column is correct as listed, with one minor change. Just before the last line (.END or END), the variable SNAME needs to be declared. The proper declaration is

```
SNAME .BYTE "S:"
```

Amiga Marbles

This program, from the October 1987 issue, is correct as listed, but it needs the graphics.bmap file on your Extras disk. If you are missing this file, the 1.2 Extras disk contains a program called ConvertFD which will create it for you. Run the program and enter **Extras:fd-1.2/graphics_lib.fd** for the file to convert. Enter **graphics.bmap** for the output file. When you run Marbles, this file must either be copied to the current directory or the LIBS directory on your boot disk; otherwise, Amiga Basic will stop with a *file not found* error. ©

The Beginners Page

C. Regena

Typing In BASIC Programs

First, a correction. In my discussion of the random function (COMPUTE!, November 1987), I introduced a formula that simulates the rolling of two dice. This formula—INT(11*RND(1)) + 2—is correct if you want each number from 2 to 12 to have an equal chance of being chosen. However, as anyone who has played games of chance knows, the odds of getting 2 (two 1s) is not the same as getting a total of 7 (1 and 6, 2 and 5, or 3 and 4). To simulate two dice correctly, we must generate two random numbers between 1 and 6, and add them together like so: INT(6*RND (1)) + 1 + INT(6*RND(1)) + 1

Last month I celebrated an anniversary of sorts—five years of writing articles and columns for COMPUTE! Publications, and seven years of programming home computers. Most of my articles have contained program listings—many beginners learn to program by typing in listings from books and magazines. Not only can you build up a library of programs rather inexpensively, you can learn many programming styles and techniques from other authors. If you study each line of code as you type it in, you can learn how to get the computer to do certain things. This month I'm going to give you a few hints on typing in programs from printed listings.

Common Mistakes

Watch carefully for characters that look the same, such as the number 1 and the lowercase letter *l*, the number 0 and the capital letter *O*, or the number 8 and the capital letter *B*. Usually, you can tell by context what the symbol should be, but elusive errors are often caused by mistyped characters. As a programmer, I avoid using the letter *O* and the lowercase *l* as variable names.

Be careful when typing in lines with similar code. Check the line numbers as you type. If there are sections of lines that look alike, you might accidentally skip part of the program. Some people like to keep a card or line-guide under the line being entered. This way, you know exactly where you are.

Judging from my mail, the most common error occurs in DATA statements. Computer programming is exacting, and you have to get every single comma and number exactly right. You may see commas together in a DATA statement with nothing between them. This is a fairly common way of indicating a null string, and those commas are absolutely necessary.

Finally, you must be sure to press RETURN or ENTER at the end of each line. Once in a while you'll encounter a line that has exactly the same number of characters as the width of your screen. When entered, the cursor goes to the next line just as if you pressed RETURN, even though you didn't. If you do not press RETURN, the line is not properly entered.

This has nothing to do with avoiding typing errors, but you should try to SAVE your program every 20 minutes or so. You never know when a power glitch will occur, and it's better to pause every so often to save your work rather than take the chance of losing hours of typing. I also recommend using two separate disks or cassettes to save the program. Otherwise, if a power failure occured during the saving process, it would ruin your one and only copy.

Special Listings

Learn the particular conventions of the magazine or book from which you're typing. For example, COMPUTE! listings use braces { } to indicate a number of spaces, a number of symbols, or a special keypress. You do not actually type the brace symbol. For example, on Atari eight-bit computers {CLEAR} means to type ESC SHIFT <. On Commodore 64/128 listings, you may see {CLR}, which means to press SHIFT CLR/HOME. Almost all magazines have a page such as "COMPUTE!'s Guide To Typing In Programs," which explains how to type in the program listings.

Many magazines use checksums to help you enter programs more accurately. Different magazines use different checksum programs, but the basic idea is that you type in a line, compare a number shown on the screen with one printed in the listing, and if the two numbers differ, you have made a typing mistake. Checksum programs are not foolproof, but they do catch most errors.

Debugging

Most of your problems will occur when you actually run a typed-in program. The process of correcting these problems is called *debugging*. Some say that debugging accounts for 90 percent of a program's development.

Before you run a program, be sure to save a copy to tape or disk first. If a program contains an error in a POKE statement, for example, it could "lock up" the computer, causing you to lose the program in memory.

When an error occurs, the computer usually gives the line number of the incorrect line. The most common type of error is a *syntax error*. A syntax error usually means a spelling error or an incorrect use of a BASIC command or function.

There are certain things that you should look for in lines that produce errors. Make sure that all the BASIC words are spelled correctly. Count the parentheses to make sure there are even pairs: For every left parenthesis there must be a corresponding right parenthesis. Check commas to make sure you have the right number of parameters. For example, a CIRCLE command may require numbers for the X and Y coordinates of the center, the radius, and other attributes.

Another cause of errors is trying to use numbers that are out of range. For example, if your program prints at a certain row and column, you have to make sure the row and column specified are within the boundaries of your screen (you cannot have a column value of 50 if your computer has a 40-column screen). Any time the computer stops with an error message, you can PRINT the value of variables. Let's try an example that might happen on the IBM. Suppose the computer stops with an error in Line 500. LIST 500 to see what that line is:

```
500 LOCATE R,C:PRINT "HELLO"
```

Now, type PRINT R,C and press RETURN to see what the values of R and C are. You may find that line 500 is correct, but a previous line that calculates the value of R or C is incorrect. Once you know the values of the variables, you can look back to see how those values were obtained, hopefully pinpointing what went wrong.

DATA statements are the greatest source of errors. You may have a loop that reads items from DATA statements and then performs some operation using that data. If you get an error in a such a line, the data may be causing the error, not the statement. Because of this, you should always check your DATA statements as well as the program line indicated in the error message.

TRON and TROFF (TRace ON and TRace OFF) are two commands that help you debug programs. These BASIC commands are available on the Amiga, Apple, Atari ST, and IBM PC/PCjr. When you enter TRON and run a program, the computer prints each program line number as it is executed. This way, you get to see how the program flows. If the computer seems to lock up when a program is run, the TRON feature may reveal that the computer is actually in an infinite loop. TROFF returns things to normal. ©

The Elementary Amiga

Part 3

Jim Butterfield, Associate Editor

This installment looks at multitasking, filenames, and disk commands. There's also a CLI program that speeds up common CLI commands by taking advantage of the Amiga's easy-to-use ramdisk.

As I write these articles on my Amiga, the computer is doing several things. While entering text, I can call up the CLI or Workbench, write an example, and place the results in my word processor. I can start up the transfer of a document from disk to printer, and while that's going on, I keep typing. The reason I can do this is because the Amiga is a *multitasking* machine; it has the ability to run several programs at once.

Multiple Tasks With The CLI

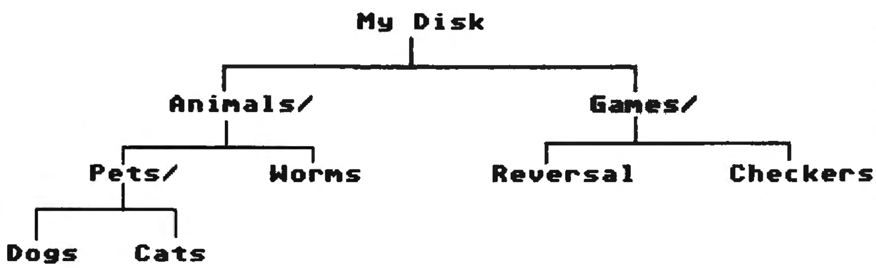
When you want your Amiga to do several jobs at once, you usually accomplish this by using several CLI windows. There are three popular ways of doing this:

- Start a new CLI by double-clicking the CLI icon.
- Start a new CLI by entering the command NEWCLI.
- Start a new CLI with the RUN command.

The last of these options is often the most convenient, but it has a drawback: RUN doesn't set up a new CLI window. Whatever happens, happens in the original CLI window. Sometimes this doesn't matter. If it does, you can always get around it with I/O redirection. For the moment, however, let's look at the first two methods of running multiple tasks from the CLI.

You probably noticed that your first CLI window uses a 1> prompt. Enter the command NEWCLI and a new window opens up with a 2> prompt. You may click in this new window, start a program running, and then click back into your first CLI window to do other jobs. If the program in CLI 2 should need attention—say, input—click in that window, do what's needed, and then go back to CLI 1. When you're finished with a CLI, click in the CLI window and enter ENDCLI. The CLI terminates and the window vanishes.

Example Disk Directory Structure



Filenames

CLI parameters are always separated by spaces. This makes it a good idea *not* to use spaces within a filename. For example, to edit a file called HOT DOG you cannot type ED HOT DOG. The CLI would separate the HOT from the DOG, taking HOT as the entire filename. Instead, you need to use quotes: ED "HOT DOG"

Some users substitute the underline character for a space. For example, the above file would be called HOT_DOG. This way, no quotation marks are needed.

There are two characters that cannot be used within filenames: the colon (:) and the slash (/). The colon is associated with devices, such as DF0: for disk or PRT: for printer. If you tried to enter the filename HOT:DOG from the CLI, the system would look for the file DOG in the device HOT. The slash is associated with directories. So HOT/DOG signals the CLI to look for the file DOG in the directory HOT.

When entered before a filename, the colon and slash have special "go back" significance. Suppose we have a disk organized as shown in the example disk directory figure. Using the CD (Current Directory) command, we may have placed ourselves deep within the Pets directory (MyDisk:Animals/Pets). At this moment, if we enter the command TYPE WORMS, nothing would happen because WORMS is not a file within our current directory (CATS and DOGS are all we have there).

We could supply the file's full path name by entering TYPE MyDisk: Animals/Worms, but this is a bit wordy. Starting a filename with a colon tells the computer to "Go back to the root directory," so we could use TYPE :Animals/Worms to get the desired effect. Alternatively, typing a slash at the start of a filename means "Go back a directory level." Thus, TYPE /Worms also does the trick. Think of it this way: These characters tell CLI to go back until they hit the same special character in the directory structure. If we move back from PETS looking for a slash, we find the one right after ANIMALS.

So, colons and slashes are forbidden from filenames. Spaces are allowed, but are awkward. Here are some other difficult characters:

(;)	often signifies a CLI comment (like BASIC's REM)
# and ?	often used in pattern matching
< and >	used with I/O redirection
= and +	special CLI significance
(" ")	the CLI normally removes these

All of these are legal filename characters, but they usually cause more trouble than they're worth. It's best to avoid them altogether.

Commands vs. Programs

In Part 2 of "The Elementary Amiga," we looked at a few CLI commands. What's interesting about CLI commands is that they are actually disk files. In fact, they're programs. You see, the command names that you enter are the filenames of programs that are loaded and run by the CLI.

If we enter the command COPY FROM DOG TO CAT, the Amiga searches the current directory and the special C command directory for a program called COPY. If it finds such a file, it loads and runs the program, passing along any command parameters. The parameters for our example would be the words FROM DOG TO CAT. The COPY program has to figure out what this list means and how to use it.

In a sense, you don't have commands, only programs. If you're accustomed to a traditional Commodore environment, this may come as something of a shock. In Commodore eight-bit machines, commands are fixed in ROM; the commands are always there and never change. Computerists experienced in CP/M or MS-DOS, on the other hand, find the Amiga system familiar. It all depends on what you're used to.

Your CLI commands can be treated like any other program file. If you don't need a command, you can delete it. If the command word doesn't suit you—perhaps it's too long—you can change it by renaming the program. If you find a better version of a command, you can update it simply by replacing the file.

Be careful: You're part of an Amiga community and when you customize your CLI commands excessively, you lose touch with "regular" systems. Deleting unused commands is fairly safe, as long as you keep a backup disk containing the original files. Changing command names is slightly more dangerous. EX is much easier to type than EXECUTE, but your new name might conflict with someone else's program. You might even forget the original command and be unable to handle an unmodified CLI disk. And, if someone offers you a new, improved directory program, I suggest that you name it DIRN rather than DIR so that you'll be able to distinguish between the new and old versions.

If you thin out your CLI commands by deleting files, you don't lose any capabilities—as long as you keep a backup. Suppose, for example, you delete the text editors ED and EDIT from your system disk's C directory; they take up a lot of space and you may not use them very much. Suddenly, you discover that you need to do some work using ED. No problem. Slip a disk

CLI Hit Parade

Everyone has their favorite set of CLI commands. Here are my personal CLI favorites:

Copy	Copies files from anywhere to anywhere (not just disk to disk).
CD	Stands for Current Directory. Lets you move from one directory to another.
Echo	Outputs strings (surprisingly useful).
List	An alternative form of DIR (more detailed and very handy).
MakeDir	Lets you create a new directory/drawer.
Dir	Tells you what's in a directory.
Rename	Gives a program a new filename or moves it to a specific directory.
Type	Outputs the contents of a file, in hex if you like.
EndCLI	Closes a CLI window.
NewCLI	Opens a new CLI window.
Execute	Executes instructions from a text or batch file.
Run	Executes commands as separate tasks by opening a new CLI.
Date	Shows or sets the system date and time.
Info	Tells you what devices are hooked up and lists their status.
Delete	Gets rid of a file.
Assign	Assigns a logical device to a particular disk directory.

containing ED into drive 1 and enter DF1:C/ED FILENAME. This tells the computer to execute the command located in the C directory in drive 1. The computer cheerfully loads and runs ED just as effectively as if it were on your system disk.

Many other commands such as Format, Install, DiskDoctor, and ED are important, but since I don't use them hour to hour, they didn't make my "hot list."

How does the CLI find these programs? It follows a path. If I enter DATE, it looks in the current directory for a file called DATE. If it doesn't find the program there, it continues along its path and looks in the C directory. To be exact, it looks for the logical device C:, which happens to be assigned to the C directory. Here it finds the program DATE and the command is executed.

If you'd like to see what path the CLI follows when searching for a program, enter the command PATH and you'll get the whole list.

Speeding CLI

Every time I enter one of my favorite commands, the program has to be located, loaded, and run. Sometimes my system disk isn't even in the drive, in which case I get a message telling me to put it back. I need a faster method.

To make CLI commands run faster, we should move them from the disk and place them somewhere where they can be reached more quickly. RAM does the job nicely.

Here's a sequence that used to be popular with the Amiga. It's a little dated now, and I'll suggest a better one in a moment.

```
copy :c ram:
assign c: ram:
```

The first command copies files from the C directory to RAM. (On the Amiga, you can use RAM just as if it were a disk.) Everything is copied, including commands that take up a lot of memory and are never used. Next, ASSIGN C: RAM: tells the computer to assign the logical device C: to the ramdisk. (Note that CLI commands and filenames can be entered in either uppercase or lowercase.) Originally C: is assigned to the C directory. From this point on, however, the system looks to RAM when searching for commands.

These two commands work well, but they use up more RAM than most people are willing to surrender. Here's a better plan: Move only the commands that you use often and then add RAM: to the CLI's path. Instead of ASSIGN C: RAM:, which prevents the Amiga from looking at the disk's C directory, we can use the command PATH RAM: ADD to *add* to the CLI's path. Now the CLI searches RAM: first, and if the command isn't found there, the Amiga searches your system disk's C directory.

Let's set up such a system. Enter the Amiga's editor with the command ED SYS:S/RAMDOS and type in the program below. When you're finished, press the ESC key followed by X and then RETURN. This saves the file RAMDOS to the S directory on the disk you booted from.

Looking at the program, note that we create a subdirectory called "c" in RAM, and put the commands there. This is done to avoid cluttering up the root directory. Also note that the command COPY is the first to go to RAM:C. This way, subsequent COPY commands run faster; the COPY program doesn't need to be dragged from disk each time it's used.

After the program is typed in and saved, enter EXECUTE RAMDOS from the CLI prompt to transfer commonly used commands to RAM. An interesting point: EXECUTE also has a path—if it doesn't find RAMDOS in the current directory, it looks in the S directory (to be accurate, logical device S: has been *assigned* to the S directory). As the file executes, it echoes periodic reports on the commands that it is setting up in RAM.

Watch This Space

We're just getting started. Next time we'll explain even more secrets about the CLI and Amiga disk directories. We'll also look closely at some CLI commands—many of them give you unexpected bonuses.

RAMDOS

```
cd sys:c
echo  ""
echo  "Moving CLI Commands to
       Ram:C"
echo  ""
makedir ram:c
copy  copy to ram:c
path  ram:c add
copy  cd to ram:c
copy  echo to ram:c
copy  list to ram:c
echo  "(Copy/CD/Echo/List)"
copy  makedir to ram:c
copy  dir to ram:c
copy  rename to ram:c
copy  type to ram:c
echo  "(MakeDir/Dir/Rename/Type)"
copy  endcli to ram:c
copy  newcli to ram:c
copy  execute to ram:c
copy  run to ram:c
echo  "(EndCLI/NewCLI/Execute/Run)"
copy  date to ram:c
copy  info to ram:c
copy  delete to ram:c
copy  assign to ram:c
echo  "(Date/Info/Delete/Assign)"
echo  ""
echo  "Commands Installed."
echo  ""
```

©

Rapid Reflex For IBM

Jason Pummill

You'll need fast thinking and quick hand movement to win at this challenging game. You can test your reflexes against yourself or up to five friends, and with the game's speed control, you can choose just the right pace for anyone in the family. A color/graphics adapter or equivalent hardware is required, along with BASICA for the PC, GW-BASIC for compatibles, or Cartridge BASIC for the PCjr.

If you want to test your reflexes, improve them, or just have some fun, then "Rapid Reflex" will challenge and entertain you. In Rapid Reflex, you must maneuver a fast-moving, growing line around the screen without touching a wall or allowing the line to cross itself. The longer the line becomes before you crash, the more points you'll acquire. You can play Rapid Reflex by yourself or with up to five people, and you can vary the game's speed. At first Rapid Reflex may seem simple, but the longer you play, the more challenging it gets.

Since Rapid Reflex is written entirely in BASIC, simply type it in, save a copy to disk, and type RUN. When the title screen appears, press any key to begin. First you'll be asked how many players will be playing this round, followed by a request for each player's name. Names must be less than ten characters long, and no more than five people can play in one round.

Next you'll be asked what speed, from 1 to 100 mph, you want to use for the game. If you're using a normal-speed PC or compatible, try starting with a speed of 75 mph. If you're using a machine with a faster processor, try 25 mph first.

Playing Rapid Reflex

Now you're ready to play. The first thing the computer does is set up the playing area as a pattern of dots. You'll see the name of the first person to play and hear three beeps. When the beeping stops, it's the first player's turn to control the constantly moving, expanding line.

You control the line's direction with the cursor control keys. The object of the game is to join as many dots as possible with your line. The line moves at a rapid pace, so it may take some practice to get the feel of the game. And it's important to remember that Rapid Reflex doesn't wait until you're ready to start. It beeps three times, and the line starts moving.

When the line crashes against a wall or itself, it makes an appropriate noise and displays the number of dots you've hit. Then the screen returns to the ring of dots, and a new game begins with the name of the second player displayed at the top. This process continues until each player has played three times.

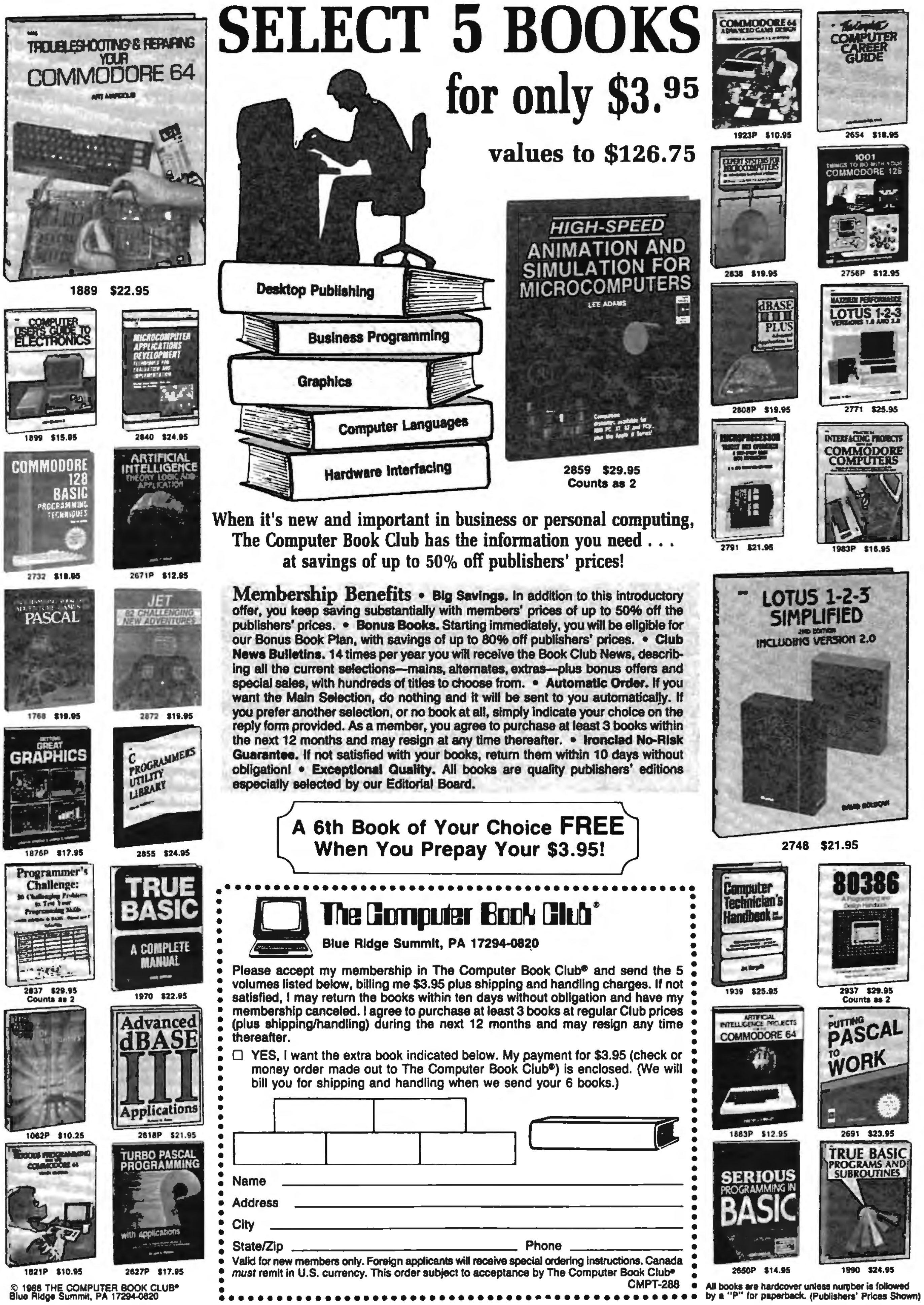
After each player has finished with his or her turn, the screen clears, and Rapid Reflex displays the final scores. These are presented as a chart with each player's name, score for each round, and total score. The names are ranked according to scores—highest score first.

You can play another game or return to BASIC. If you choose to play again, you'll be asked the number of players once more, and things will continue just the way they did the first time. If you choose to quit, you'll return to BASIC.

Rapid Reflex For IBM

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
HK 10 REM Copyright 1988 COMPUTE
      ! Publications, Inc.  All
      Rights Reserved
OJ 20 SCREEN 1:KEY OFF:WIDTH 40:
      COLOR 0,1:CLS
PL 30 REM make Title Screen
GH 40 PRESET(15,100)
IA 50 DRAW "C1 U40 R40 D20 L40 R
      20 F20"
ME 60 PRESET(105,100)
GF 70 DRAW "C1 L40 U20 R30 L30 U
      20 R40"
NC 80 PRESET(115,100)
EL 90 DRAW "C1 U20 R30 L30 U20 R
      40"
QL 100 PRESET(165,60)
LB 110 DRAW "C1 D40 R40"
DH 120 PRESET(255,100)
BD 130 DRAW "C1 L40 U20 R30 L30
       U20 R40"
FI 140 PRESET(265,100)
LB 150 DRAW "C1 E40 G20 H20 F40"
JM 160 LOCATE 5,13:PRINT"R  A  P
          I  D"
NH 170 LOCATE 17,4:PRINT "Copyri
       ght 1988 COMPUTE! Pub., I
       nc."
MJ 180 LOCATE 18,11:PRINT "All R
       ights Reserved"
EO 190 LOCATE 23,8:PRINT"<press
       any key to begin>"
KO 200 A$=INKEY$:IF A$="" THEN 2
       00
AA 210 CLS
```



SELECT 5 BOOKS
for only $3.95
values to $126.75
Desktop Publishing
Business Programming
Graphics
Computer Languages
Hardware Interfacing
TROUBLESHOOTING & REPAIRING YOUR COMMODORE 64
1889 $22.95
HIGH-SPEED ANIMATION AND SIMULATION FOR MICROCOMPUTERS
LEE ADAMS
2859 $29.95
Counts as 2
COMMODORE 64 ADVANCED GAME DESIGN
1923P $10.95
The Complete COMPUTER CAREER GUIDE
2654 $18.95
EXPERT SYSTEMS FOR MICROCOMPUTERS
2838 $19.95
1001 THINGS TO DO WITH YOUR COMMODORE 128
2756P $12.95
dBASE III PLUS
2808P $19.95
MAXIMUM PERFORMANCE LOTUS 1-2-3 VERSIONS 1A AND 2.0
2771 $25.95
MICROPROCESSOR
2791 $21.95
INTERFACING PROJECTS COMMODORE COMPUTERS
1983P $16.95
COMPUTER USER'S GUIDE TO ELECTRONICS
1899 $15.95
MICROCOMPUTER APPLICATIONS DEVELOPMENT
2840 $24.95
COMMODORE 128 BASIC PROGRAMMING TECHNIQUES
2732 $18.95
ARTIFICIAL INTELLIGENCE THEORY LOGIC AND APPLICATION
2671P $12.95
PASCAL
1768 $19.95
JET 82 CHALLENGING NEW ADVENTURES
2872 $19.95
GREAT GRAPHICS
1876P $17.95
C PROGRAMMERS UTILITY LIBRARY
2855 $24.95
When it's new and important in business or personal computing,
The Computer Book Club has the information you need . . .
at savings of up to 50% off publishers' prices!
Membership Benefits • Big Savings. In addition to this introductory offer, you keep saving substantially with members' prices of up to 50% off the publishers' prices. • Bonus Books. Starting immediately, you will be eligible for our Bonus Book Plan, with savings of up to 80% off publishers' prices. • Club News Bulletins. 14 times per year you will receive the Book Club News, describing all the current selections—mains, alternates, extras—plus bonus offers and special sales, with hundreds of titles to choose from. • Automatic Order. If you want the Main Selection, do nothing and it will be sent to you automatically. If you prefer another selection, or no book at all, simply indicate your choice on the reply form provided. As a member, you agree to purchase at least 3 books within the next 12 months and may resign at any time thereafter. • Ironclad No-Risk Guarantee. If not satisfied with your books, return them within 10 days without obligation! • Exceptional Quality. All books are quality publishers' editions especially selected by our Editorial Board.
LOTUS 1-2-3 SIMPLIFIED
2ND EDITION INCLUDING VERSION 2.0
2748 $21.95
A 6th Book of Your Choice FREE
When You Prepay Your $3.95!
Programmer's Challenge:
2837 $29.95
Counts as 2
TRUE BASIC A COMPLETE MANUAL
1970 $22.95
1062P $10.25
Advanced dBASE III Applications
2618P $21.95
SERIOUS PROGRAMMING FOR THE COMMODORE 64
1821P $10.95
TURBO PASCAL PROGRAMMING with applications
2627P $17.95
The Computer Book Club®
Blue Ridge Summit, PA 17294-0820
Please accept my membership in The Computer Book Club® and send the 5 volumes listed below, billing me $3.95 plus shipping and handling charges. If not satisfied, I may return the books within ten days without obligation and have my membership canceled. I agree to purchase at least 3 books at regular Club prices (plus shipping/handling) during the next 12 months and may resign any time thereafter.
☐ YES, I want the extra book indicated below. My payment for $3.95 (check or money order made out to The Computer Book Club®) is enclosed. (We will bill you for shipping and handling when we send your 6 books.)
Name
Address
City
State/Zip
Phone
Valid for new members only. Foreign applicants will receive special ordering instructions. Canada must remit in U.S. currency. This order subject to acceptance by The Computer Book Club®
CMPT-288
Computer Technician's Handbook
1939 $25.95
80386 A Programming and Design Handbook
2937 $29.95
Counts as 2
ARTIFICIAL INTELLIGENCE PROJECTS COMMODORE 64
1883P $12.95
PUTTING PASCAL TO WORK
2691 $23.95
SERIOUS PROGRAMMING IN BASIC
2650P $14.95
TRUE BASIC PROGRAMS AND SUBROUTINES
1990 $24.95
© 1988 THE COMPUTER BOOK CLUB®
Blue Ridge Summit, PA 17294-0820
All books are hardcover unless number is followed by a "P" for paperback. (Publishers' Prices Shown)

```
OB 220 REM Input screen for numb
       er of players and their n
       ames
NM 230 INPUT "No. of competitors
       ";NOC
CD 240 IF NOC<1 OR NOC>5 THEN BE
       EP:PRINT:GOTO 230
KH 250 NOC=INT(NOC)
JN 260 PRINT
IB 270 FOR X=1 TO NOC
JD 280 PRINT "Name of Competitor
        Number";
FF 290 PRINT USING "##";X;:PRINT
       "? ";
CM 300 LINE INPUT C$(X)
IE 310 PRINT
KK 320 IF LEN(C$(X))<1 OR LEN(C$
       (X))>10 THEN BEEP:GOTO 28
       0
GF 330 NEXT X
AH 340 CLS
LD 350 INPUT "Speed (1-100 mph)"
       ;SP$:SP=VAL(SP$)
IB 360 IF SP>100 OR SP<1 THEN BE
       EP:GOTO 350
BN 370 CLS
KF 380 REM Set up ring routine
HH 390 LOCATE 1,11:PRINT "Settin
       g up ring..."
EE 400 LINE(8,8)-(312,192),2,B
FC 410 LINE(10,10)-(310,190),3,B
       F
GJ 420 FOR X=11 TO 310 STEP 5
CP 430 LINE(X,10)-(X+3,190),0,BF
HI 440 NEXT X
HN 450 FOR X=11 TO 190 STEP 5
KJ 460 LINE(10,X)-(310,X+3),0,BF
HO 470 NEXT X
JF 480 DIM DOTS(4000)
OF 490 GET(10,10)-(310,190),DOTS
FD 500 LOCATE 1,11:PRINT SPC(18)
ID 510 REM Pause for next player
        to get ready and name of
        variables
QO 520 P=1
FP 530 M$="Get ready: "+C$(P)
JC 540 LOCATE 1,20-LEN(M$)/2:PRI
       NT M$
OA 550 FOR X=1 TO 3
MA 560 SOUND 1000,3
HF 570 SOUND 32767,10
HB 580 NEXT X
LJ 585 FOR X=1 TO 1000:NEXT X
GD 590 PLAY "MF"
IK 600 LOCATE 1,20-LEN(M$)/2:PRI
       NT SPC(LEN(M$))
PM 610 N=0
EB 620 A=10:D=10
PC 630 S=77
BD 640 REM Subroutine for actual
        game play
FO 650 IF INKEY$<>"" THEN 650
NC 660 PRESET(10,10)
DC 670 A$=INKEY$:IF A$="" THEN 6
       90
NP 680 IF LEN(A$)>1 THEN S=ASC(M
       ID$(A$,2))
PK 690 IF S=77 THEN A=A+5
PN 700 IF S=75 THEN A=A-5
EA 710 IF S=72 THEN D=D-5
BM 720 IF S=80 THEN D=D+5
NE 730 N=N+1
CL 740 IF POINT(A,D)<>3 THEN 790
ID 750 LINE-(A,D),1
CA 760 FOR Q=1 TO 101-SP:NEXT
JP 770 GOTO 670
MN 780 REM Subroutine for hittin
       g wall or self
GA 790 FOR X=1 TO 16
MN 800 SOUND 100,.5
OD 810 SOUND 130,.5
HI 820 NEXT X
BD 830 LOCATE 1,14:PRINT "Points
       :"N
KF 840 FOR X=1 TO 2000:NEXT X
MD 850 LOCATE 1,14:PRINT SPC(20)
II 860 REM Record score and see
       how many turns left
II 870 TM(P)=TM(P)+N
LA 880 SCR(P,TMS+1)=N
QJ 890 P=P+1
HN 900 IF P>NOC THEN P=1:TMS=TMS
       +1:IF TMS=3 THEN 960
OI 910 REM Recover dots and retu
       rn to Next player subrout
       ine
BI 920 LINE(10,10)-(310,190),0,B
       F
GE 930 PUT(10,10),DOTS
FF 940 GOTO 530
EO 950 REM Change to text screen
        and figure placings
OA 960 SCREEN 0:WIDTH 40:CLS
CH 970 LP=LP+1:IF LP>NOC THEN 11
       00
JP 980 FOR X=1 TO NOC
PE 990 IF TM(X)>SC(LP) THEN GOSU
       B 1040
PE 1000 NEXT X
DD 1010 PL(LP)=P
JC 1020 P=0
IL 1030 GOTO 970
QN 1040 FOR Y=1 TO NOC
OE 1050 IF PL(Y)=X THEN RETURN
BB 1060 NEXT Y
BD 1070 SC(LP)=TM(X):P=X
JD 1080 RETURN
IF 1090 REM Make Reflex Finals s
        creen
HJ 1100 COLOR 4
JN 1110 PRINT SPC(13) CHR$(201)
        STRING$(13,205) CHR$(187
        )
IC 1120 PRINT SPC(13) CHR$(186);
JB 1130 COLOR 7
DK 1140 PRINT "Reflex Finals";
II 1150 COLOR 4
LM 1160 PRINT CHR$(186)
MI 1170 PRINT SPC(13) CHR$(200)
        STRING$(13,205) CHR$(188
        )
AH 1180 PRINT
BK 1190 PRINT
EH 1200 COLOR 1
KJ 1210 ST$=STRING$(5,196)
MO 1220 PRINT SPC(2) CHR$(218) S
        T$ ST$ CHR$(194) ST$ CHR
        $(194) ST$ CHR$(194) ST$
         CHR$(194) ST$ CHR$(191)
PE 1230 PRINT SPC(2) CHR$(179);
KL 1240 COLOR 7
BJ 1250 PRINT "Competitor";
GJ 1260 COLOR 1
LP 1270 PRINT CHR$(179);
KH 1280 COLOR 7
BA 1290 PRINT SPC(2) "1" SPC(2);
FJ 1300 COLOR 1
KP 1310 PRINT CHR$(179);
JH 1320 COLOR 7
GE 1330 PRINT SPC(2) "2" SPC(2);
FF 1340 COLOR 1
LL 1350 PRINT CHR$(179);
KD 1360 COLOR 7
IE 1370 PRINT SPC(2) "3" SPC(2);
GB 1380 COLOR 1
JC 1390 PRINT CHR$(179) ;
JD 1400 COLOR 7
LD 1410 PRINT "total";
FB 1420 COLOR 1
NA 1430 PRINT CHR$(179)
IK 1440 PRINT SPC(2) CHR$(192) S
        T$ ST$ CHR$(193) ST$ CHR
        $(193) ST$ CHR$(193) ST$
         CHR$(193) ST$ CHR$(217)
AE 1450 PRINT
NO 1460 PRINT SPC(2) CHR$(218) S
        T$ ST$ CHR$(194) ST$ CHR
        $(194) ST$ CHR$(194) ST$
         CHR$(194) ST$ CHR$(191)
NO 1470 S1$=CHR$(179)+STRING$(10
        ,32)+CHR$(179)+STRING$(5
        ,32)+CHR$(179)+STRING$(5
        ,32)+CHR$(179)+STRING$(5
        ,32)+CHR$(179)+STRING$(5
        ,32)+CHR$(179)
LC 1480 S2$=CHR$(195)+ST$+ST$+CH
        R$(197)+ST$+CHR$(197)+ST
        $+CHR$(197)+ST$+CHR$(197
        )+ST$+CHR$(180)
JD 1490 FOR X=1 TO NOC-1
FH 1500 PRINT SPC(2) S1$
HD 1510 PRINT SPC(2) S2$
AE 1520 NEXT X
GA 1530 PRINT SPC(2) S1$
IM 1540 PRINT SPC(2) CHR$(192) S
        T$ ST$ CHR$(193) ST$ CHR
        $(193) ST$ CHR$(193) ST$
         CHR$(193) ST$ CHR$(217)
KE 1550 COLOR 7
LM 1560 REM Fill in score data a
        nd ask to play again
AG 1570 FOR X=1 TO NOC
NM 1580 LOCATE X*2+9,4:PRINT C$(
        PL(X))
PO 1590 LOCATE X*2+9,15:PRINT MI
        D$(STR$(SCR(PL(X),1)),2)
NI 1600 LOCATE X*2+9,21:PRINT MI
        D$(STR$(SCR(PL(X),2)),2)
HD 1610 LOCATE X*2+9,27:PRINT MI
        D$(STR$(SCR(PL(X),3)),2)
IJ 1620 LOCATE X*2+9,33:PRINT MI
        D$(STR$(TM(PL(X))),2)
BJ 1630 NEXT X
OE 1640 CLEAR
PM 1650 LOCATE CSRLIN+3,16:PRINT
         "Play Again?"
EN 1660 A$=INKEY$:IF A$="" THEN
        1660
OI 1670 IF A$="Y" OR A$="y" THEN
         SCREEN 1:COLOR 0,1:GOTO
         210
ME 1680 IF A$="N" OR A$="n" THEN
         CLS:END
CD 1690 GOTO 1660
```
©

Apple Fast Scan

Bruce E. Howell, D.D.S.

Apple programmers can save a lot of time with this disk-based string-search utility. At machine language speed, random access files may be searched for any combination of characters. ProDOS is required.

Random access text files provide a fast means of storing and retrieving large amounts of data—fast, that is, if you know the record number of the data you want to access. Without knowing the specific record number, random access files must be searched sequentially, one record at a time, until the desired piece of information is found.

"Fast Scan" is a combination machine language/BASIC program that searches random access text files for any sequence of characters. You can use it to search for the first occurrence, second occurrence, or all occurrences of a specified string. Whenever an item is found, the number of the matching record is returned in a BASIC variable.

Typing It In

Program 1 is the machine language portion of Fast Scan. Type it in using "MLX," the machine language entry program found elsewhere in this issue. When prompted, enter the following information:

```
STARTING ADDRESS?     02F0
ENDING ADDRESS?       03D7
```

After you have entered all the data from Program 1, save a copy to disk using the filename FASTSCAN.BIN.

Program 2 is the BASIC part of Fast Scan. Type it in using "The Automatic Proofreader," found elsewhere in this issue. Program 2 is short in size, but because it manipulates disk files, it requires accurate typing. Be sure to save Program 2 to disk when you're finished typing it in.

Programs 3 and 4 provide examples of how to use Fast Scan. Although not required, it is recommended that you type in Programs 3 and 4 to see how Fast Scan works. To type in Program 3, first load Program 2, enter the additional lines found in Program 3, and then save the resulting program to disk.

Program 4 generates a sample random access text file you can use to test Program 3. If you wish to use Program 4, type it in and save a copy to disk.

Using The Program

Before you can use Fast Scan, you must BRUN the machine language file FASTSCAN.BIN. You can do this from within a program using the following code:

```
PRINT CHR$(4);"BRUN FASTSCAN.BIN"
```

See line 10 in Program 3 for a similar example.

Program 2 contains the subroutines needed to use Fast Scan. Currently, these subroutines occupy program lines 5000–5230. Using a renumber utility, you can move the subroutine to a more suitable location.

Only two variables must be set prior to calling Fast Scan: SS$ and NF$. Set SS$ equal to the string of characters you're searching for, and NF$ equal to the name of the random access text file. To search for the string "JOHNSON" in the file ADDRESSES, for example, use the statement

```
SS$ = "JOHNSON":NF$ =
  "ADDRESSES"
```

Once these two variables are set, simply GOSUB 5000. If the string JOHNSON is found—in a record by itself, or as part of another string—the variable R% is set equal to 1, and N% contains the record number where the string is located. If the string is not found, R% returns a 0. Below are all the possible values for R%:

Value	Explanation
1	String found at record number N%
0	String not found
−1	File is not on disk
−2	File is not a text file
−3	File is not a random access file
−4	File is empty

As you can see, Fast Scan provides complete error checking.

The subroutine at 5000 returns the *first* occurrence of the search string. After executing a GOSUB 5000, however, you may search for subsequent occurrences as well. You continue the search with a GOSUB 5200. The values returned are the same for the initial search: R% contains the status of the search—found or not found—and

N% contains the record number. To find all occurrences of a string, just continue to GOSUB 5200 until R% returns a 0.

An Example

Program 3 provides a good example of how to use Fast Scan. Before running Program 3, load and run Program 4. The file TEST is written to disk. TEST is a random access text file containing 1000 records with the following information:

RECORD #*n*
NAME #*n*
ADDRESS #*n*

where *n* is the record number 0–999.

Now, load and run Program 3. Be sure a disk containing the file FASTSCAN.BIN (Program 1) is in the drive.

First, Program 3 asks you for the name of the random access file you wish to search. Answer TEST. Next, you're asked to enter a search string. Enter NAME #365. The program searches the disk for the file TEST, prints the message FIRST FOUND IN RECORD #365, and then aborts with the message NO MORE.

Try searching for other things, such as the number 10—you may be surprised how often this number occurs. If you wish, you may use this program to sift through your own random access files. (You do have some, don't you?)

How It Works

Fast Scan works by BLOADing the text file into memory, in segments, and then searching the file from machine language. Because DOS 3.3 does not permit BLOADing text files or BLOADing files in segments, Fast Scan works in ProDOS only.

Fast Scan determines the record length and file size of random access files through use of the CATALOG command. To calculate the maximum number of records that can fit into memory at one time, the FRE command is used. After a FRE, memory locations 109 and 110 point to the current end of variable storage, while locations 111 and 112 specify the start of string storage.

Most of Fast Scan's housekeeping is done from BASIC. Machine language performs the actual string searching, however. Fast Scan's machine language search routine is called using the following syntax:

& F,RL,NR,S$,N%,R%

where F stands for Find, RL is the record length, NR is the number of records present in memory, S$ is the string to be searched for, N% is the record number in memory where the string is found, and R% is the result of the search (1 = found, 0 = not found).

To find the next or subsequent records in memory, the syntax is

& N,N%,R%

where N stands for next, and N% and R% are the same as shown above.

For instructions on entering these programs, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

Program 1: Fast Scan—Machine Language

```
02F0: A9 4C 8D F5 03 A9 00 8D 3A
02F8: F6 03 A9 03 8D F7 03 60 51
0300: 20 B7 00 48 20 B1 00 68 B8
0308: C9 46 F0 0A C9 4E D0 03 6F
0310: 4C 9C 03 4C C9 DE 20 BE 51
0318: DE 20 7B DD 20 08 E1 A5 6D
0320: A0 85 E8 A5 A1 85 E9 20 66
0328: BE DE 20 7B DD 20 08 E1 62
0330: A5 A0 85 1A A5 A1 85 1B 5D
0338: 20 BE DE 20 7B DD A0 00 70
0340: B1 A0 85 E0 C8 B1 A0 85 D9
0348: E1 C8 B1 A0 85 E2 18 A5 3F
0350: 6E 69 01 85 E5 A9 00 85 BB
0358: 1C 85 1D 85 E4 A2 00 86 02
0360: EA 86 E6 E8 86 E7 A2 00 02
0368: A0 00 B1 E4 F0 30 D1 E1 11
0370: D0 2C C8 C4 E0 D0 F3 86 08
0378: F9 20 BE DE 20 E3 DF A0 3A
0380: 00 A5 1C 91 83 C8 A5 1D 34
0388: 91 83 20 BE DE 20 E3 DF 47
0390: A0 00 A5 E6 91 83 C8 A5 DB
0398: E7 91 83 60 A6 F9 E6 E4 3D
03A0: D0 02 E6 E5 E8 D0 02 E6 40
03A8: EA E4 E9 D0 BB A5 EA C5 B7
03B0: E8 D0 B5 A5 1C C5 1A D0 6D
03B8: 0C A5 1D C5 1B D0 06 A9 FF
03C0: 00 85 E7 F0 B2 E6 1D D0 70
03C8: 02 E6 1C A5 70 C5 E5 D0 9E
03D0: 8C A2 02 4C 12 D4 4C FD 45
```

Program 2: Fast Scan—BASIC Subroutine

```
9F 5000 REM FAST SCAN
51 5010 REM COPYRIGHT 1988 COMPU
        TE! PUBL.
9F 5020 T = 0:SB = 0:EB = 0:BS =
         0:RB = 0:CB = 0:HR% = 0
        :B = 0:N% = 0:R% = 0: PR
        INT CHR$ (4);"PREFIX": I
        NPUT DN$
A8 5030 NF$ = LEFT$ (NF$ + "
                    ",15)
89 5040 PRINT CHR$ (4);"OPEN";DN
        $;",TDIR": PRINT CHR$ (4
        );"READ";DN$
BA 5050 INPUT A$,A$,A$
8B 5060 FOR I = 1 TO 3:I = 1: IN
        PUT I$: IF I$ = "" THEN
        PRINT CHR$ (4);"CLOSE";D
        N$:I = 3: NEXT :R% = - 1
        : RETURN
D8 5070 IF MID$ (I$,2,15) < > NF
        $ THEN NEXT
4C 5080 I = 3: NEXT : PRINT CHR$
         (4);"CLOSE";DN$
1D 5090 IF MID$ (I$,18,3) < > "T
        XT" THEN R% = - 2: RETUR
        N
C6 5100 RL = VAL ( MID$ (I$,75,5
        )): IF RL = 0 THEN R% =
        - 3: RETURN
B7 5110 SZ = VAL ( MID$ (I$,66,6
        )): IF SZ = 0 THEN R% =
        - 4: RETURN
85 5120 PRINT CHR$ (4);"FRE"
33 5130 SB = INT ((255 + PEEK (1
        09) + PEEK (110) * 256)
        / 256) * 256:EB = INT ((
         PEEK (111) + PEEK (112)
         * 256) / 256) * 256 - 1
        :BS = EB - SB:RB = INT (
        BS / RL):CB = RB * RL
CA 5140 HR% = 0:B = 0
EB 5150 PRINT CHR$ (4);"BLOAD";D
        N$;NF$;",TTXT,A";SB;",L"
        ;CB;",B";B
87 5160 T = INT ((SZ - B) / RL):
         IF T < RB THEN RB = T
87 5170 & F,RL,RB,SS$,N%,R%: IF
        R% = 1 THEN N% = N% + HR
        %: RETURN
17 5180 HR% = HR% + N%:B = B + C
        B: IF B < = SZ THEN 5150
FB 5190 RETURN
DD 5200 REM  SEARCH FOR NEXT OCC
        URANCE
E2 5210 R% = 0: IF B > SZ THEN R
        ETURN
17 5220 & N,N%,R%: IF R% = 1 THE
        N N% = N% + HR%: RETURN
84 5230 GOTO 5180
```

Program 3: Fast Scan—Demo Program

```
75 6 REM COPYRIGHT 1988 COMPUTE!
      PUBLICATIONS, INC. ALL RIG
     HTS RESERVED.
87 7 HOME : PRINT "COPYRIGHT 198
     8 COMPUTE! PUBL.": PRINT "A
     LL RIGHTS RESERVED."
59 8 FOR X = 1 TO 1500: NEXT X:
     HOME
54 10 PRINT CHR$ (4);"BRUN FASTS
      CAN.BIN"
BA 20 INPUT "FILENAME: ";NF$: IN
      PUT "SEARCH STRING: ";SS$
85 30 GOSUB 5000: IF R% = 0 THEN
       PRINT "NOT FOUND": END
E9 40 PRINT "FIRST FOUND IN RECO
      RD #";N%
A6 50 GOSUB 5200: IF R% = 0 THEN
       PRINT "NO MORE": END
D1 60 PRINT "ALSO FOUND IN RECOR
      D  #";N%: GOTO 50
```

Program 4: Fast Scan—Test-File Creator

```
51 10 D$ = CHR$ (4)
2C 20 PRINT D$;"OPEN TEST,L50"
A1 30 FOR I = 0 TO 999
AD 40 PRINT D$;"WRITE TEST,R";I
29 50 PRINT "RECORD #";I
89 60 PRINT "NAME #";I
C3 70 PRINT "ADDRESS #";I
AB 80 NEXT
09 90 PRINT D$;"CLOSE"
```

©

Field Sort For The 64

William J. Baird

Combine the lightning-fast speed of Quicksort with the flexibility of multiple-field sorting, add the ability to sort in both ascending and descending order, and you have what is probably the best sorting program we've ever published for the 64. A disk drive is required.

In September 1983, COMPUTE! published "Ultrasort"—at that time, the fastest sorting program available for home computers. Ultrasort was based on C.A.R. Hoare's Quicksort algorithm and sorted a 1000-element string array in less than eight seconds. In September 1984, COMPUTE! published an enhancement that was even *faster*—"Lightning Sort." This program sorted the 1000-element array in an almost unbelievable 2.1 seconds. Now, there is "Field Sort," which combines the speed of Lightning Sort with the added features of sorting across multiple fields and sequencing in either ascending or descending order.

Field Sort consists of two machine language programs—"Driver" and "Main"—and a BASIC program—"Mover". Driver (Program 1) is a one-block movable driver. Main (Program 2) is a three-block, immovable sort program that resides beneath the 64's Kernal ROM. Since both programs are written entirely in machine language, they both must be entered using the "MLX" machine language entry program found elsewhere in this issue. Be sure to read the instructions for using MLX before you begin entering data. When you run MLX, you'll be asked for a starting and an ending address for the data you'll be entering. The correct values for Driver (Program 1) are as follows:

Starting address: C000
Ending address: C067

After you've entered all the data, be sure to save a copy with the filename FSORT.DRIVER before leaving MLX—the Field Sort demo programs, which we will discuss later, look for a file with this name.

Main, the actual Field Sort program code, resides in RAM beneath the Kernal ROM. This causes a minor problem in entering Program 2 using MLX. Since special steps are required to use this area of memory for program storage, MLX's built-in address checking normally rejects addresses in this range. Thus, you must make a temporary modification to MLX while entering the data from Program 2. After loading MLX, but before running it, replace line 1040 with the following:

```
1040 GOSUB 1080: F=0: RETURN
```

Note that this modification is only for entering Program 2; it is not a correction to the MLX program. You need not make a permanent change to MLX.

The starting and ending addresses for Program 2 are:

Starting address: E000
Ending address: E28F

After you've entered all the data, be sure to save a copy with the filename FSORT.MAIN before leaving MLX—again, the demo programs look for a file with this name.

Mover (Program 3) is a BASIC program that allows you to move the driver to another place in memory. You can use Mover to move the driver to any address in the ranges of 828–918, 2048–40858, or 49152–53146. Knowing the location of Driver is important, since this is the address to which BASIC must SYS for Field Sort. If you forget Driver's load address, you can run Mover, and it will report it.

Using Field Sort

You call Field Sort from BASIC with the command

```
SYS Q,F[$],N,A$(K)
```

where *Q* is the location of the driver subroutine, F[$] describes one to eight fields to be sorted, *N* is the number of array elements to be sorted, and A$(K) identifies the first element of the array that is to be included in the sort. The square brackets indicate an optional parameter type. If you choose a string, F$, then field sorting is flagged. If a number, F, is used, then position sorting occurs.

Sorting by position means that there is exactly one field to be sorted, and it will begin in the string position indicated by F. The field stretches out to the end of the string, so length information is not needed. You must build the numeric expression F with the formula $F = d + p$ where *d* is the direction (a value of 0 indicates ascending sequence; 256, descending), and *p* is the position within the string. (*P* may be any value greater than 0 and less than 256.)

For field sorting, each field is defined by three ingredients: field position within the string, field length, and sort direction (ascending or descending).

F$ consists of from one to eight triples, with each triple having the form

CHR$(*p*)+CHR$(*l*)+CHR$(*d*)

where *p* is the field position within the string, *l* is the field length, and *d* is the sort direction (a value of 0 indicates ascending sequence; 1 indicates descending). Both *p* and *l* values must be greater than 0, and their sum must be less than or equal to 256. Field Sort processes the leftmost triple first and continues left to right, so place the most important field definition at the front of F$.

Field Sort manages up to eight fields within the string array. (Eight is an arbitrary limit that seems sufficient without being wasteful.) Since field sorting can be slightly slower than position sorting, single-field sorts are automatically converted to position sorts. Nothing prevents you from defining overlapping fields except your own desire to sort as fast as possible.

Sort Direction

You may wonder why sort direction is included, since BASIC arrays can be accessed backwards. Descending sorts are helpful in special applications where it is not practical to access arrays backwards. This happens when small ranges within the array need special sequencing, in statistical reports or spreadsheet applications, for example. Multiple-field sorts that allow this option in each field can yield sophisticated results.

Position Sorting

The ability to sort a field by its position within a string is important because it makes processing complex information in each string array element easy. This means you can include more information in each string. For example, imagine a string array in which the first three characters of each element are reserved for a person's age and are aligned so that the one's digit is always in position three. Leading unused digits will be left blank. Also, visualize a blank in position four and a person's name starting in position five. Here's an example of the layout:

Figure 1

1	4	5
Age		Name . . .

If you sort this array using position one, you'll have a list of people sequenced by age and also by name where ages are the same. If you sort the same array using position five, you'll develop a list of people sequenced by name, without regard to age. Notice the economy of placing all the information under one roof. If age and name information were kept in separate arrays, then they could become disassociated during sorting, so more RAM would be needed for array descriptor space, and the 64's garbage-collection passes would take longer.

During position sorts, Field Sort uses the tails of the array strings beginning at the indicated position. Valid sort positions lie in the range of 1–255, which is the length limit for non-null BASIC string variables. Any request outside these values is diagnosed and an ILLEGAL QUANTITY error message is returned. Strings shorter than the starting sort position are evaluated as less than CHR$(0) during the sort.

Alternate-Position Sorting

Strangely enough, position sorting vastly improves performance in certain cases. The Quicksort algorithm becomes dramatically slower if the array is not initially in random sequence. For example, suppose a few items are added to the end of a previously sorted array. This is a typical situation, and it will take a long time to resort just to integrate the last few elements. This is quite a problem until you consider that an array sequenced on position one may already be in random sequence when viewed from a different position, say three. (You may pick any alternate position that makes sense for your application.)

So, if the array were sorted on position three, it might appear to be in random order relative to position one. The preordered array can be put into random sequence quickly by sorting it using the "wrong" field position. The array can then be rapidly sorted again using the "correct" field position. Program 4, a demonstration program, shows the difference in timings. The demo takes 40 seconds to sort a presequenced 1000-element array directly. However, it takes only 4 seconds to sort the same array twice using the alternate-position approach.

Multiple-Field Sorting

Sometimes, even position sorting is not adequate. An example is a date (mm/dd/yy) and name list, where month (mm), day (dd), and year (yy) are each two decimal digit fields. Any field less than ten has a leading zero in the first position. The numeric pairs are separated by the / character.

Figure 2

1		4		7		10
mm	/	dd	/	yy		Name . . .

Sorting this array in position one yields a list of dates by month and day. All array elements having the same month and day are also ordered by year. If this is an annual calendar application, then we've done well. But what if we also want a chronological list of dates spanning all years? Sorting on position seven sorts the array into year order, but there is no month and day sequence within each year. One solution is to rebuild the array placing the year first (yy/mm/dd). But this is a slow process, especially if there is a need to keep both sequences. How can this problem be solved? Field Sort is the perfect solution.

The date example is sorted in chronological sequence by using the following F$ encoding (each triple is shown on a separate line for readability):

F$ = CHR$(7)+CHR$(2)+CHR$(0)
+ CHR$(1)+CHR$(5)+CHR$(0)
+ CHR$(10)+CHR$(246)+CHR$(0)

The first triple selects the yy field, the second triple describes the mm/dd field, and the last triple defines the name field.

Managing Complex Strings

It should be clear that BASIC programmers need to follow certain rules when using Field Sort. First, similar chunks of data must start in like positions throughout the array. Notice that this rule automatically says that all similar fields must be the same length, except for the last one in the string. Building array strings is usually done by concatenating individual field strings, where each field is first subjected to content, format, and length verification. The BASIC program Dates (Program 5) is provided as an example. It demonstrates string-handling techniques and sorting applied to a birthday list. It also demonstrates that position five (the first digit of day of the month field) is a fair randomizer to set up resorts.

Technical Notes

There is a natural tendency to use Field Sort in conjunction with other software. It avoids popular RAM locations by using the RAM below the Kernal ROM where few BASIC-support ML routines reside. This frees more accessible RAM for either data or other software.

All interrupts must be disabled to bank the Kernal ROM. Hence, the keyboard scan and software jiffy clock updates are also suspended. (The BASIC reserved variables TI and TI$ are used to access the software jiffy clock on the 64.) Those who need timing services, accurate to one-tenth of a second, should turn to the hardware updated BCD clock, located at 56328–56311. Demo shows how to use the clock. All other specialized routines that depend on interrupts are also disabled during sorts.

Since no processor time is used for keyboard scans and jiffy clock updates, the processor can concentrate on sorting. Screen blanking is added to eliminate the I/O bus memory access competition between the CPU and the video chip. The overall sort performance improvement is around 7 percent. Disruption of any split-screen raster interrupts is also masked by screen blanking to make sorting more palatable. Screen blanking can be disquieting during long sorts. Mover (Program 3) can eliminate the screen-blanking option for those who find themselves sorting longer than they'd like but cannot reduce the time with the approach described earlier.

With the Kernal ROM switched out during sorts, Field Sort is effective for strings in the 8K of RAM beneath the BASIC ROM. This opens new vistas for expanded BASIC string-management routines. It is interesting that this method, which makes the sort faster and more powerful, also allows for larger arrays that need the better service.

Please refer to the "MLX" article in this issue before entering the programs Driver and Main.

Program 1: Driver

```
C000:20 FD AE 20 9E AD 24 0D E9
C008:30 0B 20 F7 B7 20 51 C0 89
C010:20 0C E0 F0 06 20 51 C0 E3
C018:20 25 E0 20 3B C0 20 FD 2C
C020:AE 20 8A AD 20 F7 B7 20 9D
C028:FD AE 20 9E AD AD 11 D0 59
C030:29 EF 8D 11 D0 20 51 C0 6F
C038:20 44 E0 A5 01 09 07 85 11
C040:01 58 AD 11 D0 09 10 8D 77
C048:11 D0 A6 D8 F0 17 4C 37 9C
C050:A4 78 A5 01 29 F8 85 01 40
C058:A9 1D 85 D8 CD E8 E1 D0 DA
C060:DA A9 00 85 D8 60 0D 00 74
```

Program 2: Main

```
E000:A0 00 B1 03 85 14 C8 C8 4F
E008:B1 03 85 15 A5 15 C9 02 7C
E010:B0 2F 85 93 A6 14 F0 29 70
E018:CA CA 8E F6 E1 8E 0B E2 75
E020:A0 00 84 02 60 A0 02 B1 1E
E028:64 99 02 00 88 10 F8 C9 03
E030:03 90 0E F0 CB C9 19 B0 D0
E038:08 38 E9 03 30 03 D0 FA A3
E040:60 4C BF E0 38 A5 14 E9 B5
E048:01 85 FB A5 15 E9 00 85 9B
E050:FC A5 0D F0 6D 38 A5 58 9A
E058:E9 07 85 5A A5 59 E9 00 8D
E060:85 5B C5 30 90 5C D0 06 15
E068:A5 5A C5 2F 90 54 A0 04 5A
E070:B1 5A C9 01 D0 4C A0 02 E5
E078:18 A5 5A 71 5A 85 5C C8 7C
E080:A5 5B 71 5A 85 5D A2 01 A7
E088:A5 47 9D 00 D0 85 05 A5 EE
E090:48 9D 14 D0 85 06 A0 03 F5
E098:18 A5 05 65 FB 85 05 9D 64
E0A0:28 D0 A5 06 65 FC 85 06 EF
E0A8:9D 3C D0 88 D0 EA A5 5D C5
E0B0:C5 06 90 08 D0 11 A5 05 84
E0B8:C5 5C 90 0B A2 12 2C A2 8F
E0C0:0E 2C A2 16 86 D8 60 A4 47
E0C8:02 F0 38 88 88 88 84 02 C8
E0D0:C8 C8 A2 00 B1 03 C9 02 AC
E0D8:B0 E5 95 58 88 B1 03 F0 A6
E0E0:DE 95 59 88 B1 03 F0 D7 7E
E0E8:38 E9 02 95 5A 38 75 59 D2
E0F0:D1 03 90 CB 95 59 E8 E8 F7
E0F8:E8 88 10 D8 A9 1F A0 E2 CE
E100:4C 0F E1 A9 F5 A0 E1 A6 21
E108:93 F0 04 A9 0A A0 E2 8D 13
E110:64 E1 8C 65 E1 8D 82 E1 92
E118:8C 83 E1 A2 01 BD 00 D0 39
E120:85 49 BD 14 D0 85 4A BD E0
E128:28 D0 85 FB BD 3C D0 85 AA
E130:FC 20 EA E1 90 04 CA D0 F0
E138:E4 60 A5 FB 85 47 A5 FC 8C
E140:85 48 A0 02 B1 47 99 FD E8
E148:00 88 10 F8 30 0B 18 A5 43
E150:49 69 03 85 49 90 02 E6 43
E158:4A A0 02 B1 49 99 4B 00 0C
E160:88 10 F8 20 F5 E1 90 E6 CC
E168:A5 FB E9 03 85 FB B0 02 EA
E170:C6 FC 20 EA E1 B0 1F A0 3A
E178:02 B1 FB 99 4B 00 88 10 3E
E180:F8 20 F5 E1 B0 E2 A0 02 F9
E188:B1 49 91 FB B9 4B 00 91 F5
E190:49 88 10 F4 30 B8 A0 02 14
E198:B1 49 91 47 B9 FD 00 91 85
E1A0:49 88 10 F4 18 BD 00 D0 05
E1A8:7D 28 D0 85 FB BD 14 D0 77
E1B0:7D 3C D0 85 FC 66 FC 66 96
E1B8:FB 20 EA E1 B0 16 BD 00 57
E1C0:D0 9D 01 D0 BD 14 D0 9D FE
E1C8:15 D0 20 6B E2 E8 20 7B 7C
E1D0:E2 4C 1D E1 BD 28 D0 9D A8
E1D8:29 D0 BD 3C D0 9D 3D D0 29
E1E0:20 7B E2 E8 20 6B E2 4C 3F
E1E8:1D E1 A5 4A C5 FC D0 04 D4
E1F0:A5 49 C5 FB 60 A0 FF C8 A0
E1F8:C4 4B B0 0B C4 FD B0 06 3E
E200:B1 4C D1 FE F0 F1 60 C4 B0
E208:FD 60 A0 FF C8 C4 FD B0 FE
E210:0B C4 4B B0 06 B1 FE D1 C7
E218:4C F0 F1 60 C4 4B 60 86 1F
E220:93 A6 02 B5 59 85 57 B5 3A
E228:5A A8 B5 58 D0 23 C8 C4 EA
E230:57 B0 17 C4 4B B0 0C C4 F6
E238:FD B0 0C B1 4C D1 FE F0 5E
E240:ED D0 04 C4 FD B0 03 A6 5D
E248:93 60 CA CA CA 10 D4 30 66
E250:F6 C8 C4 57 B0 F4 C4 FD B2
E258:B0 0C C4 4B B0 E9 B1 FE 56
E260:D1 4C F0 ED D0 E1 C4 4B 02
E268:4C 45 E2 18 A5 49 69 03 AB
E270:9D 00 D0 A5 4A 69 00 9D 0F
E278:14 D0 60 38 A5 49 E9 03 35
E280:9D 28 D0 A5 4A E9 00 9D 2B
E288:3C D0 60 0D 00 00 00 00 7D
```

For instructions on entering the following programs please refer to "COMPUTE!'s Guide to Typing in Programs" found elsewhere in this issue.

Program 3: Mover

```
FH 5 REM COPYRIGHT 1988 COMPUT
     E! PUBLICATIONS, INC.
     {2 SPACES}ALL RIGHTS RESE
     RVED.
FX 6 PRINT"{CLR}COPYRIGHT 1988
     ":PRINT"COMPUTE! PUBLICAT
     IONS, INC."
QK 7 PRINT"ALL RIGHTS RESERVED
     ."
MX 8 FOR TT=1 TO 1500:NEXT
HP 10 REM ----- RELOCATE FSORT
       DRIVER ----
XF 20 GOTO70:REM SKIP SUBROUTI
      NE
GP 30 PRINT"{2 SPACES}NAME";
JH 40 X$="":INPUTX$:IFX$=""THE
      NPRINT"NAME REQUIRED.";:
      GOTO30
```

```
RX 50 IFLEN(X$)>16THENPRINT"NA
      ME TOO LONG.";:GOTO30
HC 60 RETURN
FM 70 N=101:U1=3:V1=1:L1=2:DIM
      A,B,U,V,X,Y,A(N),U(U1,V1
      ),L(L1,1),X$
KB 80 P=256:C0$=CHR$(.):U=59:V
      =81:XX=50
HD 90 U(.,.)=V:U(1,.)=V:U(2,.)
      =U:U(3,.)=V
BJ 100 U(.,1)=14:U(1,1)=22:U(2
       ,1)=28:U(3,1)=54
JD 110 L(.,.)={2 SPACES}828:L(
       .,1)= 1019
QC 120 L(1,.)= 2048:L(1,1)=409
       59
SX 130 L(2,.)=49152:L(2,1)=532
       47
SK 140 REM ------ READ FSORT D
       RIVER ------
DF 150 PRINT"{CLR}PLACE DISK I
       N DRIVE AND ENTER ..."
HK 160 PRINT"OLD SORT DRIVER N
       AME";:GOSUB40
FF 170 FORJ=.TO15:CLOSEJ:NEXT:
       OPEN15,8,15,"U;":OPEN8,
       8,8,"0:"+X$+",P,R"
BQ 180 INPUT#15,E,E$,ET,ES:IFE
       THENPRINTE;E$;ET;ES:CLO
       SE15:GOTO160
GX 190 GET#8,A$:IFA$=""THENA$=
       C0$
AM 200 GET#8,B$:IFB$=""THENB$=
       C0$
QB 210 LO=ASC(A$)+P*ASC(B$):PR
       INTX$" LOCATED AT"LO
PF 220 FORJ=.TON:GET#8,A$:IFA$
       =""THENA$=C0$
AC 230 A(J)=ASC(A$):NEXT:CLOSE
       8:CLOSE15
RS 240 REM -- VERIFY VALID FSO
       RT DRIVER --
DF 250 FORX=.TOU1:Y=A(U(X,1))+
       P*A(U(X,1)+1):IFY-LO=U(
       X,.)THEN270
PE 260 PRINT"ERROR"X"IN FSORT
       {SPACE}DRIVER":END
HQ 270 NEXT:IFA(N)<>96THEN260
PR 280 IFA(XX)<>141ANDA(XX)<>1
       73THEN260
RB 290 REM ----- SET OUTPUT LO
       CATION -----
GB 300 FORJ=.TOL1:PRINTL(J,.)"
       -"L(J,1)-N:NEXT
MK 310 LN$="":INPUT"NEW LOCATI
       ON";LN$:LN=VAL(LN$):J=.
DR 320 IFLN$=""THENPRINT"NUMBE
       R REQUIRED{DOWN}":GOTO3
       00
XB 330 IF(LN>=L(J,.))AND(LN+N<
       =L(J,1))THEN370
JM 340 J=J+1:IFJ>L1THENPRINT"L
       OCATION INVALID ... TRY
        ANOTHER":GOTO300
JK 350 GOTO330
DA 360 REM ----- SET SCREEN BL
       ANKING -----
RE 370 PRINT"{DOWN}SCREEN BLAN
       KING REDUCES SORT TIME
       {SPACE}BY 7%.";
DG 380 PRINT"SOME FIND THE BLA
       NK SCREEN DISQUIETING."
RQ 390 PRINT"HOWEVER, SCREEN B
       LANKING IS RECOMMENDED.
       "
XQ 400 PRINT"{3 SPACES}SCREEN
       {SPACE}BLANKING? (Y/N)"
XX 410 GETA$:IFA$<>"Y"ANDA$<>"
       N"THEN410
DX 420 A(XX)=141:IFA$="N"THENA
       (XX)=173
CM 430 REM ---- RELOCATE FSORT
        DRIVER ----
FB 440 FORJ=.TOU1:X=U(J,.)+LN:
       B=INT(X/P):A=X-B*P
ME 450 A(U(J,1))=A:A(U(J,1)+1)
       =B:NEXT
JH 460 REM ------- SET OUTPUT
       {SPACE}NAME -------
RS 470 PRINT"NEW SORT DRIVER N
       AME";:GOSUB40
AP 480 REM ------ WRITE OUTPUT
        FILE ------
MD 490 OPEN15,8,15,"I0":OPEN8,
       8,8,"@0:"+X$+",P,W":INP
       UT#15,E,E$,ET,ES
EX 500 IFETHENPRINTE;E$;ET;ES:
       PRINT"TAKE CORRECTIVE A
       CTION":CLOSE15:END
DH 510 B=INT(LN/P):A=LN-P*B:PR
       INT#8,CHR$(A);CHR$(B);
RX 520 FORJ=.TON-1:PRINT#8,CHR
       $(A(J));:NEXT:PRINT#8,C
       HR$(A(N)):CLOSE8:CLOSE1
       5:END
```

Program 4: Demo

```
FH 5 REM COPYRIGHT 1988 COMPUT
     E! PUBLICATIONS, INC.
     {2 SPACES}ALL RIGHTS RESE
     RVED.
JK 10 REM ------ DEMONSTRATE F
      SORT -------
JP 20 GOTO160:REM ENTER BOOT R
      OUTINE
SQ 30 REM --- SET BCD CLOCK=00
      :00:00.0 ---
DB 40 POKEH,.:POKEM,.:POKES,.:
      POKET,.:RETURN
GF 50 REM --- CAPTURE BCD CLOC
      K VALUE{2 SPACES}---
KP 60 H1=PEEK(H):M1=PEEK(M):S1
      =PEEK(S):T1=PEEK(T)
KM 70 H1=HH*(TN*(H1ANDSX)/SX+(
      H1ANDNL))
FC 80 M1=MM*(TN*(M1ANDNH)/SX+(
      M1ANDNL))
GJ 90 S1=SS*(TN*(S1ANDNH)/SX+(
      S1ANDNL))
CQ 100 T1=(TT*(T1ANDNL)+S1+M1+
       H1)/SS:IFT1=.THENT1=.09
MC 110 PRINTT1"SECONDS":RETURN
MP 120 REM ----- WAIT FOR KEYS
       TROKE -----
GX 130 GETA$:IFA$=""THEN130
CS 140 RETURN
KS 150 REM -------- BOOT ROUTI
       NE ---------
QK 160 ONBBGOTO190,210
MH 170 PRINT"{CLR}COPYRIGHT 19
       88":PRINT"COMPUTE! PUBL
       ICATIONS, INC."
AF 172 PRINT"ALL RIGHTS RESERV
       ED."
CA 175 PRINT"{DOWN}BOOTING ML
       {SPACE}SORT ROUTINES...
       "
MG 180 BB=1:LOAD"FSORT.DRIVER"
       ,8,1
MS 190 BB=2:LOAD"FSORT.MAIN",8
       ,1
KQ 200 REM ------- INITIALIZAT
       ION --------
EK 210 P=256:SX=16:TN=10:NL=15
       :NH=NL*SX:B$="
       {2 SPACES}":N=1000:DIMJ
       ,T,H1,M1,S1,T1,S$(N)
JH 220 H=56331:M=H-1:S=M-1:T=S
       -1:TT=1:SS=TT*10:MM=SS*
       60:HH=MM*60:CR$=CHR$(13
       )
JF 230 E=26:Z=65:Q=49152
SB 240 REM -------- BUILD ARRA
       Y ----------
RX 250 PRINTCR$"CREATING STRIN
       G ARRAY OF"N"ELEMENTS"
JK 260 PRINTCR$"PLEASE WAIT".0
       27*N"SECONDS"
PA 270 FORJ=1TON:S$(J)=CHR$(RN
       D(.)*E+Z)+CHR$(RND(.)*E
       +Z)+CHR$(RND(.)*E+Z):NE
       XT
BS 280 REM ----------- ML SORT
        -----------
RG 290 PRINTCR$"HIT ANY KEY TO
        SORT"N"ITEMS":GOSUB130
RC 300 PRINTCR$"ML SORTING ...
       ";
XH 310 GOSUB40:SYSQ,1,N,S$(1):
       GOSUB60
DR 320 REM -------- SLOW RE-SO
       RT ---------
GP 330 PRINTCR$"THE ARRAY IS N
       OW SORTED.{2 SPACES}A R
       E-SORT{5 SPACES}WOULD B
       E VERY SLOW";
XJ 340 PRINT" BECAUSE THE 'QUI
       CK{3 SPACES}SORT' ALGOR
       ITHM IS FAST ONLY FOR "
QG 350 PRINT"RANDOMLY SEQUENCE
       D ARRAYS.{2 SPACES}DEMO
       :"
SJ 360 PRINTCR$"HIT ANY KEY TO
        RE-DO SORT THE SLOW WA
       Y":GOSUB130
BQ 370 PRINTCR$"ML SORTING ...
       ";
AQ 380 GOSUB40:SYSQ,1,N,S$(1):
       GOSUB60
FQ 390 REM -------- FAST RE-SO
       RT ---------
AJ 400 PRINTCR$"THE ARRAY IS N
       OW SORTED AGAIN.
       {2 SPACES}NOW{5 SPACES}
       LET'S RE-SORT USING";
QK 410 PRINT" THE FASTER 'DOUB
       LE{3 SPACES}SORT' METHO
       D TO RANDOMIZE THE ARRA
       Y "
GB 420 PRINT"AND THEN PROPERLY
        SORT.{3 SPACES}DEMO:"
MH 430 PRINTCR$"HIT ANY KEY TO
        RE-DO SORT THE FAST WA
       Y":GOSUB130
AJ 440 PRINTCR$"ML SORTING ...
       ";
MS 450 GOSUB40:SYSQ,2,N,S$(1):
       SYSQ,1,N,S$(1):GOSUB60
DH 460 REM -------- VERIFY SOR
       T ----------
JB 470 PRINTCR$"DONE"CR$CR$"HI
       T ANY KEY TO SHOW SORTE
       D STRINGS":GOSUB130
FA 480 PRINTCR$"NOW VERIFYING
       {SPACE}SORT"CR$
KP 490 FORJ=1TON:PRINTB$S$(J);
       :IFS$(J)<S$(J-1)THENPRI
       NTCR$"SEQUENCE ERROR":S
       TOP
JK 500 NEXT:PRINTCR$"VERIFICAT
       ION COMPLETE"
QB 510 REM -------- FIELD SORT
        ----------
MK 520 PRINTCR$"HERE'S A DEMO
       {SPACE}OF FIELD SORT, W
       HICH{6 SPACES}SEQUENCES
        THE ARRAY ";
HJ 530 PRINT"AS FOLLOWS:"CR$
DQ 540 PRINT"{4 SPACES}POS LEN
        DIRECTION"
JH 550 PRINT"{4 SPACES}=== ===
       =========="
```

```
CJ 56Ø PRINT"{5 SPACES}3
       {3 SPACES}1{2 SPACES}AS
       CENDING"
JP 57Ø PRINT"{5 SPACES}2
       {3 SPACES}1{2 SPACES}DE
       SCENDING"
EJ 58Ø PRINT"{5 SPACES}1
       {3 SPACES}1{2 SPACES}AS
       CENDING"
EH 59Ø CØ$=CHR$(.):C1$=CHR$(1)
       :C2$=CHR$(2):C3$=CHR$(3
       )
HK 6ØØ F1$=C3$+C1$+CØ$:F2$=C2$
       +C2$+C1$:F3$=C1$+C1$+CØ
       $
AQ 61Ø PRINTCR$"HIT ANY KEY TO
        FIELD SORT THE ARRAY":
       GOSUB13Ø
BX 62Ø PRINTCR$"ML SORTING ...
       ";
QG 63Ø GOSUB4Ø:SYSQ,F1$+F2$+F3
       $,N,S$(1):GOSUB6Ø
QF 64Ø REM ------- VERIFY FSOR
       T ----------
MK 65Ø PRINTCR$"DONE"CR$CR$"HI
       T ANY KEY TO SHOW SORTE
       D STRINGS"CR$:GOSUB13Ø
PM 66Ø FORJ=1TON:PRINTB$S$(J);
       :NEXT
HG 67Ø PRINTCR$"LIST COMPLETE
       {SPACE}- DEMO DONE"
```

Program 5: Dates

```
FH 5 REM COPYRIGHT 1988 COMPUT
     E! PUBLICATIONS, INC.
     {2 SPACES}ALL RIGHTS RESE
     RVED.
PX 1Ø REM -------- BIRTHDAY LI
      STER -------
KJ 2Ø ONLDGOTO4Ø,2ØØ
ER 22 PRINT"{CLR}COPYRIGHT 198
      8":PRINT"COMPUTE! PUBLIC
      ATIONS, INC."
EQ 25 PRINT"ALL RIGHTS RESERVE
      D."
PF 3Ø LD=1:LOAD"FSORT.DRIVER",
      8,1
RB 4Ø LD=2:LOAD"FSORT.MAIN",8,
      1
BG 5Ø REM -------- DATE VALIDA
      TION -------
JK 6Ø E=.:IFM*D*YTHEN9Ø
GP 7Ø IFM+D+Y=.THENE=2:GOTO13Ø
JE 8Ø E=1:GOTO13Ø
GB 9Ø IFM>12ORM<1THEN8Ø
AX 1ØØ IFD>D(M-1)ORD<1THEN8Ø
HQ 11Ø IFM<>2ORD<>29THEN13Ø
SH 12Ø IF4*INT(Y/4)<>YTHEN8Ø
SS 13Ø RETURN
BF 14Ø REM ------- I/O VALIDAT
       ION --------
SC 15Ø E1=ST:INPUT#15,E,E$:IFE
       THEN17Ø
KQ 16Ø E=E1:E$="END OF FILE"
XR 17Ø IFETHENPRINTE;E$
BB 18Ø RETURN
HQ 19Ø REM --------- MAIN LINE
        -----------
BP 2ØØ MX=1ØØØ:DIMD$(MX),D(11)
       ,M,D,Y,A:H=1ØØØØ:SL$="/
       ":NL$="":Q=49152
PA 21Ø CØ$=CHR$(.):C1$=CHR$(1)
       :CR$=CHR$(13)
PC 22Ø F1$=C1$+CHR$(5)+CØ$:F2$
       =CHR$(7)+CHR$(2)+CØ$:F3
       $=CHR$(1Ø)+CHR$(99)+CØ$
JR 23Ø S$(1)=F2$+F1$+F3$:S$(2)
       =C1$+CHR$(255)+CØ$:S$(3
       )=F3$+F1$+F2$
FF 24Ø PRINT"{CLR}"SPC(11)"
       {WHT}{RVS} BIRTHDAY LIS
       TER {OFF}"CR$
ME 25Ø PRINT"THIS PROGRAM WILL
        ALLOW YOU TO KEEP A
        {3 SPACES}LIST OF BIRTH
        DATES (OR ANY ";
RB 26Ø PRINT"DATES) ON
       {4 SPACES}DISK.
       {2 SPACES}THE LIST CAN
       {SPACE}BE PRESENTED IN
       {SPACE}ANY OF THREE ";
EC 27Ø PRINT"SEQUENCES OFFERED
       .{2 SPACES}THE LIST
       {3 SPACES}CAN BE DIRECT
       ED TO THE SCREEN ";
EE 28Ø PRINT"OR THE{4 SPACES}P
       RINTER."CR$
AJ 29Ø PRINT"WHILE USING THIS
       {SPACE}PROGRAM, AVOID C
       APITAL LETTERS.
       {2 SPACES}ALL LOWER-CAS
       E ";
AR 3ØØ PRINT"LETTERS SORT
       {3 SPACES}AHEAD OF ALL
       {SPACE}UPPER-CASE LETTE
       RS."CR$
AX 31Ø PRINT"WHEN ADDING NEW D
       ATES AND NAMES, PROVIDE
        THE EXACT NUMBER OF ";
KQ 32Ø PRINT"COMMAS"CR$"REQUES
       TED.{3 SPACES}{RVS} HAV
       E FUN! {OFF}"CR$
XA 33Ø TT$="<-DATE-> <--------
       -- NAME ---------->"
FS 34Ø BL$="{38 SPACES}"
CM 35Ø FF=255:FF$=CHR$(FF):FOR
       J=1TO4:FF$=FF$+FF$:NEXT
SM 36Ø FORJ=.TO11:READD(J):NEX
       T:GOTO38Ø:DATA 31,29,31
       ,3Ø,31,3Ø,31,31,3Ø,31,3
       Ø,31
FH 37Ø REM ------ RETRIEVE OLD
        DATA ------
GA 38Ø PRINT"SHALL WE LOAD AN
       {SPACE}OLD DATA FILE? (
       Y/N)"CR$
AQ 39Ø GETA$:IFA$<>"Y"ANDA$<>"
       N"THEN39Ø
SH 4ØØ IFA$="N"THEN53Ø
FC 41Ø F$=NL$:INPUT"ENTER FILE
       NAME";F$
SM 42Ø L=LEN(F$):IFL=.ORL>16TH
       ENPRINT"{2 SPACES}{RVS}
        FILENAME ERROR ... RET
       RY {OFF}":GOTO38Ø
GJ 43Ø PRINT"PUT DISK IN DRIVE
        - HIT RETURN"
HR 44Ø GETA$:IFA$<>CR$THEN44Ø
CQ 45Ø CLOSE15:OPEN15,8,15,"IØ
       ":GOSUB15Ø:IFETHEN38Ø
ED 46Ø CLOSE8:OPEN8,8,8,"Ø:"+F
       $+",S,R":GOSUB15Ø:IFETH
       EN38Ø
CQ 47Ø I=I+1:INPUT#8,D$(I):IFD
       $(I)=""THENI=I-1
CJ 48Ø GOSUB15Ø:PRINTD$(I):IFE
       THENCLOSE8:CLOSE15:GOTO
       5ØØ
JF 49Ø GOTO47Ø
GK 5ØØ IFITHENSQ=2:GOTO53Ø
PJ 51Ø PRINT" {RVS} EMPTY FILE
       {OFF}":GOTO38Ø
MX 52Ø REM -------- CHOOSE OPT
       ION --------
CA 53Ø REM IFI=.THEN77Ø
KE 54Ø PRINT"{CLR}{3 SPACES}
       {RVS}L{OFF}IST, {RVS}F
       {OFF}IX, {RVS}A{OFF}DD,
        {RVS}S{OFF}AVE DATA OR
        {RVS}E{OFF}ND?"
FG 55Ø GETA$:K=6:FORJ=1TO5:IFM
       ID$("LFASE",J,1)=A$THEN
       K=J:J=5
FB 56Ø NEXT:ONKGOTO58Ø,91Ø,76Ø
       ,13ØØ,125Ø,55Ø
DX 57Ø REM ---------- SORT DAT
       A ----------
RJ 58Ø PRINT"{DOWN}{2 SPACES}S
       ORT BY {RVS}Y{OFF}EAR,
       {SPACE}{RVS}M{OFF}ONTH
       {SPACE}OR {RVS}N{OFF}AM
       E?":IFI=.THEN91Ø
GJ 59Ø GETA$:K=.:FORJ=1TO3:IFM
       ID$("YMN",J,1)=A$THENK=
       J:J=3
BQ 6ØØ NEXT:IFK=.THEN59Ø
AS 61Ø IFK=SQTHENSYSQ,5,I,D$(1
       )
JF 62Ø SYSQ,S$(K),I,D$(1):SQ=K
XH 63Ø IFFGTHENFORJ=ITOI-FG+1S
       TEP-1:D$(J)="":NEXT:I=I
       -FG:FG=.
XP 64Ø PRINT"{DOWN}{6 SPACES}
       {RVS}S{OFF}CREEN OR
       {RVS}P{OFF}RINT?"
RX 65Ø GETA$:K=3:FORJ=1TO2:IFM
       ID$("SP",J,1)=A$THENK=J
       :J=2
GQ 66Ø NEXT:ONKGOTO91Ø,68Ø,65Ø
RC 67Ø REM ------- LIST TO PRI
       NTER -------
XR 68Ø PRINT"{DOWN}SETUP PRINT
       ER - THEN HIT RETURN"
BG 69Ø GETA$:IFA$<>CR$THEN69Ø
SR 7ØØ OPEN4,4:PG=6Ø
BD 71Ø FORJ=1TOISTEP2*PG:FORK=
       1TO2:PRINT#4,"
       {3 SPACES}"TT$;:NEXT:PR
       INT#4,CR$
JD 72Ø FORK=.TOPG-1:IFD$(J+K)=
       ""THENPRINT#4:GOTO74Ø
FQ 73Ø PRINT#4,"{3 SPACES}"D$(
       J+K)LEFT$(BL$,39-LEN(D$
       (J+K)))" "D$(J+K+PG)
JR 74Ø NEXT:PRINT#4,CR$CR$CR$:
       NEXT:CLOSE4:GOTO53Ø
AA 75Ø REM ------- ENTER NEW D
       ATA --------
PS 76Ø M=.:D=.:Y=.:INPUT"
       {DOWN}ENTER DATE {RVS}M
       O,DY,YR{OFF}{2 SPACES}(
       '{RVS}Ø,Ø,Ø{OFF}' TO QU
       IT)";M,D,Y
HM 77Ø M=INT(M):D=INT(D):Y=INT
       (Y):GOSUB6Ø:ONEGOTO82Ø,
       53Ø:I=I+1
PC 78Ø D$(I)=RIGHT$(STR$(M+H),
       2)+SL$+RIGHT$(STR$(D+H)
       ,2)+SL$+RIGHT$(STR$(Y+H
       ),2)
XJ 79Ø N1$=NL$:N2$=NL$:N3$=NL$
       :PRINT"ENTER {RVS}LAST-
       NAME,FIRST-NAME MIDDLE-
       INIT{OFF}";
FG 8ØØ N1$="":N2$="":INPUTN1$,
       N2$:L1=LEN(N1$):L2=LEN(
       N2$):IFL1+L2THEN83Ø
RS 81Ø PRINT"{RVS} INVALID NAM
       E {OFF}{DOWN}":GOTO79Ø
EG 82Ø PRINT"{RVS} INVALID DAT
       E {OFF}{DOWN}":GOTO76Ø
QG 83Ø IFL1+L2>25THEN81Ø
BB 84Ø D$(I)=D$(I)+" ":IFL1THE
       ND$(I)=D$(I)+N1$:IFL2TH
       END$(I)=D$(I)+" - "
KA 85Ø IFL2THEND$(I)=D$(I)+N2$
DH 86Ø PRINTD$(I)"{DOWN}":PRIN
       T"IS THIS CORRECT (Y/N)
       ?";
AA 87Ø GETA$:IFA$<>"Y"ANDA$<>"
       N"THEN87Ø
```

```
EE  880 IFA$="N"THENI=I-1:PRINT
        "{2 SPACES}{RVS} RETRY
        {SPACE}{OFF}";
RH  890 PRINT:GOTO760
EH  900 REM ----- REPAIR CURREN
        T DATA -----
FA  910 PG=20:FORJ=1TOISTEPPG
MP  920 PRINT"{CLR}{RVS} LIST/F
        IX DATA {OFF}"SPC(8);
FJ  930 PRINT"PAGE"INT((J+PG)/P
        G)"OF"INT((I+PG-1)/PG)C
        R$"{2 SPACES}"TT$"
        {DOWN}"
DC  940 FORK=.TOPG-1:IFJ+K>ITHE
        NPRINT:GOTO980
KA  950 PRINT"{YEL}"CHR$(65+K)"
        {WHT}{SHIFT-SPACE}";:L=
        K
FQ  960 IFASC(D$(J+K))<>FFTHENP
        RINTD$(J+K):GOTO980
MK  970 PRINT" *** DELETED ***"
PF  980 NEXT:PRINT"↑"CR$"{Z}=ED
        IT{3 SPACES}{RVS}+{OFF}
        =FORWARD{2 SPACES}{RVS}
        -{OFF}=BACKWARD
        {3 SPACES}{RVS}0{OFF}=Q
        UIT";
JH  990 REM NEXT:PRINT"↑"CR$"
        {Z}{RVS}{YEL}ALPHA{WHT}
        {OFF}=EDIT {RVS}0{OFF}=
        QUIT {RVS}+{OFF}=FORWAR
        D {RVS}-{OFF}=BACKWARD"
        ;
GE 1000 GETA$:IFA$=""THEN1000
ER 1010 IFIANDA$="+"THEN1220
QX 1020 IFIANDA$="-"THENJ=J-PG
        :GOTO1200
XE 1030 IFA$="0"THENJ=I:GOTO12
        10
KA 1040 IFA$<"A"ORA$>CHR$(65+L
        )ORI=.THEN1000
AS 1050 A=J+ASC(A$)-65:IFASC(D
        $(A)+C0$)=FFTHEN1000
XC 1060 PRINT"{CLR}";
MM 1070 PRINT"{2 SPACES}{RVS}
        {SPACE}TO FIX, OVER-TY
        PE & HIT RETURN {OFF}"
JF 1080 PRINT"{5 DOWN}
        {2 SPACES}{RVS} TO DRO
        P, ENTER - & HIT RETUR
        N {OFF}{6 UP}"CR$
PS 1090 PRINTBL$BL$BL$"{HOME}
        {2 DOWN}{2 SPACES}"D$(
        A)"{HOME}{DOWN}":D$=""
        :INPUTD$
SD 1100 IFLEFT$(D$,1)="-"THEND
        $=FF$:FG=FG+1:GOTO1190
AA 1110 IFLEN(D$)>37ORLEN(D$)<
        11THENPRINT"{DOWN}LEN
        {SPACE}ERROR";:GOTO118
        0
RM 1120 IFLEFT$(D$,9)=LEFT$(D$
        (A),9)THEN1190
QE 1130 IFMID$(D$,3,1)="/"ANDM
        ID$(D$,6,1)="/"ANDMID$
        (D$,9,1)=" "THEN1150
GR 1140 PRINT"{DOWN}FORMAT ERR
        OR";:GOTO1180
DC 1150 M=VAL(LEFT$(D$,2)):D=V
        AL(MID$(D$,4,2)):Y=VAL
        (MID$(D$,7,2))
RM 1160 GOSUB60:IFE=.THEN1190
QG 1170 PRINT"{DOWN}DATE ERROR
        ";
PE 1180 PRINT" ... RETRY{HOME}
        {2 DOWN}"BL$"{HOME}";:
        GOTO1070
PE 1190 D$(A)=D$
PR 1200 IFJ>.THENJ=J-PG
RJ 1210 NEXT:PRINT:GOTO530
BJ 1220 IFJ+PG>ITHENJ=J-PG
DR 1230 GOTO1210
RM 1240 REM ----- SAVE CURRENT
         DATA ------
DS 1250 PRINT"DO YOU WANT TO S
        AVE DATA FILE? (Y/N)":
        IFI=.THENPRINT"{CLR}":
        GOTO1280
CA 1260 GETA$:IFA$<>"Y"ANDA$<>
        "N"THEN1260
HC 1270 IFA$="Y"THEN1300
MG 1280 IFK=4THEN530
XB 1290 END
DC 1300 F$=NL$:IFITHENINPUT"EN
        TER FILENAME";F$
AG 1310 L=LEN(F$):IFL=.ORL>16T
        HENPRINT"{2 SPACES}
        {RVS} FILENAME LEN ERR
        OR...RETRY {OFF}":GOTO
        1250
PM 1320 PRINT"PUT DISK IN DRIV
        E - HIT RETURN"
AH 1330 GETA$:IFA$<>CR$THEN133
        0
EE 1340 CLOSE15:OPEN15,8,15,"I
        0":GOSUB150:IFETHEN125
        0
DG 1350 CLOSE8:OPEN8,8,8,"@0:"
        +F$+",S,W":GOSUB150:IF
        ETHEN1250
JB 1360 IFSQ=2THENSYSQ,5,I,D$(
        1)
CJ 1370 SQ=2:SYSQ,S$(SQ),I,D$(
        1)
MJ 1380 FORJ=1TOI:PRINT#8,D$(J
        ):GOSUB150:NEXT:CLOSE8
        :CLOSE15
PJ 1390 PRINT" {RVS} FILE SAVE
        D {OFF}":FORJ=1TO1000:
        NEXT:IFK=4THEN530
```

Two 64 Emulators For The Amiga

Rhett Anderson And Randy Thompson

Commodore 64 emulators for the Amiga have been rumored for the past year or so. Glowing reports of these emulators have attracted a lot of interest in the Commodore community. We recently received two such products at COMPUTE!. Here are our first impressions.

Commodore 64 emulators for the Amiga are a hot topic. We recently obtained release copies of two such products for evaluation: *The 64 Emulator*, from ReadySoft; and *GO-64!*, from Software Insights Systems. This is not a product review—it is more of a preview. Both companies are still working on their software, and updates are sure to come.

Pros And Cons Of Emulation

Why a 64 emulator for the Amiga? The most obvious answer: Many 64 owners who upgrade to an Amiga are reluctant to lose their 64 software and hardware investments. Some popular 64 programs are simply not available for the Amiga, and many never will be. Also, emulation in general is intrinsically interesting—seeing the 64 startup message on the Amiga is fascinating.

On the other hand, why cripple the Amiga by turning it into a 64? Emulators are notoriously slow, and no emulator can offer 100 percent compatibility. But, if you sold your 64 in order to buy an Amiga, or if you just want some connection between your two computers, however tenuous, an emulator may prove a practical purchase.

Both emulators come with a hardware device that allows you to connect a 1541/1571 disk drive to your Amiga via the parallel port. (Actually, this hardware device is optional on *The 64 Emulator*, but highly recommended.) If you own an Amiga 1000, the *GO-64* emulator requires a special cable that converts your parallel port to be compatible with the 500's and the 2000's. There are two different disk drive interfaces for *The 64 Emulator*—one for the 1000 and one for the newer models. You'll need this hardware device in order to load 64 software from disk. Neither of these products are able to load 64 software from the Amiga's 5¼-inch 1010 drive.

A real 64 does its magic with special hardware. The brain of the 64 is the 6510 microprocessor. The special VIC and SID chips control the video and sound effects that nearly every 64 program uses.

Essentially, these emulators are 64s on a disk. In conjunction with your Amiga computer, they translate each instruction and video and audio command into a job for the Amiga. This translation takes a great deal of time. The 68000 microprocessor in the Amiga is far faster and more powerful than the 6510 in the 64, but it's not up to the task of playing the part of a real 64 at full speed.

How Fast Do They Go?

We wrote several short programs to test the speed of the emulators. Here are the results (all times are in seconds):

	64	*64 Emulator*	*GO-64!*
Test 1	10	49	22
Test 2	44	69	93
Test 3	9	42	28
Test 4	11	48	19
Test 5	5	19	51

Test 1 was an empty BASIC loop counting to 10,000; Test 2 tested text-screen I/O; Test 3 moved a sprite around the borders of the screen; various math functions were tested by Test 4; and Test 5 was a machine language do-nothing loop.

The benchmarks tell a strange story. *The 64 Emulator* beats the *GO-64!* emulator in two tests, and loses the other three. The 64 computer itself is the only clear winner here. We had trouble with Benchmark 3 using *GO-64!*—we had to put a PRINT statement in our program before the sprite would appear.

Although the *GO-64!* emulator wins three out of five of our benchmarks, we found that most commercial programs run faster on *The 64 Emulator*.

Other Considerations

The *GO-64!* emulator does not yet support the SID chip. *The 64 Emulator* does, but at the slower executions speeds, a cheerful march becomes a dirge.

Sprites on *The 64 Emulator* are handled through use of the Amiga's virtual sprite system. This can cause the sprites to flicker when more than four sprites are displayed horizontally. Because *GO-64!* uses bobs (blitter objects, which are directly drawn on the screen) to emulate sprites, it does not have this problem. There is a price to pay for using bobs—vsprites are faster. The *GO-64!* screen seems to jump instead of moving smoothly.

The way in which these emulators handle the keyboard is quite different. While *GO-64!* tries its best to duplicate the 64's keyboard (a quote is still SHIFT-2 and cursor keys must be shifted to move the cursor up or left), *The 64 Emulator* uses the Amiga's keyboard layout. We found it very frustrating on the *GO-64!* emulator to press right-bracket to get an asterisk. People who learned to type on the 64, however, may feel right at home. The *GO-64!* had trouble keeping up with fast typing.

A Better 64?

One of the advantages of the Amiga over the 64 is that it uses an RGB monitor, which provides better resolution and color than the composite monitors used by the 64. Both 64 emulators let you take advantage of your Amiga screen to see 64 graphics as they've never been seen before. If you have a favorite piece of artwork on your 64, load it up on a 64 emulator and you'll be pleasantly suprised.

The 64 Emulator has a control panel with some interesting options. You can choose to use your Amiga mouse to emulate a 1530 or 1531 mouse for your 64. In addition, you can use your ramdisk and Amiga disk drives from 64 mode with *The 64 Emulator*. This is great for transferring text files from one computer to another. *The 64 Emulator* is also able to emulate the 1764 ram expansion unit.

Compatibility

We tested several popular programs on both emulators with mixed results. Both products were able to run *GEOS*, with *The 64 Emulator* running it a bit faster. *SpeedScript* ran on *The 64 Emulator*, but not on *GO-64!*. "MetaBASIC," COMPUTE!'s popular BASIC programming utility, would not run on either. If you have specific software that you want to run on your emulator, be sure to try it out at a computer store first.

To ensure the highest possible compatibility, producers of the *GO-64!* emulator suggest that you transfer the ROMs from a 64 to your *GO-64!* disk. This process requires telecommunications software and modems for each computer. We used the 64 ROMs for all of our tests. If you choose not to copy the ROMs, the emulator will still work.

GO-64!
Software Insight Systems
16E International Dr.
East Granby, CT 06026
$69.95

The 64 Emulator
Readysoft
P.O. Box 1222
Lewiston, NY 14092
$39.95
$59.95 for serial interface

The 128/MS-DOS Connection

Ernest R. Hunter

If you're a 128 user who works with an IBM PC at your office, or if you have a friend with a PC and you would like to share files, then this program is for you. With it, you can use your 128 and 1571 disk drive to read and write to disks for MS-DOS 2.0 or higher.

Many Commodore 128 owners use an IBM PC at their office but find that they can't take work home because the 128 can't read the PC's disks. You run into the same problem if you want to share some of your 128 files with a friend who uses a PC—the disks are incompatible. "MS Read 128," the program accompanying this article, breaks this language barrier between the 128 and MS-DOS.

In conjunction with a 1571 disk drive, MS Read 128 will read and write to 5¼-inch floppy disks in Commodore or MS-DOS format, and it will automatically translate to and from true ASCII and Commodore ASCII. MS Read 128 uses machine language routines and the 1571's burst mode so it is fast: It will copy a complete MS-DOS disk in less than four minutes.

Despite all of its features, MS Read 128 can't perform miracles. It allows you to read and manipulate *text files* from MS-DOS disks, but the program *will not* allow you to run programs written for the IBM PC on your 128. The 128 simply doesn't have the hardware to handle PC software, and no program can change that fact. Furthermore, many PC programs store their data files in special formats that MS Read 128 may not be able to understand. MS Read 128 is designed for use with pure ASCII text files. Check the manual for your PC software to determine whether it provides the option of reading or writing files of this type.

Getting Started

MS Read 128 consists of a BASIC program, MS READ 128 (Program 1), and a machine language program, CBM/MSDOS.ML (Program 2). Enter both programs and save them on the same disk.

Because CBM/MSDOS.ML (Program 2) is written entirely in machine language, it must be entered using the "128 MLX" machine language entry program found elsewhere in this issue. Be sure to read the instructions for using MLX before you begin entering data. When you run MLX, you'll be asked for a starting and an ending address for the data you'll be entering. The correct values for CBM/MSDOS.ML are as follows:

Starting address: 0B90
Ending address: 0DC7

After you've entered all the data, be sure to save a copy before leaving MLX.

You may use any filename you want for Program 1, but the filename for Program 2, the machine language program, must agree with the name in line 30 of Program 1. If you don't save Program 2 as CBM/MSDOS.ML, you'll need to change line 30 to match the name you used. To get MS Read 128 going, just load and run it. Be sure CBM/MSDOS.ML is on the same disk. You'll be reminded to insert the first disk you want to work with, and you'll be asked whether you want to use a 40- or an 80-column display. Then you'll be presented with a list of MS Read 128's commands. Before discussing the commands, MS Read 128 has a few limitations you should note.

MS Read 128 will work only with MS-DOS disks formatted single-sided with 9 sectors per track. MS-DOS versions 2.0 or above can produce such disks when used with a 360K disk drive. The 1.2-megabyte drives supplied with the PC AT and compatibles cannot produce this format. You should also be aware that current versions of MS-DOS format disks double-sided by default. Because of this, you must specially prepare disks on your MS-DOS sustem so that they can be read by MS Read 128. For the standard MS-DOS FORMAT command, you must add the /1 option to indicate that the disk is to be formatted single-sided.

MS Read 128 will read the directory of any MS-DOS disk and give you a warning if the format is one to which MS Read 128 cannot write. Attempting to *save, delete,* or *dcopy* on a disk that is not single-sided, 9-sectors-per-track, 512-bytes-per-sector will cause the message ***WARNING** MSDOS FORMAT NOT COMPATIBLE* to appear on the screen. But don't worry—no damage will have been done to the data currently on the disk. If you are not sure of the MS-DOS format on the disk, use MS Read 128's *regdsk* command. If you do not receive a warning, the disk's format is fine. MS Read 128 will read only the root directory of the MS-DOS disk, but this will not present a problem for most users.

MS Read 128 can read any type of MS-DOS file into memory and can save it out again as a Commodore SEQ file, but the primary purpose of the program is to transfer text files from word processors, spreadsheets, database programs, and other applications. The 128 and the PC use different microprocessors so programs for one won't run on the other, but text files should present few problems.

Commands

MS Read 128 has the following commands: *regdsk, dir, list, help, quit, load, save, delete, dspec, ldir, plist, errdsk, mformat, cformat, dcopy,* and *scrdsk.* Both commands and parameters should be entered in *lowercase.* You may execute any of the commands by entering only the first two characters. For example, *load testfile* and *lo testfile* will both load the file named *testfile* into memory. Some commands will ask you to press C to continue; pressing any other key will abort the command. An explanation of each command follows.

- **regdsk** determines whether the format of the disk in the drive is the one to which MS Read 128 can write. You'll receive a warning if the format is not the one needed. This command is automatically executed by *dir, load, save, dcopy, dspec,* and *scrdsk.* If the disk is of an acceptable type, *regdsk* will cause the prompt to display *CBM disk* or *MSDOS disk* depending on the type of disk in the drive.
- **dir** reads and displays the directory of the disk currently in the drive. The command works for both Commodore and MS-DOS disks.
- **list** displays the file currently in memory. You can also use the syntax **list** *xx yy* or **list** *xx* to display only a portion of the file. The *xx* and *yy* values are numbers between 0 and 100 and define the portion of the file to be listed. For example, *list 20.5 60.6* will list from the 20.5 percent point to the 60.6 percent point of the file in memory. The command *list 1 50* will list the first 50 percent of the file, and *list 50* will list the last half (from the 50 percent point to the end). Be sure to place a space between the command and the parameter(s).
- **help** simply displays a complete list of MS Read 128's commands on the screen.
- **quit** exits MS Read 128 and returns you to BASIC. MS Read 128 is still in memory, however, and typing CONT will restart it.
- **load** *filename* loads the specified file from the disk into memory. You'll see the file scroll across the screen as it is being loaded. You can pause this scrolling with the NO SCROLL key. MS Read 128 will create both a true ASCII and a Commodore ASCII version of the file and will automatically use the correct one based on the disk in the drive when you execute the save command. The size of the file must not exceed 44544 bytes, roughly 174 Commodore disk blocks or 87 MS-DOS sectors. MS Read 128 does not check for file size before it loads, so be sure you don't try to load a file that exceeds these limits. If you do, you'll get unpredictable results.
- **save** *filename* saves the file in memory to disk using the specified filename. MS Read 128 checks to see which kind of disk is in the drive and automatically picks the correct copy of the file to save (true ASCII for MS-DOS or Commodore ASCII for Commodore format). Commodore files are stored in sequential (SEQ) format. Thus, you can transfer between formats by loading a file from an MS-DOS disk and saving it to a Commodore disk, or by loading from a Commodore disk and saving to an MS-DOS disk. Keep in mind the respective restrictions on filenames: Commodore filenames can be no more than 16 characters long, while MS-DOS filenames are limited to eight characters, plus an optional three-character extension. If the extension is used, it should be separated from the filename with a period.
- **delete** *filename* scratches or deletes the specified file from the disk.
- **dspec** registers the disk by executing *regdsk;* then it prints the disk format information to the screen.
- **ldir** lists the last directory read into memory. If the last disk read was MS-DOS, then the disk will not be read again. If the last disk read was Commodore, however, this command will read the disk currently in the drive.
- **plist** lists the file currently in memory to the printer. It can take the same parameters as list, so you can use **plist** *xx yy* or **plist** *xx* to display only a specified portion of the file.
- **errdsk** reads and displays the drive's error channel. Use this command to determine the problem when the drive's error light is flashing.
- **mformat** *volumename* formats a disk for MS-DOS files. The *volumename* parameter can be omitted if no volume name is desired. If specified, the volume name should be no more than 11 characters long. You will need a disk formatted with MS-DOS to get started. MS Read 128 copies the disk control area from this disk into memory and writes the information to the blank disk you wish to format.
- **cformat** *diskname id* formats a disk for files.
- **dcopy** copies from one MS-DOS disk to another MS-DOS disk. You'll be given the opportunity to format the target disk if it isn't already formatted.
- **scrdsk** scratches all the files on an MS-DOS disk. To scratch the files *and* change the volume name, use the syntax **scrdsk** *volumename.* This command is faster than reformatting when you want to clear all the files from an MS-DOS disk.

Program 1: MS Read 128

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
GC 10 POKE57,0:POKE58,57:POKE4
      8,17:CLR:M0=57*256:TB=57
      +24:FORJ=1TO60:BL$=BL$+C
      HR$(32):NEXT
```

```
KC 20 PRINTCHR$(14)"{RVS}SELEC
      T ONE":PRINT"A - 80 COL
      {SPACE}MONITOR":PRINT"B
      {SPACE}- 40 COL":GETKEYA
      $:IFA$<>"B"THENFAST:A=20
BJ 30 IFPEEK(2960)<>133ORPEEK(
      2961)<>250THENBANK1:BLOA
      D"CBM/MSDOS.ML":BANK15:B
      LOAD"CBM/MSDOS.ML"
ES 40 PRINT"{CLR}";TAB(14+A);C
      HR$(14);"MS READ 128":PR
      INTTAB(12+A);"COPYRIGHT
      {SPACE}1987":PRINTTAB(6+
      A);"COMPUTE! PUBLICATION
      S, INC."
DF 45 PRINTTAB(10+A);"ALL RIGH
      TS RESERVED."
HB 50 IFPEEK(3612)<>32ORPEEK(3
      613)<>33THENGOSUB2540
QG 60 DIMNF$(113),EF$(113),BF(
      113),AF(113),CF(113),MF(
      300),MS(100),DR$(20),CD$
      (20),EN(113):PRINTCHR$(1
      4):BB=17:GOTO80
GE 70 GOSUB1700:GOSUB500:GOSUB
      820:GOSUB1330:GOSUB1370:
      END
EA 80 FORJ=1TOBB:READCD$(J):PR
      INT"{RVS}"CD$(J)"{OFF} "
      ;:NEXT:PRINT
MM 90 GOSUB500
GD 100 PRINT#15,"UJ"
CE 110 PRINTBL$;CHR$(141);"
       {UP}";
CK 120 C$="":CD=0:PRINTDT$(DT)
       ;:INPUTC$:GOSUB410
BE 130 FORJ=1TOBB:IFLEFT$(C1$,
       2)=LEFT$(CD$(J),2)THENC
       D=J:J=BB+1
MM 140 NEXT
AC 150 ONCD GOTO 180,200,220,2
       10,190,230,270,330,250,
       260,290,320,350,170,360
       ,370,380
ED 160 REM{4 SPACES}GOTO21
CQ 170 PRINT"{2 UP}";CHR$(141)
       ;:GOTO120
HA 180 GOSUB500:IFDT=2THENGOSU
       B550:GOTO100:ELSE GOTO1
       00
ER 190 END :GOTO110
PC 200 GOSUB390:GOSUB500:IFDT=
       1THENCATALOG(C2$):GOTO1
       10:ELSE GOSUB1700:GOSUB
       550:GOSUB1330:GOSUB1370
       :GOSUB1480:GOTO100
JS 210 PRINT"{UP}";:FORJ=1TOBB
       :PRINT"{RVS}"CD$(J)"
       {OFF} ";:NEXT:PRINT:GOT
       O110
JM 220 IFSZ<>0THENGOSUB1900:PR
       INT:GOTO110:ELSE GOTO11
       0
AH 230 GOSUB500:IFDT=1THENGOSU
       B1850:PRINT:GOTO110:ELS
       E GOSUB550:IFASC(MD$)<>
       252THEN100:ELSE GOSUB17
       00:GOSUB1330:GOSUB1370:
       GOSUB2200
JH 240 IFNO<>0THENGOSUB1740:PR
       INT:GOTO100:ELSEPRINTC2
       $;" {RVS}NOT FOUND{OFF}
       "BL$:GOTO100
EF 250 GOSUB500:IFDT=2THENGOSU
       B550:GOSUB820:GOTO100:E
       LSE CATALOG"#####":GOTO
       100
RQ 260 IFDT=2THENGOSUB1370:GOS
       UB1480:GOTO110:ELSE
       {2 SPACES}GOSUB390:CATA
       LOG (C2$):GOTO100
FP 270 GOSUB500:IFDT=1THENGOSU
       B1880:GOTO110:ELSE GOSU
       B550:IFASC(MD$)<>252THE
       N100:ELSE GOSUB1700:GOS
       UB1330:GOSUB1370:GOSUB2
       200
KC 280 IFNO=0THENGOSUB300:PRIN
       T:GOTO100:ELSE PRINTBL$
       ;CHR$(141)+"{UP}";C2$;"
        {RVS}EXISTS{OFF}":GOTO
       100
EP 290 IFSZ<>0THENGOSUB1890:PR
       INT:GOTO110:ELSE GOTO11
       0
PK 300 PRINT".";:GOSUB1970:PRI
       NT".";:GOSUB2050:PRINT"
       .";:GOSUB2110:PRINT".";
       :GOSUB2140:PRINT".";:GO
       SUB2280:PRINT".";:GOSUB
       2340:GOTO100
JS 310 STOP:GOTO100
DH 320 PRINT"{RVS}";DS$:GOTO11
       0:
GH 330 H1=CC:H2=SZ:GOSUB500:IF
       DT=1THENGOSUB1060:CC=H1
       :SZ=H2:GOTO110:ELSEGOSU
       B550:IFASC(MD$)<>252THE
       N100:ELSEGOSUB1700:GOSU
       B1330:GOSUB1370:GOSUB22
       00
XD 340 IFNO<>0THENGOSUB1060:PR
       INT:CC=H1:SZ=H2:GOTO100
       :ELSE PRINTBL$"{UP}";C2
       $;" {RVS}NOT FOUND{OFF}
       ":CC=H1:SZ=H2:GOTO100
HM 350 GOSUB 1190:GOTO100
FM 360 GOSUB1160:GOTO110
XS 370 GOSUB2390:GOTO110
RX 380 GOSUB1280:GOTO110
QR 390 E=0:F=0:G=0:H=0:IFC2$=
       ""THENC2$="*":RETURN: EL
       SE RETURN
SA 400 PRINT"{2 UP}";CHR$(141)
       ;:GOTO110
HJ 410 E=INSTR(C$," "):IFE=0TH
       ENE=LEN(C$)+1:GOTO450
HJ 420 F=INSTR(C$," ",E+1):IFF
       =0THENF=LEN(C$)+1:GOTO4
       50
AC 430 G=INSTR(C$," ",F+1):IFG
       =0THENG=LEN(C$)+1:GOTO4
       50
CE 440 H=INSTR(C$," ",G+1):IFH
       =0THENH=LEN(C$)+1:GOTO4
       50
BJ 450 C1$="":C2$="":C3$="":C4
       $="":IFE<>0THENC1$=LEFT
       $(C$,E-1)
CM 460 IFF<>0THENC2$=MID$(C$,E
       +1,F-E-1)
RC 470 IFG<>0THENC3$=MID$(C$,F
       +1,G-F-1)
HK 480 IFH<>0THENC4$=MID$(C$,G
       +1,H-G-1)
FF 490 RETURN
FB 500 REM ========{2 SPACES}R
       EAD DISK ID INFO ======
       ==
GB 510 PRINTBL$+CHR$(141)"{UP}
       {RVS}READING DISK ID
       {OFF}....{UP}";CHR$(141
       );
PM 520 CLOSE15:OPEN15,8,15,"I"
       :PRINT#15,"U0"+CHR$(10)
FP 530 PRINT#15,"M-R"+CHR$(94)
       +CHR$(0)+CHR$(1):GET#15
       ,A$:IFASC(A$)<128THENDT
       $(1)="{RVS}CBM DISK
       {OFF}-->":DT=1:RETURN
SM 540 DT$(2)="{RVS}MSDOS DISK
       {OFF}-->":DT=2:RETURN
CK 550 REM ==== READ MSDOS DIS
       K CONTROL ELEMENTS
       {2 SPACES}======
HR 560 REM
GH 570 PRINT#15,"U0"+CHR$(192)
       +CHR$(0)+CHR$(1)+CHR$(1
       )
FQ 580 PRINT#15,"M-R"+CHR$(0)+
       CHR$(3)+CHR$(0)
BH 590 JM$="":FORJ=1TO3:GET#15
       ,A$:JM$=JM$+A$:NEXT
QH 600 EM$="":FORJ=1TO8:GET#15
       ,A$:EM$=EM$+A$:NEXT
AR 610 GOSUB780:BS=A
FG 620 GET#15,A$:SC=ASC(A$)
KX 630 GOSUB780:RS=A
EG 640 GET#15,A$:NF=ASC(A$)
FB 650 GOSUB780:RE=A
BB 660 GOSUB780:SV=A
AF 670 GET#15,A$:MD$=A$
KD 680 IFASC(MD$)<>252THENPRIN
       T"{RVS}*** W A R N I N
       {SPACE}G **{2 SPACES}M
        S{SHIFT-SPACE}D
       {SHIFT-SPACE}O S
       {2 SPACES}F O
       {SHIFT-SPACE}R M
       {SHIFT-SPACE}A T
       {2 SPACES}N O T
       {2 SPACES}C O M P A T I
        B L E{OFF}"
SF 690 GOSUB780:SF=A
BB 700 GOSUB780:TS=A
SH 710 GOSUB780:NH=A
GK 720 GOSUB780:HS=A
PR 730 TC=SV/SC
KA 740 F1%(1)=RS:M1(1)=M0+F1%(
       1)*BS:FORJ=2TONF:F1%(J)
       =F1%(J-1)+SF:M1(J)=M0+F
       1%(J)*BS:NEXT
BB 750 RD=F1%(NF)+SF:M2=M0+RD*
       BS
QK 760 FA=RD+(RE*32)/BS:M3=M0+
       FA*BS:RETURN
RR 770 REM == PART OF ABOVE ==
       =======
CH 780 GET#15,A$:GET#15,B$:A=A
       SC(A$)+ASC(B$)*256:RETU
       RN
AC 790 PRINT#15,"U0"+CHR$(192+
       SD*16)+CHR$(TK)+CHR$(SE
       )+CHR$(1):RETURN
BJ 800 PRINT#15,"M-R"+CHR$(0)+
       CHR$(2+BU)+CHR$(0):RETU
       RN
HD 810 REM === PRINT MSDOS FOR
       MAT INFO =====
ER 820 PRINT"{5 SPACES}{RVS}DI
       SK INFORMATION"
CS 830 PRINT"COMPANY NAME{RVS}
       ",EM$
KP 840 PRINT"BYTE PER SECTOR
       {RVS}",BS
AD 850 PRINT"SECTORS PER CLUST
       ER{RVS}",SC
GC 860 PRINT"RESERVED SECTORS
       {RVS}",RS
FR 870 PRINT"NUMBER OF FATS
       {RVS}",NF
AS 880 PRINT"ROOT DIR ENTRIES
       {RVS}",RE
DB 890 PRINT"SECTORS IN VOLUME
       {RVS}",SV
FQ 900 PRINT"MEDIA DESCRIPTOR
       {RVS}",ASC(MD$)
BQ 910 PRINT"SECTORS PER FATS
       {RVS}",SF
EX 920 PRINT"SECTORS PER TRACK
       {RVS}",TS
AJ 930 PRINT"NUMBER OF SIDES
       {RVS}",NH
```

```
JC 940  PRINT"HIDDEN SECTORS
        {RVS}",HS
XB 950  PRINT"CLUSTERS IN VOL
        {RVS}",TC
KD 960  PRINT"FAT LOG SECTORS
        {RVS}",:FORJ=1TONF:PRIN
        TF1%(J);:NEXT:PRINT
EP 970  PRINT"ROOT DIR LOG SECT
        OR{RVS}",RD
BG 980  PRINT"FILE AREA LOG SEC
        {RVS}",FA
CE 990  RETURN
CD 1000 REM ===CONVERT CLUSTER
        /LOG SEC TO HEAD/TRACK
        /SEC
MQ 1010 LS=(CN-2)*SC+FA
FX 1020 IFNH=1THEN1040
RJ 1030 IFLS+1>SV/2THENSD=1:TK
        =INT((LS-SV/2)/TS):SE=
        INT(1.1+((LS-SV/2)/TS-
        TK)*TS):GOTO1050
GE 1040 SD=0:TK=INT(LS/TS):SE=
        INT(1.1+(LS/TS-TK)*TS)
AD 1050 RETURN
EK 1060 REM =====DELETE CBM ==
        ===
BH 1070 IFC$=""THENPRINT:RETUR
        N:ELSE PRINTBL$+CHR$(1
        41)"{UP}{RVS}DELETE
        {OFF} ";C2$" {RVS}? (Y
        /N){OFF}":GETKEYA$:IFA
        $<>"Y"THENPRINT:RETURN
PM 1080 IFDT=2THEN1090:ELSE SC
        RATCH(C2$):PRINT"{RVS}
        "DS$:PRINT:RETURN
BF 1090 REM ===={2 SPACES}DELE
        TE MSDOS ====
HJ 1100 PRINT"{UP}"+BL$+CHR$(1
        41)"{UP}{RVS}DELETING
        {OFF} ";C2$" {OFF}..."
        ;
KP 1110 GOSUB2200:GOSUB1550:BA
        NK1:POKE(M2+EN(NO)*32)
        ,229:BANK15
SF 1120 FORJ=1TOCC:FP=INT(MF(J
        )*1.5):H=0:L=0
JA 1130 BANK1:POKE(M1(1)+FP+1)
        ,H:POKE(M1(1)+FP+0),L
PB 1140 NEXT:BANK15
AJ 1150 GOSUB2300:PRINT"{UP}";
        BL$;CHR$(141);C2$;"
        {RVS}DELETED{OFF}":GOT
        O100
PP 1160 PRINTBL$+CHR$(141)+"
        {UP}{RVS}INSERT DISK T
        O BE FORMATTED FOR CBM
         AND PRESS 'C' {OFF}
        {UP}":GETKEYA$:IFA$<>"
        C"THENPRINT:RETURN:ELS
        E PRINT#15,"N0:"+C2$+"
        ,"+C3$
FB 1170 PRINTBL$+CHR$(141)"
        {UP}{RVS}FORMATTING FO
        R CBM{OFF}..."
RB 1180 PRINT"{RVS}";DS$:RETUR
        N
JX 1190 REM ===== FORMAT MSDOS
        ======
HJ 1200 PRINTBL$+CHR$(141)"
        {RVS}INSERT PRE-FORMAT
        TED MS-DOS DISK
        {2 SPACES}AND PRESS 'C
        ' {OFF}":GETKEYA$:IFA$
        <>"C"THENPRINT:RETURN:
        ELSE GOSUB500
RP 1210 IFDT<>2THENPRINT"{RVS}
        NOT MS-DOS":PRINT:RETU
        RN:ELSE GOSUB1700:GOSU
        B550:GOSUB1330:GOSUB13
        70
CF 1220 PRINTBL$+CHR$(141)"
        {RVS}INSERT TARGET DIS
        K TO BE FORMATTED AND
        {SPACE}PRESS 'C' {OFF}
        ":A$="":GETKEYA$:IFA$<
        >"C"THENPRINT:RETURN
BR 1230 IFC2$<>""THENFORJ=1TO1
        1:POKEM2+J-1,ASC(MID$(
        C2$+"{10 SPACES}",J,1)
        ):NEXT:POKEM2+11,40
BG 1240 PRINT#15,"U0"+CHR$(4):
        PRINT"{RVS}";DS$
EF 1250 PRINT#15,"U0"+CHR$(6)+
        CHR$(129)+CHR$(1)+CHR$
        (2)+CHR$(39)+CHR$(9)+C
        HR$(0)+CHR$(0)+CHR$(22
        9):PRINT"{RVS}";DS$:GO
        SUB1300
AR 1260 GOSUB2280:PRINT:RETURN
FQ 1270 REM=== SCRATCH DSK ===
QD 1280 PRINTBL$+CHR$(141)"
        {RVS}INSERT MS-DOS DIS
        K TO BE SCRATCHED AND
        {SPACE}PRESS 'C' {OFF}
        ":GETKEYA$:IFA$<>"C"TH
        ENPRINT:RETURN:ELSE GO
        SUB500
GQ 1290 IFDT<>2THENPRINT"{RVS}
        NOT MS-DOS DSK ":PRINT
        :RETURN:ELSE GOSUB550:
        IFASC(MD$)<>252THENRET
        URN:ELSE GOSUB1700:GOS
        UB1330:GOSUB1370:GOSUB
        1300:GOSUB2280:RETURN
AJ 1300 BANK1:PRINT"{RVS}WAIT
        {OFF}...";BL$:FORJ=0TO
        111:POKEM2+J*32,0:NEXT
        :FORJ=3TO524:POKEM1(1)
        +J,0:POKEM1(2)+J,0:NEX
        T:BANK15
GC 1310 IFC2$<>""THENBANK1:FOR
        J=1TO10:POKEM2+J-1,ASC
        (MID$(C2$+"{10 SPACES}
        ",J,1)):POKEM2+11,40:N
        EXT:BANK15:RETURN:ELSE
         RETURN
EQ 1320 REM=== READ MSDOS DISK
         ID AND ROOT DIR INTO
        {SPACE}MEMORY =====
BM 1330 PRINTBL$+CHR$(141)"
        {UP}{RVS}READING ROOT/
        FAT/DIR SECTORS{OFF}..
        ..{UP}";CHR$(141);
FA 1340 PRINT#15,"U0"+CHR$(8)+
        CHR$(1):BANK15:A=M0:LS
        =0:GOSUB1020:PRINT#15,
        "U0"+CHR$(64+SD*16)+CH
        R$(TK)+CHR$(SE)+CHR$(F
        A)
HQ 1350 HI=INT(A/256):LO=(A/25
        6-HI)*256:POKE250,LO:P
        OKE251,HI:BANK15:SYS31
        28,127,FA:BANK15
GH 1360 RETURN
SA 1370 REM === COMPUTE MSDOS
        {SPACE}DIR FROM MEMORY
        =====
BJ 1380 NF$(1)="":DR$(1)="":EF
        $(1)="":BF(1)=0:AF(1)=
        0:LB$="":PRINTBL$+CHR$
        (141)"{UP}{RVS}COMPUTI
        NG DIRECTORY{OFF}....
        {UP}";CHR$(141);
MM 1390 BANK1:TF=0:B=0:FORJ=0T
        O111:P=J*32:IFPEEK(M2+
        P)=229THEN1470
PD 1400 IFPEEK(M2+P)=0ORPEEK(M
        2+P)>90THENJ=112+1:GOT
        O1470
RF 1410 IFPEEK(M2+11+P)=40THEN
        LB$="":FORI=0TO10:LB$=
        LB$+CHR$(PEEK(M2+I+P))
        :NEXT:GOTO1470
PM 1420 IFPEEK(M2+11+P)=16THEN
        B=B+1:DN=B:DR$(B)="":F
        ORI=0TO7:DR$(B)=DR$(B)
        +CHR$(PEEK(M2+I+P)):NE
        XT:GOTO1470
EX 1430 TF=TF+1:NF$(TF)="":FOR
        I=0TO7:NF$(TF)=NF$(TF)
        +CHR$(PEEK(M2+I+P)):NE
        XT
AH 1440 EF$(TF)="":FORI=0TO2:E
        F$(TF)=EF$(TF)+CHR$(PE
        EK(M2+8+I+P)):NEXT
FQ 1450 BF(TF)=PEEK(M2+28+P)+P
        EEK(M2+29+P)*256+(PEEK
        (M2+30+P)+PEEK(M2+31+P
        )*256)*65536
PQ 1460 AF(TF)=PEEK(M2+11+P):C
        F(TF)=PEEK(M2+26+P)+PE
        EK(M2+27+P)*256:EN(TF)
        =J
FE 1470 NEXT:RETURN
SF 1480 PRINTBL$;CHR$(141)"
        {UP}{RVS}VOLUMN{OFF} /
        ";LB$:REM === LIST DIR
        ECTORY =====
SF 1490 IFNF$(1)=""THENRETURN:
        ELSE PRINT"{RVS}NAME
        {5 SPACES}EXT
        {2 SPACES}BYTES
        {2 SPACES}";"
        {3 SPACES}ATT";"
        {3 SPACES}LOC"
QQ 1500 IFDN<>0THENFORJ=1TODN:
        PRINTDR$(J);"
        {12 SPACES}<DIR>":NEXT
        :BANK15
CD 1510 FORJ=1TOTF
DP 1520 PRINTNF$(J);".";EF$(J)
        ;"{2 SPACES}"LEFT$(STR
        $(BF(J))+"{5 SPACES}",
        6);"{3 SPACES}"AF(J);"
        {3 SPACES}";CF(J)
BP 1530 NEXT:BANK15:RETURN
GM 1540 REM === FIND CLUSTERS
        {SPACE}IN FILE ====
DD 1550 A=1:MF(A)=CF(NO)
EQ 1560 BANK1:FP=INT(1.5*MF(A)
        )
JH 1570 IF1.5*MF(A)<>INT(1.5*M
        F(A))THEN1590
BF 1580 BANK1:NP=256*(15AND(PE
        EK(M1(1)+FP+1)))+PEEK(
        M1(1)+FP+0):GOTO1600
RD 1590 BANK1:NP=16*PEEK(M1(1)
        +FP+1)+PEEK(M1(1)+FP+0
        )/16:NP=INT(NP)
DB 1600 IFNP>=4088ORNP>TCTHEN1
        620
AF 1610 A=A+1:MF(A)=NP:GOTO156
        0
HM 1620 CC=A:BANK15
KH 1630 FORJ=1TOA:NEXT:RETURN
AB 1640 REM ========= PRINT BY
        TES FM DISK ======
CJ 1650 BANK1:POKE2979,76:POKE
        2980,01:POKE2981,12:PO
        KE3002,188
MG 1660 BANK15:POKE2979,76:POK
        E2980,01:POKE2981,12:P
        OKE3002,188:RETURN
RC 1670 REM ========= BYTES FM
         DISK TO MEMORY CBM
        {2 SPACES}DSK ====
XA 1680 BANK1:POKE2979,76:POKE
        2980,197:POKE2981,11:P
        OKE3002,158
JB 1690 BANK15:POKE2979,76:POK
        E2980,197:POKE2981,11:
        POKE3002,158:RETURN
PA 1700 REM ========= BYTES FM
```

```
              DISK TO MEMORY MSDOS
              {SPACE}DSK ====
XC 1710 BANK1:POKE2979,142:POK
        E2980,0:POKE2981,255:P
        OKE3002,158
DB 1720 BANK15:POKE2979,142:PO
        KE2980,0:POKE2981,255:
        POKE3002,158:RETURN
MD 1730 REM=== LOAD MSDOS FILE
        ===========
AH 1740 SZ=BF(NO):POKE251,TB:P
        OKE250,0:POKE2996,0:GO
        SUB1640:GOSUB1550:POKE
        3002,158:IFCC=1THEN177
        0
SM 1750 FORJ=1TOCC-1:CN=MF(J):
        GOSUB1010:Q=SE:FORSE=Q
        TOQ+SC-1:GOSUB790:FORB
        U=1TOBS/256
QH 1760 GOSUB800:BANK15:SYS296
        4:NEXT:NEXT:NEXT
XS 1770 RM=BF(NO)-(CC-1)*SC*BS
        :RN=INT(RM/256):RO=RM-
        RN*256:CN=MF(CC):GOSUB
        1010:POKE3002,188
ME 1780 GOSUB790
EM 1790 IFRM<=256THENBANK1:POK
        E2996,RO:BANK15:POKE29
        96,RO
CH 1800 IFRM>0THENBU=1:GOSUB80
        0:BANK15:SYS2964:RM=RM
        -256:IFRM<256THENBANK1
        :POKE2996,RO:BANK15:PO
        KE2996,RO
AP 1810 IFRM>0THENBU=2:GOSUB80
        0:BANK15:SYS2964:RM=RM
        -256
CD 1820 IFRM>0THENSE=SE+1:GOTO
        1780
XP 1830 GOSUB1700:BANK1:POKE29
        96,0:BANK15:POKE2996,0
        :PRINT:RETURN
KS 1840 REM ==== LOAD CBM FILE
        =====
HS 1850 CLOSE1:OPEN1,8,0,MID$(
        C$,E+1)+",S,R":IFDS<>0
        THENPRINT"{RVS}"DS$"
        {UP}":RETURN:ELSE BANK
        1:POKE2996,0:BANK15:PO
        KE2996,0:GOSUB1670
GA 1860 BANK1:POKE251,TB:POKE2
        50,0:BANK15:POKE251,TB
        :POKE250,0:SYS2968,,1:
        CLOSE1
HF 1870 SZ=PEEK(250)+PEEK(251)
        *256-TB*256:POKE250,0:
        RETURN
BR 1880 IFC$=""THENRETURN:ELSE
         CLOSE1:OPEN1,8,1,MID$
        (C$,E+1)+",S,W":PRINT
        "";BL$+CHR$(141)"{RVS}S
        AVING{OFF} ";MID$(C$,E
        +1):IFDS<>0THENPRINT"R
        "DS$"":RETURN:ELSEGOTO
        1920
XQ 1890 CLOSE1:OPEN1,4,7:GOTO1
        920:REM ==PRN==
MG 1900 CLOSE1:OPEN1,3,1:GOTO1
        920:REM ==SCN==
KR 1910 REM === LIST BUFFER ==
        ======
CQ 1920 REM
BC 1930 IFVAL(C3$)=0THENC3$="1
        00"
KJ 1940 IFVAL(C3$)<VAL(C2$)THE
        NC3$=STR$(VAL(C2$)+1)
FD 1950 BZ=(TB*256)+INT(VAL(C2
        $)*.01*SZ):EZ=(TB*256)
        +INT(VAL(C3$)*.01*SZ):
        H=INT(BZ/256):L=(BZ/25
        6-H)*256:POKE250,L:POK
        E251,H
XQ 1960 H=INT(EZ/256):L=(EZ/25
        6-H)*256:BANK15:SYS 31
        25,L,H:CLOSE1:RETURN
QR 1970 REM===={2 SPACES}FIND
        {SPACE}AVAIL CLUSTERS
        {SPACE}======
KF 1980 PRINTBL$+CHR$(141)"
        {UP}{RVS}SEARCHING CLU
        STERS{OFF}....{UP}";CH
        R$(141);
SX 1990 A=0:TU=2
MH 2000 FP=INT(TU*1.5):IFFP<>T
        U*1.5THEN2020
GE 2010 BANK1:NP=256*(15AND(PE
        EK(M1(1)+FP+1)))+PEEK(
        M1(1)+FP+0):GOTO2030
PM 2020 BANK1:NP=16*PEEK(M1(1)
        +FP+1)+PEEK(M1(1)+FP+0
        )/16:NP=INT(NP)
XM 2030 IFNP=0THENA=A+1:MS(A)=
        TU:IFA>SZ/(SC*BS)THENC
        C=A:RETURN
HC 2040 TU=TU+1:GOTO2000
SX 2050 REM== CAL FAT ENTRIES
        {SPACE}AND MODIFY FAT
        {SPACE}IN MEMORY ====
AM 2060 FORJ=1TOCC-1:FP=INT(MS
        (J)*1.5):H=INT(MS(J+1)
        /256)AND16:L=(MS(J+1)/
        256-H)*256
CX 2070 IFFP<>MS(J)*1.5THEN:H=
        INT(16*MS(J+1)/256):L=
        (16*MS(J+1)/256-H)*256
XQ 2080 BANK1:POKE(M1(1)+FP+1)
        ,H:POKE(M1(1)+FP+0),L
QG 2090 NEXT:FP=INT(MS(CC)*1.5
        ):POKE(M1(1)+FP+1),255
        :POKE(M1(1)+FP+0),PEEK
        (M1(1)+FP+0)OR240
MC 2100 RETURN
XD 2110 REM ==== FIND AVAIL DI
        R ENTRY =====
MM 2120 BANK1:TF=0:FORJ=0TO111
        *32{2 SPACES}STEP32:IF
        PEEK(M2+J)=229ORPEEK(M
        2+J)=0THENAE=J+M2:J=11
        1*32
BA 2130 NEXT:BANK15:RETURN
EF 2140 REM === ADD DIR ENTRY
        {SPACE}===
PC 2150 BANK1:FORJ=0TO31:POKEA
        E+J,0:NEXT
AC 2160 A$=NA$+"{8 SPACES}":FO
        RJ=1TO8:POKEAE+J-1,ASC
        (MID$(A$,J,1)):NEXT:H=
        INT(MS(1)/256):L=(MS(1
        )/256-H)*256:POKEAE+26
        ,L:POKEAE+27,H:POKEAE+
        11,32
GB 2170 A$=EX$+"{3 SPACES}":FO
        RJ=1TO3:POKEAE+8+J-1,A
        SC(MID$(A$,J,1)):NEXT:
        H=INT(MS(1)/256):L=(MS
        (1)/256-H)*256:POKEAE+
        26,L:POKEAE+27,H:POKEA
        E+11,32
RH 2180 H=INT(SZ/256):L=(SZ/25
        6-H)*256:POKEAE+28,L:P
        OKEAE+29,H:POKEAE+11,3
        2:BANK15:RETURN
KH 2190 NA$="ADD":EX$="TES":SZ
        =5674:GOSUB1970:GOSUB2
        050:GOSUB2110:GOSUB214
        0:GOSUB1370:END
EM 2200 REM===== SEARCH MSDOS
        {SPACE}DIR ======
QK 2210 NA$="":EX$="":K=INSTR(
        C2$,"."):IFK=0THENNA$=
        C2$:GOTO2220:ELSE NA$=
        LEFT$(C2$,K-1):EX$=MID
        $(C2$,K+1,LEN(C2$))
AA 2220 NA$=LEFT$(NA$+"
        {8 SPACES}",8):EX$=LEF
        T$(EX$+"{3 SPACES}",3)
XA 2230 NO=0:IFEX$="{3 SPACES}
        "THEN2250:ELSE
        {2 SPACES}FORJ=1TOTF:I
        F(NA$=NF$(J))ANDEX$=EF
        $(J)ANDNF$(J)<>"
        {8 SPACES}"THENNO=J:J=
        TF+1
XH 2240 NEXT:GOTO2270
XE 2250 FORJ=1TOTF:IF(NA$=NF$(
        J))ANDNF$(J)<>""THENNO
        =J:J=TF+1
MS 2260 NEXT
PS 2270 RETURN
XF 2280 REM ==== SAVE FAT AND
        {SPACE}DIR TO DISK ===
        =
HF 2290 IFC$=""THENRETURN:ELSE
         PRINTBL$+CHR$(141)"
        {UP}{RVS}SAVING{OFF} "
        ;C2$:IFDS<>0THENPRINT"
        {RVS}"DS$"{OFF}":REM
        {4 SPACES}==SAV==
QS 2300 BANK15
KJ 2310 OS=M0:LS=0:GOSUB1020
FP 2320 PRINT#15,"U0"+CHR$(8)+
        CHR$(1):PRINT#15,"U0"+
        CHR$(66+SD*16)+CHR$(TK
        )+CHR$(SE)+CHR$(FA):PR
        INT"+";
AD 2330 H=INT(OS/256):L=(OS/25
        6-H)*256:POKE250,L:POK
        E251,H:SYS3122,127,FA:
        PRINT"{K}";:BANK15:RET
        URN
CK 2340 REM ==== SAVE FILE CLU
        STER TO DISK ====
MQ 2350 BANK15:OS=TB*256:FORU=
        1TOCC:CN=MS(U):GOSUB10
        10
FS 2360 PRINT#15,"U0"+CHR$(66+
        SD*16)+CHR$(TK)+CHR$(S
        E)+CHR$(1):PRINT"#";
FG 2370 H=INT(OS/256):L=(OS/25
        6-H)*256:POKE250,L:POK
        E251,H:SYS3122,63,1:OS
        =OS+512:NEXT:BANK15::P
        RINT"{UP}";CHR$(141);:
        RETURN
RX 2380 REM === DISKCOPY ===
AB 2390 PRINTBL$+CHR$(141)"
        {RVS}INSERT{OFF} ORIGI
        NAL {RVS}MS-DOS DISK A
        ND PRESS 'C' {OFF}":A$
        ="":GETKEYA$:IFA$<>"C"
        THENPRINT:RETURN
AJ 2400 A$="":GOSUB500:IFDT<>2
        THENPRINT"{RVS}NOT MS-
        DOS":PRINT:RETURN:ELSE
         GOSUB550:IFASC(MD$)<>
        252THENRETURN:GOSUB170
        0:GOSUB1330:GOSUB1370:
        PRINTBL$;"{UP}"
GF 2410 A=M0:SE=1:Q1=9:QB=127:
        GOSUB2530:FORTK=0TO10:
        GOSUB2490:PRINT"R";:NE
        XT:A=M3:QB=63:GOSUB253
        0:FORTK=11TO19:GOSUB24
        90:PRINT"R";:NEXT
RJ 2420 PRINTBL$+CHR$(141)"
        {RVS}INSERT TARGET DIS
        K AND PRESS 'C' {OFF}"
        :GETKEYA$:PRINT#15,"U0
        "+CHR$(4):IFA$<>"C"THE
        NPRINT:RETURN:ELSE PRI
        NT"{RVS}FORMAT TARGET
        {SPACE}DISK? (Y/N)"
JE 2430 GETKEYA$:IFA$="N"THEN2
```

```
         440:ELSE{2 SPACES}PRIN
         T#15,"U0"+CHR$(6)+CHR$
         (129)+CHR$(0)+CHR$(2)+
         CHR$(39)+CHR$(9)+CHR$(
         0)+CHR$(0)+CHR$(229):P
         RINT"{RVS}";DS$
QH 2440  A=M0:SE=1:Q1=9:QB=127:
         GOSUB2530:FORTK=0TO10:
         GOSUB2510:PRINT"W";:NE
         XT:A=M3:QB=63:GOSUB253
         0:FORTK=11TO19:GOSUB25
         10:PRINT"W";:NEXT
GG 2450  PRINTBL$+CHR$(141)"
         {RVS}INSERT{OFF} ORIGI
         NAL {RVS}MS-DOS DISK A
         ND PRESS 'C' {OFF}":GE
         TKEYA$:PRINT#15,"U0"+C
         HR$(4):IFA$<>"C"THENPR
         INT:RETURN
AR 2460  A=M0:SE=1:Q1=9:QB=127:
         GOSUB2530:FORTK=20TO30
         :GOSUB2490:PRINT"R";:N
         EXT:A=M3:QB=63:GOSUB25
         30:FORTK=31TO39:GOSUB2
         490:PRINT"R";:NEXT
AG 2470  PRINTBL$+CHR$(141)"
         {RVS}INSERT TARGET MS-
         DOS DISK AND PRESS 'C'
          {OFF}":GETKEYA$:PRINT
         #15,"U0"+CHR$(4):IFA$<
         >"C"THENPRINT:RETURN
XG 2480  A=M0:SE=1:Q1=9:QB=127:
         GOSUB2530:FORTK=20TO30
         :GOSUB2510:PRINT"W";:N
         EXT:A=M3:QB=63:GOSUB25
         30:FORTK=31TO39:GOSUB2
         510:PRINT"W";:NEXT :PR
         INT"{RVS}COMPLETE":RET
         URN
RG 2490  PRINT#15,"U0"+CHR$(8)+
         CHR$(1):PRINT#15,"U0"+
         CHR$(64+SD*16)+CHR$(TK
         )+CHR$(SE)+CHR$(Q1)
RP 2500  BANK15:SYS3128,QB,Q1:B
         ANK15:RETURN
SR 2510  PRINT#15,"U0"+CHR$(8)+
         CHR$(1):PRINT#15,"U0"+
         CHR$(66+SD*16)+CHR$(TK
         )+CHR$(SE)+CHR$(Q1)
DR 2520  BANK15:SYS3122,QB,Q1:B
         ANK15:RETURN
HM 2530  H=INT(A/256):L=(A/256-
         H)*256:POKE250,L:POKE2
         51,H:RETURN
AC 2540  PRINT"{RVS}CREATING TR
         ANSLATE TABLES{OFF}...
         .{2 UP}";CHR$(141);
FJ 2550  BANK1:GOSUB2560:BANK15
         :GOSUB2560:RETURN
AS 2560  MM=3580:OO=MM+256:FORJ
         =1TO256:POKEMM+J,0:POK
         EOO+J,0:NEXT
AH 2570  FORJ=32TO64:POKEMM+J,J
         :NEXT
EH 2580  FORJ=65TO90:POKEMM+J,J
         +32:NEXT:FORJ=91TO95:P
         OKEMM+J,J:NEXT
PB 2590  FORJ=193TO218:POKEMM+J
         ,J-128:NEXT
PM 2600  POKEMM+13,13:POKEMM+20
         ,8:POKEMM+146,16:POKEM
         M+133,16
FH 2610  FORJ=0TO255:K=PEEK(MM+
         J)
SH 2620  IFK<>0THENPOKEOO+K,J:P
         OKEOO+K+128,J
AQ 2630  NEXT:RETURN
MA 2640  REM===================
         =========
CJ 2650  DATA REGDSK,DIR,LIST,H
         ELP,QUIT,LOAD,SAVE,DEL
         ETE,DSPEC,LDIR,PLIST,E
         RRDSK,MFORMAT,"{OFF}
         {4 SPACES}{RVS}",CFORM
         AT,DCOPY,SCRDSK
HP 2660  REM== END OF PROGRAM =
         =========
```

Program 2: CBM/MSDOS.ML

Please refer to the "MLX" article elsewhere in this issue before entering this program.

```
0B90:85 FA 86 FB A2 0F EA EA CA
0B98:EA 20 C6 FF A0 00 20 CF 1A
0BA0:FF A2 7F 8E 00 FF 91 FA 56
0BA8:A2 00 8E 00 FF 20 B7 FF D1
0BB0:D0 0A C8 C0 00 D0 E7 E6 D0
0BB8:FB 4C 9E 0B 84 FA 20 CC 81
0BC0:FF 60 EA EA 60 AA BD FC 21
0BC8:0D 91 FA A9 7F 8D 00 FF F5
0BD0:8A 91 FA A2 00 8E 00 FF 54
0BD8:20 D2 FF 4C A8 0B EA EA AA
0BE0:EA 8D 08 DD 60 EA EA EA 1E
0BE8:EA EA EA EA EA EA EA EA FE
0BF0:EA EA EA EA EA EA EA EA 07
0BF8:EA EA EA EA EA EA EA EA 0F
0C00:EA EA EA EA 8D FD 0F 8E 65
0C08:FE 0F 8C FF 0F AA BD FC 90
0C10:0E 20 D2 FF A9 7F 8D 00 F7
0C18:FF BD FC 0E 91 FA A9 00 EB
0C20:8D 00 FF AD FD 0F AE FE 62
0C28:0F AC FF 0F 4C A6 0B 00 F6
0C30:FF 00 4C 3B 0C 4C FE 0C 21
0C38:4C 34 0D 8D C8 0D 8D 01 94
0C40:D5 A9 00 8D 03 D5 A9 00 49
0C48:8D 00 FF 8E CE 0D AD 00 16
0C50:DD 8D CA 0D AD 0C DC 8D C9
0C58:CB 0D AD 0D DC 8D CC 0D E3
0C60:A2 00 78 38 20 47 FF A9 24
0C68:40 8D C9 0D A0 00 EA AD 96
0C70:00 DD CD 00 DD D0 F8 4D 2B
0C78:C9 0D 29 40 F0 F1 8D 01 4D
0C80:FF B1 FA 8D 03 FF 8D 0C 7C
0C88:DC AD C9 0D 49 40 8D C9 B4
0C90:0D A9 08 2C 0D DC F0 F9 15
0C98:E6 FA A5 FA C9 00 D0 CE 06
0CA0:E6 FB E8 E0 02 F0 03 4C 7C
0CA8:6C 0C 18 20 47 FF 2C 0D 9E
0CB0:DC AD 00 DD 09 10 8D 00 24
0CB8:DD A9 08 2C 0D DC F0 FB A7
0CC0:AD 00 DD 49 10 8D 00 DD 94
0CC8:AD 0C DC CE CE 0D AD CE 18
0CD0:0D C9 00 D0 8B 58 20 CC B9
0CD8:FF 20 CC FF AD CA 0D 8D D2
0CE0:00 DD AD CB 0D 8D 0C DC 76
0CE8:AD CC 0D 8D 0D DC A9 3F F3
0CF0:8D 01 D5 A9 7F 8D 02 D5 71
0CF8:A9 01 8D 03 D5 60 8D CD 21
0D00:0D 8E CE 0D A2 01 20 C9 12
0D08:FF A0 00 8D 02 FF B1 FA 91
0D10:A2 00 8E 00 FF 20 D2 FF 73
0D18:E6 FA A6 FA E0 00 D0 02 93
0D20:E6 FB A6 FB EC CE 0D D0 CE
0D28:E2 A6 FA EC CD 0D D0 DB AB
0D30:20 CC FF 60 8D C8 0D 8D CA
0D38:01 D5 A9 00 8D 03 D5 A9 4B
0D40:00 8D 00 FF 8D CD 0D 8E 0A
0D48:CE 0D AD 00 DD 8D CA 0D 8A
0D50:AD 0C DC 8D CB 0D AD 0D B3
0D58:DC 8D CC 0D 78 2C 0D DC 1A
0D60:AD 00 DD 49 10 8D 00 DD 36
0D68:A9 02 8D CF 0D A9 08 2C D1
0D70:0D DC F0 FB AD 00 DD 49 98
0D78:10 8D 00 DD AD 0C DC EA 1E
0D80:A9 08 2C 0D DC F0 FB AD 18
0D88:00 DD 49 10 8D 00 DD AD 1A
0D90:0C DC A0 00 8D 01 FF 91 FD
0D98:FA 8D 03 FF E6 FA A5 FA 5D
0DA0:C9 00 D0 02 E6 FB EA EE C5
0DA8:CD 0D AD CD 0D C9 00 D0 DF
0DB0:CE CE CF 0D AD CF 0D C9 41
0DB8:00 D0 C4 CE CE 0D AD CE 61
0DC0:0D C9 00 D0 A3 4C D5 0C E6
```

©

Tri-Sort For Atari

Arthur F. Horan

Here's the fastest and most versatile sort we've ever published for the Atari. A machine language version of the Shell-Metzner sorting algorithm, it will sort Atari BASIC strings on as many as three fields. For all Atari eight-bit computers.

Sorting is one of the tasks that computers do best. Whether you want to sort a mailing list or the titles of a record collection, a good, fast sort will come in handy.

Unfortunately, the easiest sorts to program are among the slowest in use. "Tri-Sort" uses one of the fastest sorting routines—the Shell-Metzner sort. Even when you have hundreds of items to sort, Tri-Sort will sort them in a split second. Best of all, Tri-Sort lets you sort up to three fields. Suppose, for example, that you have this list:

First Name	Last Name	Age
Jim	Jones	40
Sam	Smith	40
Jim	Smith	36

If you sort on the fields Last Name, First Name, and Age, then Jim Jones will appear at the start of the list, Jim Smith will appear next, and the list will end with Sam Smith. The list is sorted according to the first field. If there are any matches in this field, the order is determined by the second field. If those fields match, the third field is checked.

Typing It In

The program listed below is written in BASIC. The machine language sort program is encoded in DATA statements. Type in the program carefully and save it to tape or disk before you attempt to run it.

For a demonstration of the speed of the sorting routine, load and run the program. First, the machine language data is put into a string. This takes a few seconds. Next, 100 random records are created, each one consisting of three fields of characters. After the strings are created, you'll see the message *Press any key to see strings.* Press a key. After the strings scroll by, press a key to go to the main screen. You'll see these options:

1 Sort field 1,2,3
2 Sort field 2,3,1
3 Sort field 3,2,1
4 Quit Program
5 Create program lines for TSORT$

If you choose 1, 2, or 3, the strings that you saw previously will be sorted. The three different options specify three different selections for the primary, secondary, and tertiary fields. Option 4 allows you to escape to BASIC when you're finished trying out the program. Option 5 creates program lines designed to help you use Tri-Sort in your own programs. The dynamic keyboard technique is used to create the lines. Once created, these lines (10000–10070) can be used in your own programs. Simply LIST them to tape or disk, load your program, and then ENTER the lines.

Why not just use the DATA statements like the demo program does? Because the data uses up quite a bit of memory and takes a long time to read. DATA statements were used in the demo program because the machine language program is much easier to type in as data elements than as characters. In your own programs, add the lines 10000–10070 created with option 5, then initialize the string in the first line of your program like this:

```
1 GOSUB 10000
```

The string is dimensioned and defined within the subroutine.

Sorting Your Own Data

Tri-Sort features a simple calling mechanism. The following statement shows the syntax for calling Tri-Sort from BASIC:

```
SORT=USR(ADR(TSORT$),ADR(DAT$)
  ,RECLEN,NR-1,S1,E1,S2,E2,S3,E3)
```

TSORT$ contains the machine language Tri-Sort routine.

DAT$ holds the data that you want to sort.

RECLEN is the record length (the combined length of all the fields in each entry).

NR is the total number of records in DAT$.

S1 is the start of the first data field to be sorted. For instance, S1 would be 10 if the field you wished to sort started at the tenth character of each record, 1 if the desired field started at the first character of the record, and so on.

E1 is the end of the first data field to be sorted. For example, if the field spans the third to the eighth character of a record, S1 would be 3 and E1 would be 8. If you only wanted to sort character four, S1 would be 4 and E1 would be 4.

S2 is the start of the second data field to be sorted.

E2 is the end of the second data field to be sorted.

S3 is the start of the third data field to be sorted.

E3 is the end of the third data field to be sorted.

S2, E2, S3, and E3 are optional. You may leave them out if you don't wish to sort a secondary or tertiary field. Note that if you leave out one of the pair S2 and E2, you must leave out the other. Likewise, if you leave off either of the pair S3 and E3, you must leave off the other.

When using Tri-Sort, you must keep page 6 of memory free (memory locations 1536–1791). Tri-Sort uses this area as a temporary buffer.

Since Atari BASIC does not have genuine string arrays, Tri-Sort simulates string arrays. Here's an example:

A$="DOG . . CAT . . HORSEPIG . . BIRD."

This string can be treated as an array with five elements (five records). Note that the string above has each element padded out with periods. Normally, you would want to use spaces instead. We use periods only for clarity. Now, we can print out element N with the statement

PRINT A$(N*5−4,N*5)

where N ranges from 1 to 5.

The demo program shows more clearly how to construct the "superstring" in which the sorting will take place.

Tri-Sort

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
OD 5 REM COPYRIGHT 1988 COMPUTE! PUBLICATIONS, INC. ALL RIGHTS RESERVED.
6E 6 PRINT "{CLEAR}COPYRIGHT 1988":PRINT "COMPUTE! PUBLICATIONS, INC.":PRINT "ALL RIGHTS RESERVED."
LO 7 FOR TT=1 TO 1500:NEXT TT
HC 10 DIM TSORT$(331),MSG$(40):CLOSE #1:OPEN #1,4,0,"K:":POKE 752,1:? "{CLEAR}{2 DOWN}READing DATA, Please wait.":?
GM 20 FOR I=1 TO 331:READ A:TSORT$(I,I)=CHR$(A):NEXT I
JN 30 NR=100:RECLEN=10:DIM T$(RECLEN),DAT$(NR*10)
MO 40 PRINT "CREATING RANDOM RECORDS":?
FG 50 FOR A=1 TO NR:FOR B=1 TO RECLEN-4:T$(B,B)=CHR$(65+RND(1)*25):NEXT B
DJ 60 N=48+RND(1)*9:T$(RECLEN-3)=CHR$(N):T$(RECLEN-2)=CHR$(N):T$(RECLEN-1)=CHR$(N)
OP 70 T$(RECLEN,RECLEN)=CHR$(97+RND(1)*25)
NB 80 PRINT A;"{UP}"
KF 90 DAT$((A-1)*RECLEN+1,A*RECLEN)=T$:NEXT A
JD 100 MSG$="TO SEE STRINGS{3 TAB}(<START> PAUSES)":GOSUB 290:GOSUB 220:REM PRINT STRING
DI 110 MSG$="TO PROCEED":GOSUB 290
JO 120 ? "{CLEAR} Tri-Sort Demonstration "
NG 130 PRINT "{DOWN}1 Sort field 1,2,3":PRINT "2 Sort field 2,3,1":PRINT "3 Sort field 3,2,1":PRINT "4 Quit program"
DL 140 ? "5 Create program lines for TSORT$"
LC 150 POKE 764,255:TRAP 120:MSG$="CHOOSE ONE":GET #1,SF:? CHR$(SF):SF=SF-48:IF SF>5 OR SF<1 THEN 120
FO 160 TRAP 40000:ON SF GOSUB 330,340,350,360,770
FC 170 MSG$="TO SORT":GOSUB 290:PRINT "{DOWN}SORTING"
KN 180 SORT=USR(ADR(TSORT$),ADR(DAT$),RECLEN,NR-1,S1,E1,S2,E2,S3,E3)
ML 190 PRINT "{DOWN}DONE":PRINT "{DOWN}";:MSG$="TO SEE STRINGS{3 TAB} (<START> PAUSES)":GOSUB 290:GOSUB 220
FN 200 GOTO 110
EJ 210 REM <<< PRINT STRING >>>
PF 220 PRINT "{CLEAR}":FOR A=1 TO NR
OB 230 INDX=(A-1)*RECLEN+1:PRINT DAT$(INDX,INDX+RECLEN-5),DAT$(INDX+RECLEN-4,INDX+RECLEN-2);
IH 240 PRINT DAT$(INDX+RECLEN-1,INDX+RECLEN-1)
GC 250 IF PEEK(53279)=6 THEN GOSUB 370
BI 260 NEXT A
HJ 270 RETURN
PB 280 REM <<< WAIT FOR KEY >>>
BF 290 PRINT "PRESS ANY KEY ";
KP 300 ? MSG$:POKE 764,255
NB 310 IF PEEK(764)=255 THEN 310
HF 320 RETURN
CB 330 S1=1:E1=RECLEN-4:S2=RECLEN-3:E2=RECLEN-1:S3=RECLEN:E3=RECLEN:RETURN
CC 340 S3=1:E3=RECLEN-4:S1=RECLEN-3:E1=RECLEN-1:S2=RECLEN:E2=RECLEN:RETURN
CD 350 S3=1:E3=RECLEN-4:S2=RECLEN-3:E2=RECLEN-1:S1=RECLEN:E1=RECLEN:RETURN
ND 360 POKE 752,0:END
CA 370 IF PEEK(53279)=6 THEN 370:REM WAIT FOR RELEASE
LH 380 IF PEEK(53279)<>6 THEN 380:REM WAIT FOR ANOTHER START-PRESS
CE 390 IF PEEK(53279)=6 THEN 390:REM WAIT FOR RELEASE
HE 400 RETURN
KB 410 REM <<< READ DATA FOR M-L STRING >>>
GG 420 DATA 216,104,56,233,3,74,133,242,104,133
BL 430 DATA 213,104,133,212,104,104,133,226,104,133
MB 440 DATA 215,133,225,104,133,214,133,224,162,0
MG 450 DATA 104,104,149,227,240,2,214,227,104,104
AJ 460 DATA 149,230,232,228,242,208,239,70,225,102
BM 470 DATA 224,165,224,5,225,208,1,96,56,165
DK 480 DATA 214,229,224,133,220,165,215,229,225,133
NB 490 DATA 221,160,0,132,219,132,218,165,219,133
DL 500 DATA 217,165,218,133,216,165,224,133,233,165
MG 510 DATA 225,133,234,165,226,133,238,169,0,133
EB 520 DATA 236,133,237,162,8,70,238,144,11,24
DE 530 DATA 165,236,101,233,133,236,165,237,101,234
CI 540 DATA 106,133,237,102,236,102,235,202,208,231
DJ 550 DATA 165,235,133,222,165,236,133,223,165,212
CH 560 DATA 133,203,133,240,165,213,133,204,133,241
AE 570 DATA 24,144,2,80,148,162,0,165,203,101
CL 580 DATA 222,133,205,165,204,101,223,133,206,181
DM 590 DATA 230,133,243,180,227,177,205,209,203,144
PP 600 DATA 14,208,102,200,196,243,208,243,232,228
JL 610 DATA 242,208,232,240,90,160,0,177,205,153
AM 620 DATA 0,6,200,196,226,208,246,160,0,177
MC 630 DATA 203,145,205,200,196,226,208,247,240,4
OG 640 DATA 80,187,176,182,160,0,185,0,6,145
EJ 650 DATA 203,200,196,226,208,246,165,216,197,218
OB 660 DATA 165,217,229,219,208,8,165,203,133,240
AP 670 DATA 165,204,133,241,56,165,216,229,224,133
OF 680 DATA 216,165,217,229,225,133,217,144,16,56
DL 690 DATA 165,203,229,222,133,203,165,204,229,223
GE 700 DATA 133,204,184,80,31,230,218,208,2,230
BA 710 DATA 219,165,219,133,217,165,218,133,216,24
CJ 720 DATA 165,240,101,226,133,203,133,240,165,241
MN 730 DATA 105,0,133,204,133,241,165,220,197,218
```

```
KP 740 DATA 165,221,229,219,
        144,2,176,150,184,80
FA 750 DATA 145
OO 760 REM <<< CREATE STRING
         ASSIGNMENT LINES >>>
FB 770 LINE=10000:? "{CLEAR}
       ":POSITION 2,4:? LINE
       ;" DIM TSORT$(331)":?
        "CONT":POSITION 2,0:
       POKE 842,13:STOP
PL 780 POKE 842,12
ED 790 FIN=0:I=0:UNP=0:DIM U
       NP(20),UNP$(20):FOR C
       =0 TO 20:UNP(C)=0:NEX
       T C:LINE=LINE+10:GOTO
        810
EN 800 I=I+80:LINE=LINE+10
BB 810 ? "{CLEAR}";:POSITION
        2,4:? LINE;" TSORT$(
       ";I+1;")=";CHR$(34);
EE 820 FOR J=1 TO 80
AM 830 IF I+J<=LEN(TSORT$) T
       HEN C=ASC(TSORT$(I+J)
       ):GOTO 850
CC 840 J=80:FIN=1:GOTO 870
MB 850 IF C<>34 AND C<>155 T
       HEN ? CHR$(27);CHR$(C
       );:GOTO 870
AM 860 ? " ";:UNP(UNP)=I+J:U
       NP=UNP+1:UNP$(UNP,UNP
       )=CHR$(C)
CI 870 NEXT J
MP 880 ?
FI 890 ? "CONT"
JH 900 POSITION 2,0
HH 910 POKE 842,13:STOP
GJ 920 POKE 842,12:IF FIN TH
       EN 940
BN 930 GOTO 800
HP 940 IF UNP=0 THEN 1030
FE 950 I=0
LI 960 LINE=LINE+10
OB 970 ? "{CLEAR}":POSITION
       2,4
FN 980 ? LINE;" TSORT$(";UNP
       (I);")=CHR$(";ASC(UNP
       $(I+1));")"
KN 990 ? "CONT":POSITION 2,0
       :POKE 842,13:STOP
BM 1000 POKE 842,12
CC 1010 I=I+1:IF I=UNP THEN
        1030
JL 1020 GOTO 960
DI 1030 POSITION 2,4:LINE=LI
        NE+10:? LINE;" REM L
        ENGTH IS  ";LEN(TSOR
        T$);" CHARACTERS."
AI 1040 ? LINE+10;" RETURN":
        ? "CONT"
OA 1050 POSITION 2,0:POKE 84
        2,13:STOP
JH 1060 POKE 842,12:POKE 752
        ,0:END
```

All the programs in this issue are available on the ready-to-load *COMPUTE! Disk*. To order a one-year (four-disk) subscription, call toll free **1-800-727-6937** Please specify which computer you are using.

SINCE 1982

ComputAbility

Mon-Fri 9am-9pm CST
Sat 11am-5pm

Mon-Fri 9am-9pm CST
Sat 11am-5pm

THE HOME COMPUTER STORE THAT'S AS CLOSE AS YOUR TELEPHONE

NEW

star micronics

NX-1000 PRINTER
•144 CPS-Draft •30CPS-NLQ
•NLQ in 10 & 12 Pitch
•6 Built-In Fonts

ONLY **$175**

Panasonic Office Automation OA

1080i - II PRINTER
NEW Now at 144 CPS

ONLY **$169**

PC COMPATIBLE

VENDEX PC
512K, 8 MHZ, 2DD, 7 Slot
Serial, Parallel, Mouse & Game Ports
$1000 of Software incl.
MS DOS & BASIC
Mono $829.00
Color $999.00

BLUE CHIP PC
BLUE CHIP AT
BLUE CHIP POPULAR
512K Monitor & 3.2 DOS
One or Two Disk Drive
Limited Offer While Supplies Last

We Carry the Complete PACKARD-BELL PC LINE
CALL for your Configuration

Panasonic Office Automation OA

1091i - II PRINTER
NEW Now at 192 CPS

ONLY **$189**

Panasonic Office Automation OA

1092i PRINTER

ONLY **$299**

commodore®

1541C Disk Drive
1571 Disk Drive
1581C Disk Drive
64C Computer
C-128 Computer
C-128 D Computer
1802C Monitor
2002 Monitor

CALL FOR INDIVIDUAL & SUPER PACKAGE PRICES

1750 RAM Expander
1764 RAM Expander
Avatex 1200 HC Modem
Avatex 2400 HC Modem
1351 Mouse
I CON Troller

CALL FOR CURRENT PRICE

AMIGA® 500
COMPUTER SYSTEM
CALL FOR CURRENT PRICES & AVAILABILITY

ATARI
520 ST & 1040 ST
SYSTEM PACKAGES
CALL FOR LATEST. PRICES

IBM	
Ace of Aces	24.95
Ancient Art of War at Sea	27.95
Ancient Art of War	25.95
Amnesia	29.95
Archon	10.95
Arctic Fox	21.95
Adv Construction	10.95
Battlezone	14.95
Bards Tale	32.95
Bridge 5.0	18.95
California Games	25.95
Copy II PC	27.95
CMS 20 Meg H/D	349
Chessmaster 2000	25.95
Chuck Yeager F.S.	26.95
Dac Easy Accounting 2.0	58.95
Defender of the Crown	24.95
Destroyer	25.95
Dig Dug	14.95
Digital Solitons	CALL
DOS RX	36.95
Earl Weaver	26.95
Empire	33.95
Grand Slam Bridge	36.95
Gunship	30.95
Hardball	24.95
Instant Pages	33.95
Karateka	21.95
Loderunner	21.95
Marble Madness	25.95
Mean18	30.95
Moon Patrol	14.95
Monopoly	26.95
3.2 MS DOS GW Basic	94.95
MS Pac Man	14.95
Music Construction	10.95
One on One	10.95
Ogre	24.95
Pac Man	14.95
P.C. Quintet	59.95
Pinball Wiz	21.95
Publish It	69.95
Print Shop	36.95
PSI 5 Trading Co	14.95
Seven Cities of Gold	10.95
Sylvia Porter Personal Invest	69.95
Starfight	32.95
Starfleet 2	34.95
Sub Battle	25.95
Subogic Football	34.95
Super Boulder Dash	10.95
Tag Team Wrestling	25.95
Test Drive	24.95
Turbo Basic	64.95
Up Periscope	19.95
Ultima 1,2 or 3	CALL
Universe II	47.95
Where in the U.S.A is Carmen Sandiego	28.95
Where in the World is Carmen Sandiego	25.95
World Tour Golf	32.95
World Class L.B	28.95
1st Expedition	33.95

AMIGA	
Aegis Animator	84.95
Aegis Draw	159.95
Alternate Reality/City	26.95
Analyze 2.0	94.95
Amiga 501 Expansion	158.56
Barbarians	25.95
Bards Tale	32.95
Beyond Zork	32.95
Black Cauldron	26.95
Championship Baseball	25.95
Championship Golf	25.95
Chessmaster 2000	29.95
Defender of the Crown	32.95
Deja Vu	32.95
Deluxe Music 2.0	64.95
Deluxe Paint 2	94.95
Deluxe Print 2	51.95
Deluxe Productions	124.95
Deluxe Video 1.2	94.95
Deluxe Write	69.95
Earl Weaver Baseball	32.95
Empire	33.95
Fairy Tale Adventure	31.95
Ferrari Formula 1	33.95
First Shapes	33.95
Flight Simulator II	33.95
GFL Football	29.95
Goldspell	27.95
Grid Iron	44.95
Gum Ball Air Rally	25.95
Indoor Sports	32.95
Intellitype	33.95
Kampgruppe	39.95
Kid Talk	33.95
Kings Quest 1,2 or 3	33.95
Knight Orc	29.95
Leisure Suit Larry	33.95
Logistix	89.95
Marble Madness	32.95
Mavis Beacon Typing	29.95
Mean 18	27.95
Music Studio	32.95
On Line	42.95
Organize	62.95
Page Setter	89.95
Pawn	29.95
Phantasie 1, 2 or 3	25.95
Print Master Plus	31.95
Publisher	119.95
Roadwar Europa	25.95
Scribble	62.95
S.D.I	32.95
Shanghai	25.95
Silent Service	25.95
Stargider	29.95
Superbase Personal	89.95
Test Drive	27.95
Terror Pods	24.95
Two on Two Basketball	29.95
Ultima IV	39.95
Video Vegas	21.95
Vizawrite Amiga	89.95
Winter Games	25.95
World Games	25.95

C128	
Abacus Book	CALL
Basic Compiler 128	39.95
Beyond Zork	29.95
Cad Pak 128	39.95
Chart Pak128	25.95
Cobol 128	39.95
Datamanager128	42.95
Fleet System 4	46.95
Fontmaster 128	36.95
Geos 128	44.95
GeoCalc 128	44.95
Geo Dex 128	44.95
Geo File 128	44.95
GeoWrite 128	44.95
Mach 128	37.95
Merlin 128	69.95
Multiplan 128	27.95
Paperclip II/128	CALL
Partner 128	42.95
Pocket Filer 2	36.95
Pocket Planner 2	36.95
Pocket Writer 2	36.95
Speed Term 128	25.95
Superbase 128	57.95
Super C Compiler	39.95
Super Pak2/128	59.95
Super Pascal 128	39.95
Superscript 128	46.95
Swiftcalc 128	42.95
Sylvia Porter 128	42.95
Trinity	25.95
Wordwriter 128	42.95
Visawrite	54.95

APPLE	
Airheart	22.95
Animate	44.95
Autoduel	32.95
Bards Tale	29.95
Bards Tale II	32.95
Bop-N Wrestle	19.95
Certificate Maker	25.95
Certificate Maker Lib #1	22.95
Clip Art 1 or 3	19.95
Clip Art 2	25.95
Destroyer	25.95
F-15 Strike Eagle	22.95
Fight Night	21.95
Graphics Lib 1,2or3	16.95
Hardball	21.95
Infiltrator	19.95
Jet	27.95
Newsroom	37.95
On Balance	64.95
Pawn	29.95
Pegasus	25.95
Phantasie 1,2 or 3	25.95
Print Shop	31.95
Silent Service	22.95
Stargider	29.95
Street Sports Baseball	25.95
Street Sports Basketball	25.95
Tag Team Wrestling	22.95
Where in the U.S.A. is Carmen Sandiego	29.95
Where in the World is Carmen Sandiego	25.95

ATARI 8-BIT	
Ace of Aces	18.95
Animation Station	59.95
Atariwriter Plus	39.95
Auto Duel XE/XL	34.95
Bop-N Wrestle	19.95
Chessmaster 2000	25.95
F-15 Strike Eagle	22.95
Guild of Thieves	27.95
Hardball	18.95
Jewels of Darkness	17.95
Leaderboard	24.95
Leather Goddess	27.95
Movie Maker	10.95
Music Studio	22.95
Page Designer	18.95
Pawn XE/XL	34.95
Print Shop	28.95
P.S. Companion	22.95
P.S. Interface	18.95
Racing Destruction	10.95
Rubber Stamp	18.95
Silent Service	22.95
Silicon Dreams	17.95
Spitfire 40	23.95
Syncalc	32.95
Synfile	32.95
Top Gunner	16.95
Ultima II	31.95
Ultima IV	38.95
Video Vegas XE/XL	20.95
Wizards Crown	27.95
Zork Trilogy	38.95

ATARI ST	
Alien Fires	25.95
Art Gallery 1or2	18.95
Auto Duel	32.95
Award Maker	24.95
Balance of Power	32.95
Bard's Tale	33.95
BBS Express	56.95
Certificate Maker	25.95
Chessmaster 2000	32.95
Dac Easy Accounting 2.0	64.95
Dac Payroll	32.95
Dark Castle	25.95
Datamanager ST	48.95
Defender of the Crown	32.95
Deluxe Print 2	34.95
Degas Elite	52.95
Desk Cart	72.95
Easy Draw	48.95
Empire	33.95
EZ Track	39.95
F-15 Strike Eagle	27.95
Flash	19.95
Flight Simulator	33.95
Gauntlet	32.95
GFA Basic	49.95
GFA Compiler	49.95
GFA Companion	32.95
GFA Draft	64.95
GFA Vector	36.95
Gold Runner	24.95
Hardball	27.95
In to the Eagle's Nest	25.95
Knight Orc	29.95
Labelmaster Elite	27.95
LDW Basic	44.95
Master Ninja	22.95
Marble Madness	24.95
Mark Williams	114.95
Math Talk	31.95
Moebius ST	38.95
Paper Boy	32.95
Partner ST	48.95
Pawn	29.95
P.C. Ditto	69.95
Personal Pascal 2.0	64.95
Phantasie 1,2or3	25.95
Pinball Wizard	21.95
Printmaster Plus	24.95
Print Shop	31.95
Publishing Partner	64.95
Sentry	29.95
Silent Service	25.95
Soalter Bee	31.95
ST Art Director	48.95
Stargider	29.95
Sundog	24.95
Swiftcalc ST	48.95
Tech 3D Glasses	149.95
Test Drive	24.95
Terrapods	24.95
Thunder	26.95
Time Bandits	24.95
Word Perfect	259.95
Wordwriter ST	48.95

C64	
Bard's Tale	25.95
Bard's Tale 2	25.95
Basic Compiler 64	25.95
Bop 'n Rumble	18.95
Cad Pak 64	25.95
Carmen U.S.A	28.95
Certificate Maker	25.95
Champ. Baseball	18.95
Chart Pak	25.95
Chessmaster 2000	25.95
Chuck Yeager	24.95
Data Manager II	25.95
Defender of the Crown	22.95
Demon Stalkers	20.95
Desk Pak I	22.95
Desk Pak 2	32.95
Desktop Publisher	39.95
Echelon	29.95
Eternal Dagger	25.95
Fast Load	24.95
Fleuntraw	CALL
Final Cartridge 3	42.95
Flight Simulator II	33.95
Font Master II	29.95
Geocalc	34.95
Geo Chart	27.95
Geodex	25.95
Geofile	34.95
Geopublish	34.95
Geos	37.95
Gunship	22.95
Last Ninja	22.95
Legacy/Ancients	20.95
Maniac Mansion	22.95
Might & Magic	25.95
Newsroom	30.95
Omnicron Conspiracy	24.95
Paper Boy	22.95
Pegasus	23.95
Pirates	25.95
Printmaster Plus	21.95
Printshop	28.95
P.S. Companion	22.95
S.D.I.	22.95
Skate or Die	21.95
Stargider	25.95
Sub Battle	24.95
ST. Sports Baseball	24.95
ST. Sports Basketball	24.95
Super Bike Challenge	18.95
Super Pak 2/64	59.95
Super Star Hockey	22.95
Swiftcalc /Sideways	25.95
2 on 2 Basketball	22.95
Test Drive	18.95
The Gauntlet	22.95
Top Fuel Eliminator	18.95
Ultima 1	25.95
Ultima 5	38.95
War In S. Pacific	38.95
Wordwriter 3	32.95
Wld Class Leader	25.95
World Tour Golf	23.95
WWF Wrestling	25.95

SINCE1982

ComputAbility
Consumer Electronics

P.O. BOX 17882, MILWAUKEE, WI 53217
ORDER LINES OPEN
Mon-Fri 9am-9pm CST Sat 11am-5pm
To Order Call Toll Free
800-558-0003
For Technical info, Order Inquiries, or for Wisc. Orders
414-357-8181
MasterCard
TELEX NUMBER 9102406440
(ANSERBACK-COMPUT MILW UO)
VISA

No surchage for Mastercard or Visa

ORDERING INFORMATION: Please specify system. For fast delivery send cashier's check or money order. Personal and company checks allow 14 business days to clear. School P.O.'s welcome. C.O.D. charges are $3.00. In Continental USA include $3.00 for software orders 5% shipping for hardware minimum $5.00. MasterCard and Visa orders please include card #, expiration date and signature. WI residents please include 5% sales tax. HI, AK, FPO, APO, Puerto Rico and Canadian orders, please add 5% shipping, Minimum $6.00. All other foreign orders add 15% shipping, minimum $15.00. All order shipped outside the Continental U.S.A are shipped first class insured U.S. mail. If foreign shipping charges exceed the minimum amount, you will be charged the additional amount to get your package to you quickly and safely. All goods are new and include factory warranty. Due to our low prices all sales are final. All defective returns must have a return authorization number. Please call (414)357-8181 to obtain an RA# or your return will not be accepted. Prices and availability subject to change without notice.

Finding And Financing A College Education

Peterson's Guides has released two software programs to help high school students and their families select a college and determine how to obtain financial aid.

Peterson's College Selection Service 1988 helps identify colleges with programs and offerings that are of interest to a particular student. The menu-driven program assists students by eliminating colleges from their list according to their responses. Selection criteria include geographic location, size, control (public or private), entrance difficulty, majors, costs, academic and honors programs, ethnic and geographic mix of the student body, admission requirements, sports, housing, campus life, and others. The program includes a complete index of majors.

Users can include as many options as they wish. The program also explains why a particular college was not included in the final list.

College Selection Service is updated annually and includes facts and figures about every four-year and two-year college and university in the United States and Canada.

The four-year program and the two-year program can be purchased separately or combined, and are available for the Apple II series, the IBM PC, and the IBM PC/XT. It is also available for the TRS-80 Model III/Model 4 and other MS-DOS computers for four-year colleges only. The four-year version retails for $159 ($184 for the IBM PC/XT), while the two-year version sells for $109 ($134 for the IBM PC/XT). The combined version is available for $188 ($238 for the IBM PC/XT).

Once a student has selected a college, the next problem is financing his or her education. *Peterson's Financial Aid Service* leads the user step-by-step through the process of estimating need and then identifying possible sources of need-based and non-need based aid for each specific case. Families can use the program as early in the college-planning process as they desire and can project probable cost and need in the privacy of their own home.

The program was developed in consultation with a recognized expert on college financial aid, using the Congressional Methodology, which is the new need analysis formula based on the 1986 Tax Reform law. It is the same methodology used by colleges and the federal government in allocating aid.

Peterson's Financial Aid Service is available for the Apple II Series and the IBM PC for $195. Suggested retail price for the IBM PC/XT version is $220.

Peterson's Guides, P.O. Box 2123, Princeton, NJ 08543-2123

Circle Reader Service Number 200.

Thunder Mountain Brings Back Arcade Classics

Thunder Mountain, a division of Mindscape, has released home computer versions of popular Atari/Namco arcade games. The five titles available are *Pac Man, Ms. Pac Man, Dig Dug, Pole Position,* and *Galaxian.*

The games are available for the Commodore 64 and 128, the IBM PC, and Apple II computers. The suggested retail price for each game is $9.95.

Thunder Mountain, 3444 Dundee Rd., Northbrook, IL 60062

Circle Reader Service Number 201.

New Version of EZ-Forms

EZX Corporation has released *EZ-Forms Executive Version 1.3* for the IBM PC. The original program allowed users to create, fill out, and store custom generated forms. The new version enables users to incorporate automatic math functions and merge features into their custom forms. New applications include point-of-sale receipts, error-free invoicing, instant inventory values, and any automatic calculations needed within a form.

Suggested retail price is $89.

EZX Corporation, 403 Nasa Rd. One East, Webster, TX 77598

Circle Reader Service Number 202.

Two New Jewells

Jewell Technology has released two new utilities for IBM PCs and compatibles.

Vertigo rotates printouts—spreadsheets, flow charts, documents, presentations, and reports—90 degrees. It works with most printers. With this utility, users can specify the number of characters per inch, lines per inch, and can select one of 12 image enhancing fonts.

Most spreadsheet applications including *LOTUS 1-2-3, Symphony,* and *Multiplan* can work in conjunction with *Vertigo.* The utility can drive most dot-matrix, laser, or inkjet printers, and can print the entire IBM character set.

Suggested retail price is $49.95.

Also available from Jewell Technologies is *Grafplus,* which can turn any IBM or compatible application into a communicator and graphics master by dumping graphics directly from the computer screen to the printer. The utility has the capabilities to produce files compatible with Xerox' *Ventura Publishing* and Aldus' *Pagemaker.*

Grafplus can enhance documents, reports, and presentations by scaling images to any size, including larger or smaller than the screen size. It can print it in color, black-on-white, white-on-black, horizontally or vertically on the page, and in graphics or text mode.

The utility supports all dot-matrix, inkjet, or laser printers with bitmapped graphics. *Grafplus* also includes *Editgraf,* an editing program which can support printers that may be available in the future.

Suggested retail price is $49.95.

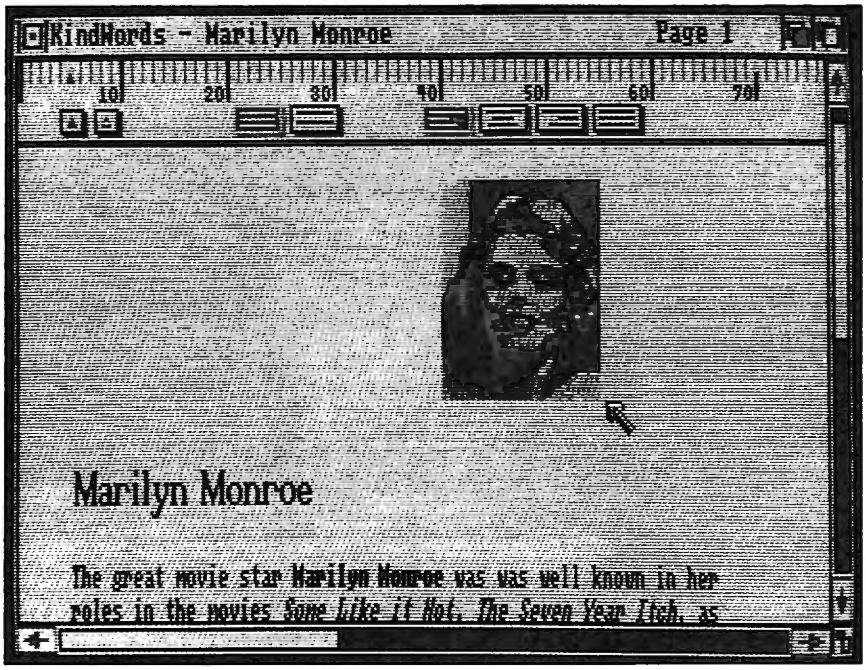
Jewell Technologies, 4740 44th Ave. SW #203, Seattle, WA 98116

Circle Reader Service Number 203.

Amiga Word Processor

The Disc Company has developed and released *KindWords,* a word processor for the Commodore Amiga. The program features a 90,000-word spell checker that suggests corrections, offers a choice of multiple font styles and sizes, and includes color graphics in text.

KindWords offers a fully integrated graphics environment that enables users to crop and manipulate image size in 16 colors within the word processing application. SuperFonts, the specially created printer fonts, allow most printers to output high-resolution printing. Other features include true what-you-



KindWords *offers a fully integrated graphics environment.*

see-is-what-you-get, unlimited document length, global select command, and find and replace.

Suggested retail price is $99.95.

The Disc Company, 3135 South State St., Ann Arbor, MI 48108

Circle Reader Service Number 204.

Pyramids And Planets

Sir Tech Software has released two new games for the IBM 5.25-inch disk format.

The Seven Spirits of Ra is a graphic realtime interactive simulation adventure based on the ancient Egyptian "Land of the Dead" myth. The game simulates the trials and tribulations of the legendary god-king Osiris in the Duad. The myth tells how Osiris overcame death and became immortal. The player, as a participant in the myth, is responsible for guiding Osiris through the mysterious Duad to overcome Set, the Dark One who has sought to take Orisis' throne for himself.

The program, which features 16-color high-resolution animated graphics, is illustrated with Egyptian art and is supported by documentation that explains the myth. Players can choose from two levels of difficulty and can use a keyboard interface or a joystick.

Available for IBM PCs and 100-percent compatibles, and the Tandy 1000, the game requires 128K RAM and either a CGA or EGA, and an RGB color monitor or TV. Suggested retail price is $49.95.

Deep Space—Operation Copernicus is a space combat simulation arcade-style game. The game is set in the year 2123 where genetically-programmed troops from the Andromedan Hegemony are trying to takeover the United Planets of Solaria's mines in the Asteroid Belt. On Earth, leaders are trying to build up their forces to resist the alien forces.

The player becomes a recruit of the Solarian Armed Forces and pilots a Katana single-seat interceptor to fight the invading Andromedans in a choice of four missions.

During the missions, all objects follow the actual laws of motion observed in space and are displayed in 3-D animated graphics. Either the keyboard or a joystick can be used as an interface.

The game is available for Apple and Commodore computers with 64K and IBM, Tandy, and compatibles with at least 256K, and a CGA or EGA card. All require one disk drive, and a color monitor is recommended but not required.

Suggested retail price is $39.95.

Sir-Tech Software, P.O. Box 245, Charlestown Ogdensburg Mall, Ogdensburg, NY 13669

Circle Reader Service Number 205.

New Utility For *Turbo Pascal*

Synergy Systems has released *Instant-Compile,* which is a new utility for programmers that use Borland International's *Turbo Pascal.* The utility reduces the time that programmers using *Turbo Pascal* must spend in repetitively compiling code that is already working properly. With *Instant-Compile,* programmers can begin their compilation with the new code they're developing.

After the programmer makes an error and then corrects it, he can return the compilation in about ¼ second to the state it was in at the time of the save. The utility also looks ahead for the next error.

Instant-Compile is a memory-resident program that is executed before *Turbo Pascal.* Each time the Turbo compiler encounters a directive in the source code, the state of the compilation is saved, allowing the programmer to direct the compilation to begin from the last of these saved states, rather than from the first line of the program.

Suggested retail price is $49.

Synergy Systems, 1124 Oak Glen Circle, Fort Worth, TX 76114

Circle Reader Service Number 206.

Keep It Under Cover

Kensington Microware Ltd. has introduced a complete line of dust covers for IBM PCs. The covers are anti-static and are made of translucent vinyl, and are machine stitched. The line includes covers for the PC, XT, and AT, as well as all models in the new Personal System/2 line.

Kensington also offers dust covers for Macintosh and Apple computers.

Suggested retail price for all of the IBM covers is $19.95 each.

Kensington Microware Ltd., 251 Park Ave. South, New York, NY 10010

Circle Reader Service Number 207.

Build Your Own Clone

Friendly Videos has released How To Build Your Own IBM PC/AT Clone, the latest in their series of Computer How-To videos.

The 48-minute video (VHS format) covers sourcing parts, assembly, initialization, and troubleshooting of an AT compatible computer. The video is designed for both first time users and experienced computer enthusiasts. Users are given instructions on how to assemble their own computer in a few hours and with only a screwdriver.

Suggested retail price is $29.95.

Friendly Videos, P.O. Box 5684, Garden Grove, CA 92645

Circle Reader Service Number 208.

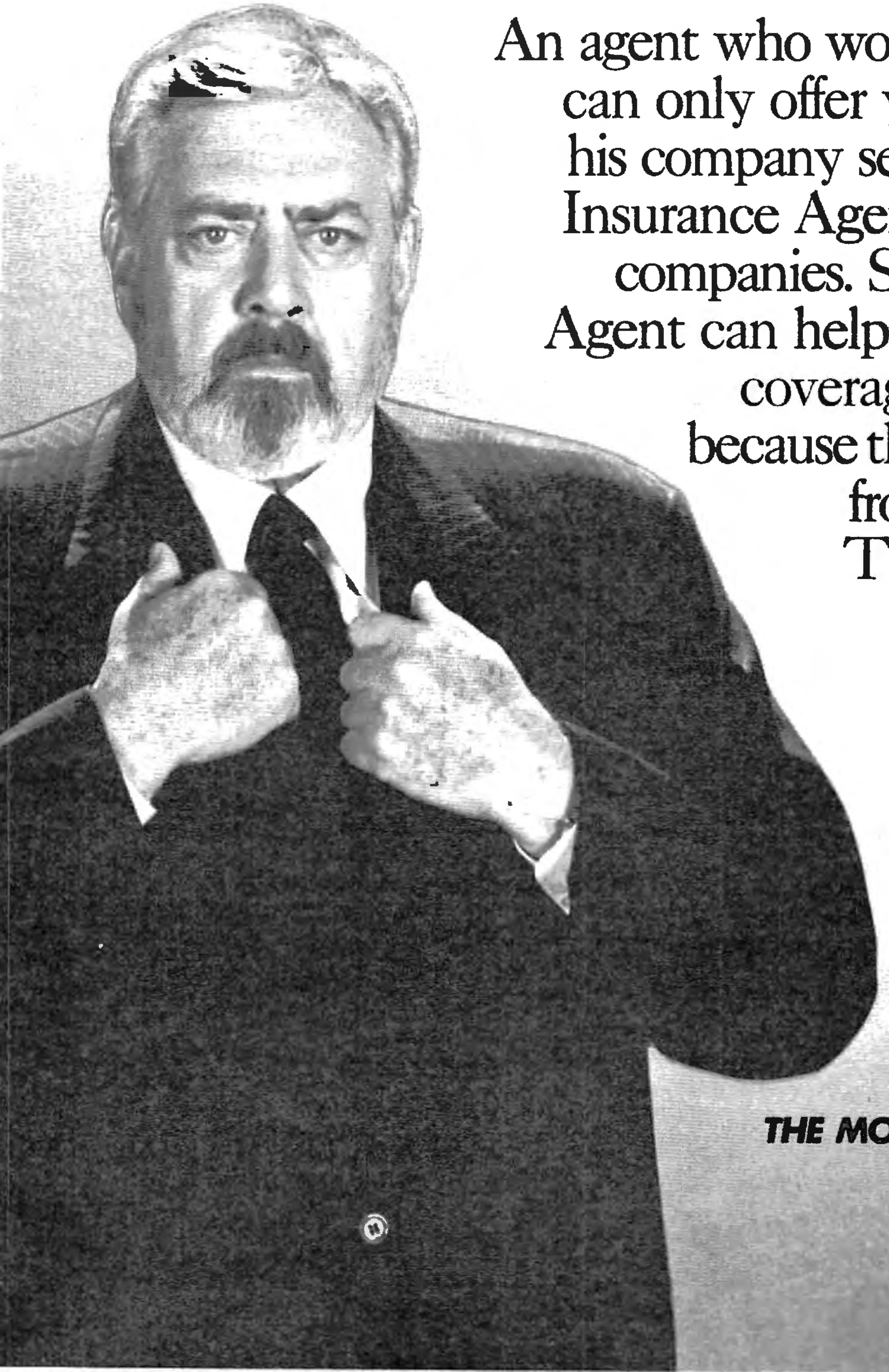
Amstrad Portable

Amstrad has introduced the PPC 640 portable computer that comes complete with a 2400-bps modem. The company is also releasing a less-powerful version without a modem, the PPC 512.

The PPC 640 is compatible with IBM XT software, weighs 11¾ pounds, and features an 8086 8-MHz, 16-bit processor. The computer has an 80 x 25 row format supertwist LCD display, single or dual 3½-inch disk drives, 640K RAM, DOS 3.3, and 300/1200/

"Before you buy insurance, examine the evidence."

RAYMOND BURR

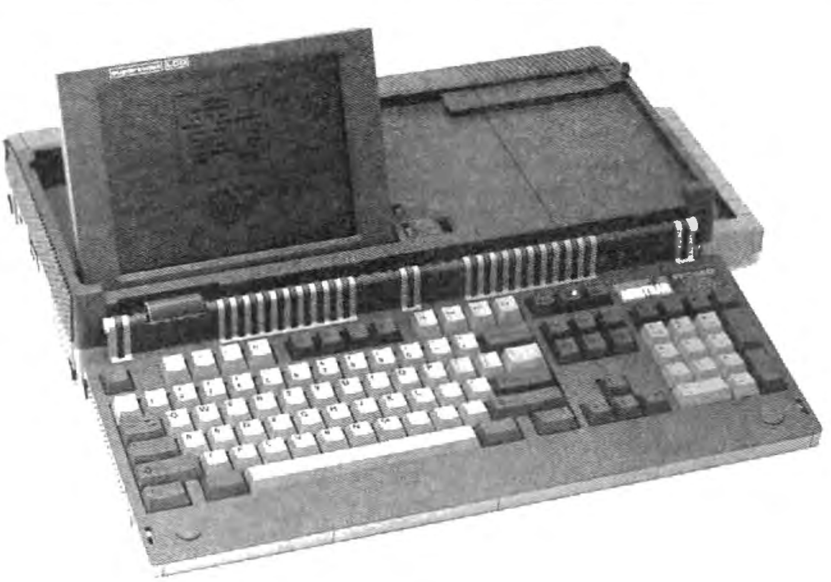


An agent who works for one company can only offer you the policies that his company sells. An Independent Insurance Agent represents several companies. So your Independent Agent can help you select the right coverage at the right price because there are more policies from which to choose. The evidence is clear.

Independent Insurance Agent,

THE MORE-THAN-ONE-COMPANY INSURANCE AGENT.

You'll find the Independent Insurance Agent nearest you in the Yellow Pages.



The Amstrad PPC 640 has its own built-in modem.

2400 baud Hayes compatible modem. Another feature is its full size AT-style enhanced keyboard.

The suggested retail price for the single drive is $999, while the dual drive sells for $1,099.

The PPC 512 has 512K of memory and also comes with MS-DOS. The single drive retails for $799, and the dual drive is $899.

Both computers feature a five-way power supply, parallel and serial expansion ports, and bus for an optional add-on-board for hard drive and/or expansion board use. The computers can run on AC, an automobile cigarette lighter, an Amstrad PC 1640's power supply, or regular C flashlight batteries. A nine-pin din plug can also be used with any PC compatible monitor.

Amstrad, 1915 Westridge Dr., Irving, TX 75038

Circle Reader Service Number 209.

Elect The Next President And Win!

Strategic Simulations has announced a contest based on their *President Elect—1988 Edition*, a strategic computer simulation of presidential elections. Amateur political analysts can compete for up to $1,988 by deciding who will win the next election.

The contest is open to any U.S. resident of any age. The Players whose Democratic and Republican candidates are the same as those on the November 8, 1988 ballot, and whose electoral vote count comes closest to matching the actual election results will win a cash prize.

Each contestant has two opportunities to win: the Pre-Primary phase and the Campaign phase. Entries for the Pre-Primary phase must be postmarked no later than February 12, 1988, while the Campaign phase must be postmarked no later than October 28, 1988. The winner of each phase will collect $944. The contestants must submit a photograph/slide of the *President Elect* computer monitor screen labeled "Electoral Vote Total," that shows the candidates and their final electoral vote counts. Complete contest rules can be found in game packages or by writing Strategic Simulations.

The game is available for the Apple II series, the Commodore 64 and 128, the Atari ST, and IBM PCs and compatibles, and has a suggested retail price of $24.95.

Strategic Simulations, 1046 N. Rengstorff Ave., Mountain View, CA 94043

Circle Reader Service Number 210.

Golfer Psyche

Golfer Style Analysis from SportsPsych is designed to assist golfers in improving their games. A teaching professional or the average golfer himself can use the program to identify the personality characteristics and traits that have an impact on the golfer's performance.

The program is designed to teach the golfer about himself and what his tendencies are as a player. Instructors can detect tendencies in a student's game that could only before be discovered over a long period of time.

Users are asked to respond to 24 questions designed to identify personality traits. The program then evaluates the golfer and gives advice and recommendations for improving the mental approach to the game and the best methods to coach him.

The program also informs the user which other types of golfers he is most compatible with and what type of impact he has on his playing partners.

SportsPsych developed the program at the request of the Professional Golfers' Association, which is considering approval of the program for use by their member professionals.

Golfer Style Analysis is available for IBM PCs for a suggested retail price of $59.95.

SportsPsych also offers *Football Style Analysis* and *Basketball Style Analysis.*

SportsPsych, 810 12th St. N.W., Mason City, IA 50401

Circle Reader Service Number 211.

Graphic Space And Underworld Adventures

Firebird has released *Silicon Dreams* and *Jewels Of Darkness* in multiple machine formats. The two titles both consist of three smaller adventure games that combine aspects of the various segments to complete the games full-term.

Players of *Silicon Dreams* assume the role of colonizer of the planet Eden in the twenty-third century. In the first scenario, Snowball, the player is en route to Eden when he's awakened by a malfunction in his spacecraft. The crew of the Snowball then travels to the Robot City of Enoch in Return to Eden, the second scenario. Finally, the player must identify and solve the problem in Enoch in the final scenario, the Worm in Paradise.

Silicon Dreams is available for the Commodore 64 and 128, Atari 8-bit computers, and the Apple II series at a suggested retail price of $24.95 each. Amiga, Atari ST, Macintosh, and IBM/Tandy/compatible PC versions are also available for $29.95 each.

Jewels Of Darkness takes place in the underground Land of the Demon Lord. In the Colossal Adventure, the player's quest is to implement a dramatic rescue of the goodly elves imprisoned in the Demon Lord's dungeons. Next, in the Adventure Quest, the player must prove himself to the Wizard's High Council by defeating the Demon Lord. After conquering the Demon Lord, the player moves on to the Dungeon Adventure to obtain glory and greater riches.

The game is available for the Commodore 64 and 128, and Atari 8-bit computers for $24.95. Amiga, Atari ST, IBM/Tandy/Compatible, and Macintosh versions are also available for $29.95.

Firebird, 71 N. Franklin Turnpike, Waldwick, NJ 07463

Circle Reader Service Number 212.

COMPUTE!'s Guide To Typing In Programs

Computers are precise—type the program *exactly* as listed, including necessary punctuation and symbols, except for special characters noted below. We have provided a special listing convention as well as a set of programs to check your typing—"The Automatic Proofreader."

Programs for the IBM and those in ST BASIC for Atari ST models should be typed exactly as listed; no special characters are used. Programs for Commodore, Apple, and Atari 400/800/XL/XE computers may contain some hard-to-read special characters, so we have a listing system that indicates these control characters. You will find these characters in curly braces; *do not type the braces*. For example, {CLEAR} or {CLR} instructs you to type the character which clears the screen on the Atari or Commodore machines. A complete list of these symbols is shown in the tables below. For Commodore, Apple, and Atari, a single symbol by itself within curly braces is a control key or graphics key. If you see {A}, hold down the CONTROL key and press A. This will produce a reverse video character on the Commodore (in quote mode), a graphics character on the Atari, and an invisible control character on the Apple.

For Commodore computers, graphics characters entered with the Commodore logo key are enclosed in a special bracket: [<A>]. In this case, you would hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as S̲. One exception is {SHIFT-SPACE}. When you see this, hold down SHIFT and press the space bar. If a number precedes a symbol, repeat the character the indicated number of times. For example, {5 RIGHT}, {6 S̲}, and [<8 Q>], mean, respectively, that you should enter five cursor rights, six shifted S's, and eight Commodore-Q's. On the Atari, inverse characters (white on black) should be entered with the inverse vid-

Atari 400/800/XL/XE

When you see	Type	See	
{CLEAR}	ESC SHIFT <	↖	Clear Screen
{UP}	ESC CTRL -	↑	Cursor Up
{DOWN}	ESC CTRL =	↓	Cursor Down
{LEFT}	ESC CTRL +	←	Cursor Left
{RIGHT}	ESC CTRL *	→	Cursor Right
{BACK S}	ESC DELETE	◀	Backspace
{DELETE}	ESC CTRL DELETE		Delete character
{INSERT}	ESC CTRL INSERT		Insert character
{DEL LINE}	ESC SHIFT DELETE		Delete line
{INS LINE}	ESC SHIFT INSERT		Insert line
{TAB}	ESC TAB	▶	TAB key
{CLR TAB}	ESC CTRL TAB		Clear tab
{SET TAB}	ESC SHIFT TAB		Set tab stop
{BELL}	ESC CTRL 2		Ring buzzer
{ESC}	ESC ESC		ESCape key

Commodore PET/CBM/VIC/64/128/16/+4

When You Read:	Press:	See:	When You Read:	Press:	See:
{CLR}	SHIFT CLR/HOME		[< 1 >]	COMMODORE 1	
{HOME}	CLR/HOME		[< 2 >]	COMMODORE 2	
{UP}	SHIFT ↑ CRSR ↓		[< 3 >]	COMMODORE 3	
{DOWN}	↑ CRSR ↓		[< 4 >]	COMMODORE 4	
{LEFT}	SHIFT ← CRSR →		[< 5 >]	COMMODORE 5	
{RIGHT}	← CRSR →		[< 6 >]	COMMODORE 6	
{RVS}	CTRL 9		[< 7 >]	COMMODORE 7	
{OFF}	CTRL 0		[< 8 >]	COMMODORE 8	
{BLK}	CTRL 1		{ F1 }	f1	
{WHT}	CTRL 2		{ F2 }	SHIFT f1	
{RED}	CTRL 3		{ F3 }	f3	
{CYN}	CTRL 4		{ F4 }	SHIFT f3	
{PUR}	CTRL 5		{ F5 }	f5	
{GRN}	CTRL 6		{ F6 }	SHIFT f5	
{BLU}	CTRL 7		{ F7 }	f7	
{YEL}	CTRL 8		{ F8 }	SHIFT f7	
			←	←	

eo key (Atari logo key on 400/800 models).

Whenever more than two spaces appear in a row, they are listed in a special format. For example, {6 SPACES} means press the space bar six times. Our Commodore listings never leave a single space at the end of a line, instead moving it to the next printed line as {SPACE}.

Amiga program listings and Atari ST program listings in GFA BASIC contain only one special character, the left arrow (←) symbol. This character marks the end of each program line. Wherever you see a left arrow, press RETURN to enter that line into memory. (For the Amiga, you can also enter the line simply by moving the cursor off the line.) Don't try to type in the left arrow symbol; it's there only as a marker to indicate where each program line ends.

The Automatic Proofreader

Type in the appropriate program listed below, then save it for future use. The Commodore Proofreader works on the Commodore 128, 64, Plus/4, 16, and VIC-20. Don't omit any lines, even if they contain unfamiliar commands or you think they don't apply to your computer. When you run the program, it installs a machine language program in memory and erases its BASIC portion automatically (so be sure to save several copies before running the program for the first time). If you're using a Commodore 128, Plus/4 or 16, do *not* use any GRAPHIC commands while the Proofreader is active. You should disable the Commodore Proofreader before running any other program. To do this, either turn the computer off and on or enter SYS 64738 (for the 64), SYS 65341 (128), SYS 64802 (VIC-20), or SYS 65526 (Plus/4 or 16). To reenable the Proofreader, reload the program and run it as usual. Unlike the original VIC/64 Proofreader, this version works the same with disk or tape.

The IBM Proofreader is a BASIC program that simulates the IBM BASIC line editor, letting you enter, edit, list, save, and load programs that you type. Type RUN to activate. Be sure to leave Caps Lock on, except when typing lowercase characters.

On the Atari, run the Proofreader to activate it (the Proofreader remains active in memory as a machine language program); you must then enter NEW to erase the BASIC loader. Pressing SYSTEM RESET deactivates the Atari Proofreader; enter PRINT USR (1536) to reenable it.

The Apple Proofreader erases the BASIC portion of itself after you run it, leaving only the machine language portion in memory. It works with either DOS 3.3 or ProDOS. Disable the Apple Proofreader by pressing CTRL-RESET before running another BASIC program.

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a hexadecimal number (on the Apple) or a pair of letters (on the Commodore, Atari, or IBM) appears. The number or pair of letters is called a *checksum.*

Compare the value displayed on the screen by the Proofreader with the checksum printed in the program listing in the magazine. The checksum is given to the left of each line number. Just type in the program a line at a time (without the printed checksum), press RETURN or Enter, and compare the checksums. If they match, go on to the next line. If not, check your typing; you've made a mistake. Because of the checksum method used, do not type abbreviations, such as ? for PRINT. On the Atari and Apple Proofreaders, spaces are not counted as part of the checksum, so be sure you type the right number of spaces between quote marks. The Atari Proofreader does not check to see that you've typed the characters in the right order, so if characters are transposed, the checksum still matches the listing. The Commodore Proofreader catches transposition errors and ignores spaces unless they're enclosed in quotation marks. The IBM Proofreader detects errors in spacing and transposition.

IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you enter NEW, the Proofreader prompts you to press Y to be especially sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program as usual (this replaces the Proofreader in memory). You can now run the program, but you may want to resave it to disk. This will shorten it on disk and make it load faster, but it can no longer be edited with the Proofreader. If you want to convert an existing BASIC program to Proofreader format, save it to disk with SAVE *"filename"*,A.

Program 1: Atari Proofreader

By Charles Brannon

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:RE
    AD A:POKE I,A:CK=CK+A
    :NEXT I
120 IF CK<>19072 THEN ? "
    Error in DATA Stateme
    nts.  Check Typing.":
    END
130 A=USR(1536)
140 ? :? "Automatic Proof
    reader Now Activated.
    "
150 END
160 DATA 104,160,0,185,26
    ,3,201,69,240,7
170 DATA 200,200,192,34,2
    08,243,96,200,169,74
180 DATA 153,26,3,200,169
    ,6,153,26,3,162
190 DATA 0,189,0,228,157,
    74,6,232,224,16
200 DATA 208,245,169,93,1
    41,78,6,169,6,141
210 DATA 79,6,24,173,4,22
    8,105,1,141,95
220 DATA 6,173,5,228,105,
    0,141,96,6,169
230 DATA 0,133,203,96,247
    ,238,125,241,93,6
240 DATA 244,241,115,241,
    124,241,76,205,238
250 DATA 0,0,0,0,0,32,62,
    246,8,201
260 DATA 155,240,13,201,3
    2,240,7,72,24,101
270 DATA 203,133,203,104,
    40,96,72,152,72,138
280 DATA 72,160,0,169,128
    ,145,88,200,192,40
290 DATA 208,249,165,203,
    74,74,74,74,24,105
300 DATA 161,160,3,145,88
    ,165,203,41,15,24
310 DATA 105,161,200,145,
    88,169,0,133,203,104
320 DATA 170,104,168,104,
    40,96
```

Program 2: Commodore Proofreader

By Philip Nelson

```
10 VEC=PEEK(772)+256*PEEK(773)
   :LO=43:HI=44
20 PRINT "AUTOMATIC PROOFREADE
   R FOR ";:IF VEC=42364 THEN
   {SPACE}PRINT "C-64"
30 IF VEC=50556 THEN PRINT "VI
   C-20"
40 IF VEC=35158 THEN GRAPHIC C
   LR:PRINT "PLUS/4 & 16"
50 IF VEC=17165 THEN LO=45:HI=
   46:GRAPHIC CLR:PRINT"128"
60 SA=(PEEK(LO)+256*PEEK(HI))+
   6:ADR=SA
70 FOR J=0 TO 166:READ BYT:POK
   E ADR,BYT:ADR=ADR+1:CHK=CHK
   +BYT:NEXT
80 IF CHK<>20570 THEN PRINT "*
   ERROR* CHECK TYPING IN DATA
    STATEMENTS":END
90 FOR J=1 TO 5:READ RF,LF,HF:
   RS=SA+RF:HB=INT(RS/256):LB=
   RS-(256*HB)
100 CHK=CHK+RF+LF+HF:POKE SA+L
    F,LB:POKE SA+HF,HB:NEXT
```

```
110 IF CHK<>22054 THEN PRINT "
 *ERROR* RELOAD PROGRAM AND
 {SPACE}CHECK FINAL LINE":EN
 D
120 POKE SA+149,PEEK(772):POKE
 SA+150,PEEK(773)
130 IF VEC=17165 THEN POKE SA+
 14,22:POKE SA+18,23:POKESA+
 29,224:POKESA+139,224
140 PRINT CHR$(147);CHR$(17);"
 PROOFREADER ACTIVE":SYS SA
150 POKE HI,PEEK(HI)+1:POKE (P
 EEK(LO)+256*PEEK(HI))-1,0:N
 EW
160 DATA 120,169,73,141,4,3,16
 9,3,141,5,3
170 DATA 88,96,165,20,133,167,
 165,21,133,168,169
180 DATA 0,141,0,255,162,31,18
 1,199,157,227,3
190 DATA 202,16,248,169,19,32,
 210,255,169,18,32
200 DATA 210,255,160,0,132,180
 ,132,176,136,230,180
210 DATA 200,185,0,2,240,46,20
 1,34,208,8,72
220 DATA 165,176,73,255,133,17
 6,104,72,201,32,208
230 DATA 7,165,176,208,3,104,2
 08,226,104,166,180
240 DATA 24,165,167,121,0,2,13
 3,167,165,168,105
250 DATA 0,133,168,202,208,239
 ,240,202,165,167,69
260 DATA 168,72,41,15,168,185,
 211,3,32,210,255
270 DATA 104,74,74,74,74,168,1
 85,211,3,32,210
280 DATA 255,162,31,189,227,3,
 149,199,202,16,248
290 DATA 169,146,32,210,255,76
 ,86,137,65,66,67
300 DATA 68,69,70,71,72,74,75,
 77,80,81,82,83,88
310 DATA 13,2,7,167,31,32,151,
 116,117,151,128,129,167,136
 ,137
```

Program 3: IBM Proofreader

By Charles Brannon

```
10 'Automatic Proofreader Ver
 sion 3.0 (Lines 205,206 ad
 ded/190 deleted/470,490 ch
 anged from V2.0)
100 DIM L$(500),LNUM(500):COL
 OR 0,7,7:KEY OFF:CLS:MAX=
 0:LNUM(0)=65536!
110 ON ERROR GOTO 120:KEY 15,
 CHR$(4)+CHR$(70):ON KEY(1
 5) GOSUB 640:KEY (15) ON:
 GOTO 130
120 RESUME 130
130 DEF SEG=&H40:W=PEEK(&H4A)
140 ON ERROR GOTO 650:PRINT:P
 RINT"Proofreader Ready."
150 LINE INPUT L$:Y=CSRLIN-IN
 T(LEN(L$)/W)-1:LOCATE Y,1
160 DEF SEG=0:POKE 1050,30:PO
 KE 1052,34:POKE 1054,0:PO
 KE 1055,79:POKE 1056,13:P
 OKE 1057,28:LINE INPUT L$
 :DEF SEG:IF L$="" THEN 15
 0
170 IF LEFT$(L$,1)=" " THEN L
 $=MID$(L$,2):GOTO 170
180 IF VAL(LEFT$(L$,2))=0 AND
  MID$(L$,3,1)=" " THEN L$
 =MID$(L$,4)
200 IF ASC(L$)>57 THEN 260 'n
 o line number, therefore
 command
205 BL=INSTR(L$," "):IF BL=0
 THEN BL$=L$:GOTO 206 ELSE
  BL$=LEFT$(L$,BL-1)
206 LNUM=VAL(BL$):TEXT$=MID$(
 L$,LEN(STR$(LNUM))+1)
210 IF TEXT$="" THEN GOSUB 54
 0:IF LNUM=LNUM(P) THEN GO
 SUB 560:GOTO 150 ELSE 150
220 CKSUM=0:FOR I=1 TO LEN(L$
 ):CKSUM=(CKSUM+ASC(MID$(L
 $,I))*I) AND 255:NEXT:LOC
 ATE Y,1:PRINT CHR$(65+CKS
 UM/16)+CHR$(65+(CKSUM AND
  15))+" "+L$
230 GOSUB 540:IF LNUM(P)=LNUM
  THEN L$(P)=TEXT$:GOTO 15
 0 'replace line
240 GOSUB 580:GOTO 150 'inser
 t the line
260 TEXT$="":FOR I=1 TO LEN(L
 $):A=ASC(MID$(L$,I)):TEXT
 $=TEXT$+CHR$(A+32*(A>96 A
 ND A<123)):NEXT
270 DELIMITER=INSTR(TEXT$," "
 ):COMMAND$=TEXT$:ARG$="":
 IF DELIMITER THEN COMMAND
 $=LEFT$(TEXT$,DELIMITER-1
 ):ARG$=MID$(TEXT$,DELIMIT
 ER+1) ELSE DELIMITER=INST
 R(TEXT$,CHR$(34)):IF DELI
 MITER THEN COMMAND$=LEFT$
 (TEXT$,DELIMITER-1):ARG$=
 MID$(TEXT$,DELIMITER)
280 IF COMMAND$<>"LIST" THEN
 410
290 OPEN "scrn:" FOR OUTPUT A
 S #1
300 IF ARG$="" THEN FIRST=0:P
 =MAX-1:GOTO 340
310 DELIMITER=INSTR(ARG$,"-")
 :IF DELIMITER=0 THEN LNUM
 =VAL(ARG$):GOSUB 540:FIRS
 T=P:GOTO 340
320 FIRST=VAL(LEFT$(ARG$,DELI
 MITER)):LAST=VAL(MID$(ARG
 $,DELIMITER+1))
330 LNUM=FIRST:GOSUB 540:FIRS
 T=P:LNUM=LAST:GOSUB 540:I
 F P=0 THEN P=MAX-1
340 FOR X=FIRST TO P:N$=MID$(
 STR$(LNUM(X)),2)+" "
350 IF CKFLAG=0 THEN A$="":GO
 TO 370
360 CKSUM=0:A$=N$+L$(X):FOR I
 =1 TO LEN(A$):CKSUM=(CKSU
 M+ASC(MID$(A$,I))*I) AND
 255:NEXT:A$=CHR$(65+CKSUM
 /16)+CHR$(65+(CKSUM AND 1
 5))+" "
370 PRINT #1,A$+N$+L$(X)
380 IF INKEY$<>"" THEN X=P
390 NEXT :CLOSE #1:CKFLAG=0
400 GOTO 130
410 IF COMMAND$="LLIST" THEN
 OPEN "lpt1:" FOR OUTPUT A
 S #1:GOTO 300
420 IF COMMAND$="CHECK" THEN
 CKFLAG=1:GOTO 290
430 IF COMMAND$<>"SAVE" THEN
 450
440 GOSUB 600:OPEN ARG$ FOR O
 UTPUT AS #1:ARG$="":GOTO
 300
450 IF COMMAND$<>"LOAD" THEN
 490
460 GOSUB 600:OPEN ARG$ FOR I
 NPUT AS #1:MAX=0:P=0
470 WHILE NOT EOF(1):LINE INP
 UT #1,L$:BL=INSTR(L$," ")
 :BL$=LEFT$(L$,BL-1):LNUM(
 P)=VAL(BL$):L$(P)=MID$(L$
 ,LEN(STR$(VAL(BL$)))+1):P
 =P+1:WEND
480 MAX=P:CLOSE #1:GOTO 130
490 IF COMMAND$="NEW" THEN IN
 PUT "Erase program - Are
 you sure";L$:IF LEFT$(L$,
 1)="y" OR LEFT$(L$,1)="Y"
  THEN MAX=0:LNUM(0)=65536
 !:GOTO 130:ELSE 130
500 IF COMMAND$="BASIC" THEN
 COLOR 7,0,0:ON ERROR GOTO
 0:CLS:END
510 IF COMMAND$<>"FILES" THEN
  520
515 IF ARG$="" THEN ARG$="A:"
  ELSE SEL=1:GOSUB 600
517 FILES ARG$:GOTO 130
520 PRINT"Syntax error":GOTO
 130
540 P=0:WHILE LNUM>LNUM(P) AN
 D P<MAX:P=P+1:WEND:RETURN
560 MAX=MAX-1:FOR X=P TO MAX:
 LNUM(X)=LNUM(X+1):L$(X)=L
 $(X+1):NEXT:RETURN
580 MAX=MAX+1:FOR X=MAX TO P+
 1 STEP -1:LNUM(X)=LNUM(X-
 1):L$(X)=L$(X-1):NEXT:L$(
 P)=TEXT$:LNUM(P)=LNUM:RET
 URN
600 IF LEFT$(ARG$,1)<>CHR$(34
 ) THEN 520 ELSE ARG$=MID$
 (ARG$,2)
610 IF RIGHT$(ARG$,1)=CHR$(34
 ) THEN ARG$=LEFT$(ARG$,LE
 N(ARG$)-1)
620 IF SEL=0 AND INSTR(ARG$,"
 .")=0 THEN ARG$=ARG$+".BA
 S"
630 SEL=0:RETURN
640 CLOSE #1:CKFLAG=0:PRINT"S
 topped.":RETURN 150
650 PRINT "Error #";ERR:RESUM
 E 150
```

Program 4: Apple Proofreader

By Tim Victor, Editorial Programmer

```
10 C = 0: FOR I = 768 TO 768
 + 68: READ A:C = C + A: PO
 KE I,A: NEXT
20 IF C < > 7258 THEN PRINT "
 ERROR IN PROOFREADER DATA
 STATEMENTS": END
30 IF PEEK (190 * 256) < > 76
  THEN POKE 56,0: POKE 57,3
 : CALL 1002: GOTO 50
40 PRINT CHR$ (4);"IN#A$300"
50 POKE 34,0: HOME : POKE 34,
 1: VTAB 2: PRINT "PROOFREA
 DER INSTALLED"
60 NEW
100 DATA 216,32,27,253,201,14
 1
110 DATA 208,60,138,72,169,0
120 DATA 72,189,255,1,201,160
130 DATA 240,8,104,10,125,255
140 DATA 1,105,0,72,202,208
150 DATA 238,104,170,41,15,9
160 DATA 48,201,58,144,2,233
170 DATA 57,141,1,4,138,74
180 DATA 74,74,74,41,15,9
190 DATA 48,201,58,144,2,233
200 DATA 57,141,0,4,104,170
210 DATA 169,141,96
```

©

MLX Machine Language Entry Program For Commodore 64 And 128

Ottis Cowper

"MLX" is a labor-saving utility that allows almost fail-safe entry of machine language programs. Included are versions for the Commodore 64 and 128.

Type in and save some copies of whichever version of MLX is appropriate for your computer (you'll want to use it to enter future ML programs from COMPUTE!). Program 1 is for the Commodore 64, and Program 2 is for the 128 (128 MLX can also be used to enter Commodore 64 ML programs for use in 64 mode). When you're ready to enter an ML program, load and run MLX. It asks you for a starting address and an ending address. These addresses appear in the article accompanying the MLX-format program listing you're typing.

If you're unfamiliar with machine language, the addresses (and all other values you enter in MLX) may appear strange. Instead of the usual decimal numbers you're accustomed to, these numbers are in *hexadecimal*—a base 16 numbering system commonly used by ML programmers. Hexadecimal—hex for short—includes the numerals 0–9 and the letters A–F. But don't worry—even if you know nothing about ML or hex, you should have no trouble using MLX.

After you enter the starting and ending addresses, you'll be offered the option of clearing the workspace. Choose this option if you're starting to enter a new listing. If you're continuing a listing that's partially typed from a previous session, don't choose this option.

A functions menu will appear. The first option in the menu is ENTER DATA. If you're just starting to type in a program, pick this. Press the E key, and type the first number in the first line of the program listing. If you've already typed in part of a program, type the line number where you left off typing at the end of the previous session (be sure to load the partially completed program before you resume entry). In any case, make sure the address you enter corresponds to the address of a line in the listing you are entering. Otherwise, you'll be unable to enter the data correctly. If you pressed E by mistake, you can return to the command menu by pressing RETURN alone when asked for the address. (You can get back to the menu from most options by pressing RETURN with no other input.)

Entering A Listing

Once you're in Enter mode, MLX prints the address for each program line for you. You then type in all nine numbers on that line, beginning with the first two-digit number after the colon (:). Each line represents eight data bytes and a checksum. Although an MLX-format listing appears similar to the "hex dump" listings from a machine language monitor program, the extra checksum number on the end allows MLX to check your typing. (Commodore 128 users *can* enter the data from an MLX listing using the built-in monitor if the rightmost column of data is omitted, but we recommend against it. It's much easier to let MLX do the proofreading and error checking for you.)

Figure 1: 64 MLX Keypad

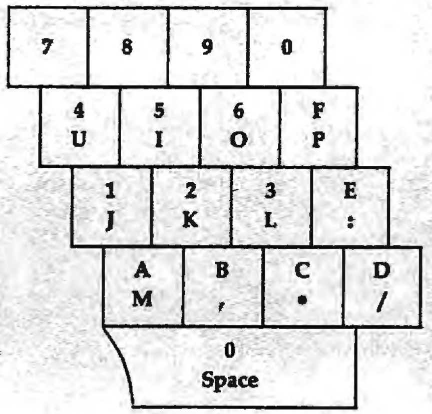
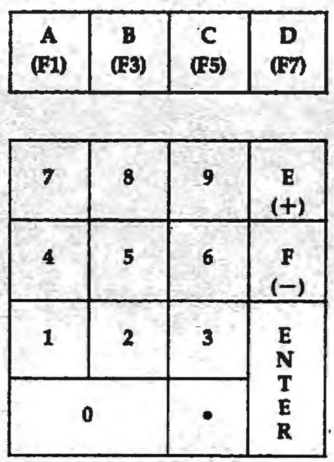


Figure 2: 128 MLX Keypad



When you enter a line, MLX recalculates the checksum from the eight bytes and the address and compares this value to the number from the ninth column. If the values match, you'll hear a bell tone, the data will be added to the workspace area, and the prompt for the next line of data will appear. But if MLX detects a typing error, you'll hear a low buzz and see an error message. The line will then be redisplayed for editing.

Invalid Characters Banned

Only a few keys are active while you're entering data, so you may have to unlearn some habits. You *do not* type spaces between the columns; MLX automatically inserts these for you. You *do not* press RETURN after typing the last number in a line; MLX automatically enters and checks the line after you type the last digit.

Only the numerals 0–9 and the letters A–F can be typed in. If you press any other key (with some exceptions noted below), you'll hear a warning buzz. To simplify typing, 128 MLX redefines the function keys and + and − keys on the numeric keypad so that you can enter data one-handed. In either case, the keypad is active only while entering data. Addresses must be entered with the normal letter and number keys. The figures below show the keypad configurations for each version.

MLX checks for transposed characters. If you're supposed to type in A0 and instead enter 0A, MLX will catch your mistake. There is one error that can slip past MLX: Because of the checksum formula used, MLX won't notice if you accidentally type FF in place of 00, and vice versa. And there's a very slim chance that you could garble a line and still end up with a combination of characters that adds up to the proper checksum. However, these mistakes should not occur if you take reasonable care while entering data.

Editing Features

To correct typing mistakes before finishing a line, use the INST/DEL key to delete the character to the left of the cursor. (The cursor-left key also deletes.) If you mess up a line really badly, press CLR/HOME to start the line over. The RETURN key is also active, but only before any data is typed on a line. Pressing RETURN at this point returns you to the command menu. After you type a character of data, MLX disables RETURN until the cursor returns to the start of a line. Remember, you can press CLR/HOME to quickly get to a line number prompt.

More editing features are available when correcting lines in which MLX has detected an error. To make corrections in a line that MLX has redisplayed for editing, compare the line on the screen with the one printed in the listing, then move the cursor to the mistake and type the correct key. The cursor left and right keys provide the normal cursor controls. (The INST/DEL key now works as an alternative cursor-left key.) You cannot move left beyond the first character in the line. If you try to move beyond the rightmost character, you'll reenter the line. During editing, RETURN is active; pressing it tells MLX to recheck the line. You can press the CLR/HOME key to clear the entire line if you want to start from scratch, or if you want to get to a line number prompt to use RETURN to get back to the menu.

Display Data

The second menu choice, DISPLAY DATA, examines memory and shows the contents in the same format as the program listing (including the checksum). When you press D, MLX asks you for a starting address. Be sure that the starting address you give corresponds to a line number in the listing. Otherwise, the checksum display will be meaningless. MLX displays program lines until it reaches the end of the program, at which point the menu is redisplayed. You can pause the display by pressing the space bar. (MLX finishes printing the current line before halting.) Press space again to restart the display. To break out of the display and get back to the menu before the ending address is reached, press RETURN.

Other Menu Options

Two more menu selections let you save programs and load them back into the computer. These are SAVE FILE and LOAD FILE; their operation is quite straightforward. When you press S or L, MLX asks you for the filename. You'll then be asked to press either D or T to select disk or tape.

You'll notice the disk drive starting and stopping several times during a load or save (save only for the 128 version). Don't panic; this is normal behavior. MLX opens and reads from or writes to the file instead of using the usual LOAD and SAVE commands (128 MLX makes use of BLOAD). Disk users should also note that the drive prefix 0: is automatically added to the filename (line 750 in 64 MLX), so this should *not* be included when entering the name. This also precludes the use of @ for Save-with-Replace, so remember to give each version you save a different name. The 128 version makes up for this by giving you the option of scratching the existing file if you want to reuse a filename.

Remember that MLX saves the entire workspace area from the starting address to the ending address, so the save or load may take longer than you might expect if you've entered only a small amount of data from a long listing. When saving a partially completed listing, make sure to note the address where you stopped typing so you'll know where to resume entry when you reload.

MLX reports the standard disk or tape error messages if any problems are detected during the save or load. (Tape users should bear in mind that Commodore computers are never able to detect errors during a save to tape.) MLX also has three special load error messages: INCORRECT STARTING ADDRESS, which means the file you're trying to load does not have the starting address you specified when you ran MLX; LOAD ENDED AT *address*, which means the file you're trying to load ends before the ending address you specified when you started MLX; and TRUNCATED AT ENDING ADDRESS, which means the file you're trying to load extends beyond the ending address you specified when you started MLX. If you see one of these messages and feel certain that you've loaded the right file, exit and rerun MLX, being careful to enter the correct starting and ending addresses.

The 128 version also has a CATALOG DISK option so you can view the contents of the disk directory before saving or loading.

The QUIT menu option has the obvious effect—it stops MLX and enters BASIC. The RUN/STOP key is disabled, so the Q option lets you exit the program without turning off the computer. (Of course, RUN/STOP-RESTORE also gets you out.) You'll be asked for verification; press Y to exit to BASIC, or any other key to return to the menu. After quitting, you can type RUN again and reenter MLX without losing your data, as long as you don't use the clear workspace option.

The Finished Product

When you've finished typing all the data for an ML program and saved your work, you're ready to see the results. The instructions for loading and using the finished product vary from program to program. Some ML programs are designed to be loaded and run like BASIC programs, so all you need to type is LOAD *"filename"*,8 for disk (DLOAD *"filename"* on the 128) or LOAD *"filename"* for tape, and then RUN. Such programs will usually have a starting address of 0801 for the 64 or 1C01 for the 128. Other programs must be reloaded to specific addresses with a command such as LOAD *"filename"*,8,1 for disk (BLOAD *"filename"* on the 128) or LOAD *"filename"*,1,1 for tape, then started with a SYS to a particular memory address. On the Commodore 64, the most common starting address for such programs is 49152, which corresponds to MLX address C000. In either case, you should always refer to the article which accompanies the ML listing for information on loading and running the program.

An Ounce Of Prevention

By the time you finish typing in the data for a long ML program, you may have several hours invested in the project. Don't take chances—use our "Automatic Proofreader" to type the new MLX, and then test your copy *thoroughly* before first using it to enter any significant amount of data. Make sure all the menu options work as they should. Enter fragments of the program starting at several different addresses, then use the Display option to verify that the data has been entered correctly. And be sure to test the Save and Load options several times to insure that you can recall your work from disk or tape. Don't let a simple typing error in the new MLX cost you several nights of hard work.

Program 1: MLX For Commodore 64

```
SS 10 REM VERSION 1.1: LINES 8
      30,950 MODIFIED, LINES 4
      85-487 ADDED
EK 100 POKE 56,50:CLR:DIM IN$,
       I,J,A,B,A$,B$,A(7),N$
DM 110 C4=48:C6=16:C7=7:Z2=2:Z
       4=254:Z5=255:Z6=256:Z7=
       127
CJ 120 FA=PEEK(45)+Z6*PEEK(46)
       :BS=PEEK(55)+Z6*PEEK(56
       ):H$="0123456789ABCDEF"
SB 130 R$=CHR$(13):L$="{LEFT}"
       :S$=" ":D$=CHR$(20):Z$=
       CHR$(0):T$="{13 RIGHT}"
CQ 140 SD=54272:FOR I=SD TO SD
       +23:POKE I,0:NEXT:POKE
       {SPACE}SD+24,15:POKE 78
       8,52
FC 150 PRINT"{CLR}"CHR$(142)CH
       R$(8):POKE 53280,15:POK
       E 53281,15
EJ 160 PRINT T$" {RED}{RVS}
       {2 SPACES}{8 @}
       {2 SPACES}"SPC(28)"
       {2 SPACES}{OFF}{BLU} ML
       X II {RED}{RVS}
       {2 SPACES}"SPC(28)"
       {12 SPACES}{BLU}"
FR 170 PRINT"{3 DOWN}
       {3 SPACES}COMPUTE!'S MA
```

```
         CHINE LANGUAGE EDITOR
         {3 DOWN}"
JB 180 PRINT"{BLK}STARTING ADD
         RESS{4}";:GOSUB300:SA=A
         D:GOSUB1040:IF F THEN18
         0
GF 190 PRINT"{BLK}{2 SPACES}EN
         DING ADDRESS{4}";:GOSUB
         300:EA=AD:GOSUB1030:IF
         {SPACE}F THEN190
KR 200 INPUT"{3 DOWN}{BLK}CLEA
         R WORKSPACE [Y/N]{4}";A
         $:IF LEFT$(A$,1)<>"Y"TH
         EN220
PG 210 PRINT"{2 DOWN}{BLU}WORK
         ING...";:FORI=BS TO BS+
         EA-SA+7:POKE I,0:NEXT:P
         RINT"DONE"
DR 220 PRINTTAB(10)"{2 DOWN}
         {BLK}{RVS} MLX COMMAND
         {SPACE}MENU {DOWN}{4}":
         PRINT T$"{RVS}E{OFF}NTE
         R DATA"
BD 230 PRINT T$"{RVS}D{OFF}ISP
         LAY DATA":PRINT T$"
         {RVS}L{OFF}OAD FILE"
JS 240 PRINT T$"{RVS}S{OFF}AVE
          FILE":PRINT T$"{RVS}Q
         {OFF}UIT{2 DOWN}{BLK}"
JH 250 GET A$:IF A$=N$ THEN250
HK 260 A=0:FOR I=1 TO 5:IF A$=
         MID$("EDLSQ",I,1)THEN A
         =I:I=5
FD 270 NEXT:ON A GOTO420,610,6
         90,700,280:GOSUB1060:GO
         TO250
EJ 280 PRINT"{RVS} QUIT ":INPU
         T"{DOWN}{4}ARE YOU SURE
          [Y/N]";A$:IF LEFT$(A$,
         1)<>"Y"THEN220
EM 290 POKE SD+24,0:END
JX 300 IN$=N$:AD=0:INPUTIN$:IF
         LEN(IN$)<>4THENRETURN
KF 310 B$=IN$:GOSUB320:AD=A:B$
         =MID$(IN$,3):GOSUB320:A
         D=AD*256+A:RETURN
PP 320 A=0:FOR J=1 TO 2:A$=MID
         $(B$,J,1):B=ASC(A$)-C4+
         (A$>"@")*C7:A=A*C6+B
JA 330 IF B<0 OR B>15 THEN AD=
         0:A=-1:J=2
GX 340 NEXT:RETURN
CH 350 B=INT(A/C6):PRINT MID$(
         H$,B+1,1);:B=A-B*C6:PRI
         NT MID$(H$,B+1,1);:RETU
         RN
RR 360 A=INT(AD/Z6):GOSUB350:A
         =AD-A*Z6:GOSUB350:PRINT
         ":";
BE 370 CK=INT(AD/Z6):CK=AD-Z4*
         CK+Z5*(CK>Z7):GOTO390
PX 380 CK=CK*Z2+Z5*(CK>Z7)+A
JC 390 CK=CK+Z5*(CK>Z5):RETURN
QS 400 PRINT"{DOWN}STARTING AT
         {4}";:GOSUB300:IF IN$<>
         N$ THEN GOSUB1030:IF F
         {SPACE}THEN400
EX 410 RETURN
HD 420 PRINT"{RVS} ENTER DATA
         {SPACE}":GOSUB400:IF IN
         $=N$ THEN220
JK 430 OPEN3,3:PRINT
SK 440 POKE198,0:GOSUB360:IF F
          THEN PRINT IN$:PRINT"
         {UP}{5 RIGHT}";
GC 450 FOR I=0 TO 24 STEP 3:B$
         =S$:FOR J=1 TO 2:IF F T
         HEN B$=MID$(IN$,I+J,1)
HA 460 PRINT"{RVS}"B$L$;:IF I<
         24THEN PRINT"{OFF}";
HD 470 GET A$:IF A$=N$ THEN470
FK 480 IF(A$>"/"ANDA$<":")OR(A
         $>"@"ANDA$<"G")THEN540
GS 485 A=-(A$="M")-2*(A$=",")-
         3*(A$=".")-4*(A$="/")-5
         *(A$="J")-6*(A$="K")
FX 486 A=A-7*(A$="L")-8*(A$=":
         ")-9*(A$="U")-10*(A$="I
         ")-11*(A$="O")-12*(A$="
         P")
CM 487 A=A-13*(A$=S$):IF A THE
         N A$=MID$("ABCD123E456F
         0",A,1):GOTO 540
MP 490 IF A$=R$ AND((I=0)AND(J
         =1)OR F)THEN PRINT B$;:
         J=2:NEXT:I=24:GOTO550
KC 500 IF A$="{HOME}" THEN PRI
         NT B$:J=2:NEXT:I=24:NEX
         T:F=0:GOTO440
MX 510 IF(A$="{RIGHT}")ANDF TH
         ENPRINT B$L$;:GOTO540
GK 520 IF A$<>L$ AND A$<>D$ OR
         ((I=0)AND(J=1))THEN GOS
         UB1060:GOTO470
HG 530 A$=L$+S$+L$:PRINT B$L$;
         :J=2-J:IF J THEN PRINT
         {SPACE}L$;:I=I-3
QS 540 PRINT A$;:NEXT J:PRINT
         {SPACE}S$;
PM 550 NEXT I:PRINT:PRINT"{UP}
         {5 RIGHT}";:INPUT#3,IN$
         :IF IN$=N$ THEN CLOSE3:
         GOTO220
QC 560 FOR I=1 TO 25 STEP3:B$=
         MID$(IN$,I):GOSUB320:IF
          I<25 THEN GOSUB380:A(I
         /3)=A
PK 570 NEXT:IF A<>CK THEN GOSU
         B1060:PRINT"{BLK}{RVS}
         {SPACE}ERROR: REENTER L
         INE {4}":F=1:GOTO440
HJ 580 GOSUB1080:B=BS+AD-SA:FO
         R I=0 TO 7:POKE B+I,A(I
         ):NEXT
QQ 590 AD=AD+8:IF AD>EA THEN C
         LOSE3:PRINT"{DOWN}{BLU}
         ** END OF ENTRY **{BLK}
         {2 DOWN}":GOTO700
GQ 600 F=0:GOTO440
QA 610 PRINT"{CLR}{DOWN}{RVS}
         {SPACE}DISPLAY DATA ":G
         OSUB400:IF IN$=N$ THEN2
         20
RJ 620 PRINT"{DOWN}{BLU}PRESS:
          {RVS}SPACE{OFF} TO PAU
         SE, {RVS}RETURN{OFF} TO
          BREAK{4}{DOWN}"
KS 630 GOSUB360:B=BS+AD-SA:FOR
         I=BTO B+7:A=PEEK(I):GOS
         UB350:GOSUB380:PRINT S$
         ;
CC 640 NEXT:PRINT"{RVS}";:A=CK
         :GOSUB350:PRINT
KH 650 F=1:AD=AD+8:IF AD>EA TH
         ENPRINT"{DOWN}{BLU}** E
         ND OF DATA **":GOTO220
KC 660 GET A$:IF A$=R$ THEN GO
         SUB1080:GOTO220
EQ 670 IF A$=S$ THEN F=F+1:GOS
         UB1080
AD 680 ONFGOTO630,660,630
CM 690 PRINT"{DOWN}{RVS} LOAD
         {SPACE}DATA ":OP=1:GOTO
         710
PC 700 PRINT"{DOWN}{RVS} SAVE
         {SPACE}FILE ":OP=0
RX 710 IN$=N$:INPUT"{DOWN}FILE
         NAME{4}";IN$:IF IN$=N$
         {SPACE}THEN220
PR 720 F=0:PRINT"{DOWN}{BLK}
         {RVS}T{OFF}APE OR {RVS}
         D{OFF}ISK: {4}";
FP 730 GET A$:IF A$="T"THEN PR
         INT"T{DOWN}":GOTO880
HQ 740 IF A$<>"D"THEN730
HH 750 PRINT"D{DOWN}":OPEN15,8
         ,15,"I0:":B=EA-SA:IN$="
         0:"+IN$:IF OP THEN810
SQ 760 OPEN 1,8,8,IN$+",P,W":G
         OSUB860:IF A THEN220
FJ 770 AH=INT(SA/256):AL=SA-(A
         H*256):PRINT#1,CHR$(AL)
         ;CHR$(AH);
PE 780 FOR I=0 TO B:PRINT#1,CH
         R$(PEEK(BS+I));:IF ST T
         HEN800
FC 790 NEXT:CLOSE1:CLOSE15:GOT
         O940
GS 800 GOSUB1060:PRINT"{DOWN}
         {BLK}ERROR DURING SAVE:
         {4}":GOSUB860:GOTO220
MA 810 OPEN 1,8,8,IN$+",P,R":G
         OSUB860:IF A THEN220
GE 820 GET#1,A$,B$:AD=ASC(A$+Z
         $)+256*ASC(B$+Z$):IF AD
         <>SA THEN F=1:GOTO850
RX 830 FOR I=0 TO B:GET#1,A$:P
         OKE BS+I,ASC(A$+Z$):IF(
         I<>B)AND ST THEN F=2:AD
         =I:I=B
FA 840 NEXT:IF ST<>64 THEN F=3
FQ 850 CLOSE1:CLOSE15:ON ABS(F
         >0)+1 GOTO960,970
SA 860 INPUT#15,A,A$:IF A THEN
          CLOSE1:CLOSE15:GOSUB10
         60:PRINT"{RVS}ERROR: "A
         $
GQ 870 RETURN
EJ 880 POKE183,PEEK(FA+2):POKE
         187,PEEK(FA+3):POKE188,
         PEEK(FA+4):IFOP=0THEN92
         0
HJ 890 SYS 63466:IF(PEEK(783)A
         ND1)THEN GOSUB1060:PRIN
         T"{DOWN}{RVS} FILE NOT
         {SPACE}FOUND ":GOTO690
CS 900 AD=PEEK(829)+256*PEEK(8
         30):IF AD<>SA THEN F=1:
         GOTO970
SC 910 A=PEEK(831)+256*PEEK(83
         2)-1:F=F-2*(A<EA)-3*(A>
         EA):AD=A-AD:GOTO930
KM 920 A=SA:B=EA+1:GOSUB1010:P
         OKE780,3:SYS 63338
JF 930 A=BS:B=BS+(EA-SA)+1:GOS
         UB1010:ON OP GOTO950:SY
         S 63591
AE 940 GOSUB1080:PRINT"{BLU}**
          SAVE COMPLETED **":GOT
         O220
XP 950 POKE147,0:SYS 63562:IF
         {SPACE}ST>0 THEN970
FR 960 GOSUB1080:PRINT"{BLU}**
          LOAD COMPLETED **":GOT
         O220
DP 970 GOSUB1060:PRINT"{BLK}
         {RVS}ERROR DURING LOAD:
         {DOWN}{4}":ON F GOSUB98
         0,990,1000:GOTO220
PP 980 PRINT"INCORRECT STARTIN
         G ADDRESS (";:GOSUB360:
         PRINT")":RETURN
GR 990 PRINT"LOAD ENDED AT ";:
         AD=SA+AD:GOSUB360:PRINT
         D$:RETURN
FD 1000 PRINT"TRUNCATED AT END
          ING ADDRESS":RETURN
RX 1010 AH=INT(A/256):AL=A-(AH
          *256):POKE193,AL:POKE1
          94,AH
FF 1020 AH=INT(B/256):AL=B-(AH
          *256):POKE174,AL:POKE1
          75,AH:RETURN
```

```
FX 1030 IF AD<SA OR AD>EA THEN
        1050
HA 1040 IF(AD>511 AND AD<40960
        )OR(AD>49151 AND AD<53
        248)THEN GOSUB1080:F=0
        :RETURN
HC 1050 GOSUB1060:PRINT"{RVS}
        {SPACE}INVALID ADDRESS
         {DOWN}{BLK}":F=1:RETU
        RN
AR 1060 POKE SD+5,31:POKE SD+6
        ,208:POKE SD,240:POKE
        {SPACE}SD+1,4:POKE SD+
        4,33
DX 1070 FOR S=1 TO 100:NEXT:GO
        TO1090
PF 1080 POKE SD+5,8:POKE SD+6,
        240:POKE SD,0:POKE SD+
        1,90:POKE SD+4,17
AC 1090 FOR S=1 TO 100:NEXT:PO
        KE SD+4,0:POKE SD,0:PO
        KE SD+1,0:RETURN
```

Program 2: MLX For Commodore 128

```
AE 100 TRAP 960:POKE 4627,128:
       DIM NL$,A(7)
XP 110 Z2=2:Z4=254:Z5=255:Z6=2
       56:Z7=127:BS=256*PEEK(4
       627):EA=65280
FB 120 BE$=CHR$(7):RT$=CHR$(13
       ):DL$=CHR$(20):SP$=CHR$
       (32):LF$=CHR$(157)
KE 130 DEF FNHB(A)=INT(A/256):
       DEF FNLB(A)=A-FNHB(A)*2
       56:DEF FNAD(A)=PEEK(A)+
       256*PEEK(A+1)
JB 140 KEY 1,"A":KEY 3,"B":KEY
        5,"C":KEY 7,"D":VOL 15
       :IF RGR(0)=5 THEN FAST
FJ 150 PRINT"{CLR}"CHR$(142);C
       HR$(8):COLOR 0,15:COLOR
        4,15:COLOR 6,15
GQ 160 PRINT TAB(12)"{RED}
       {RVS}{2 SPACES}{9 @}
       {2 SPACES}"RT$;TAB(12)"
       {RVS}{2 SPACES}{OFF}
       {BLU} 128 MLX {RED}
       {RVS}{2 SPACES}"RT$;TAB
       (12)"{RVS}{13 SPACES}
       {BLU}"
FE 170 PRINT"{2 DOWN}
       {3 SPACES}COMPUTE!'S MA
       CHINE LANGUAGE EDITOR
       {2 DOWN}"
DK 180 PRINT"{BLK}STARTING ADD
       RESS{4}";:GOSUB 260:IF
       {SPACE}AD THEN SA=AD:EL
       SE 180
FH 190 PRINT"{BLK}{2 SPACES}EN
       DING ADDRESS{4}";:GOSUB
        260:IF AD THEN EA=AD:E
       LSE 190
MF 200 PRINT"{DOWN}{BLK}CLEAR
       {SPACE}WORKSPACE [Y/N]?
       {4}":GETKEY A$:IF A$<>"
       Y" THEN 220
QH 210 PRINT"{DOWN}{BLU}WORKIN
       G...";:BANK 0:FOR A=BS
       {SPACE}TO BS+(EA-SA)+7:
       POKE A,0:NEXT A:PRINT"D
       ONE"
DC 220 PRINT TAB(10)"{DOWN}
       {BLK}{RVS} MLX COMMAND
       {SPACE}MENU {4}{DOWN}":
       PRINT TAB(13)"{RVS}E
       {OFF}NTER DATA"RT$;TAB(
       13)"{RVS}D{OFF}ISPLAY D
       ATA"RT$;TAB(13)"{RVS}L
       {OFF}OAD FILE"
HB 230 PRINT TAB(13)"{RVS}S
       {OFF}AVE FILE"RT$;TAB(1
       3)"{RVS}C{OFF}ATALOG DI
       SK"RT$;TAB(13)"{RVS}Q
       {OFF}UIT{DOWN}{BLK}"
AP 240 GETKEY A$:A=INSTR("EDLS
       CQ",A$):ON A GOTO 340,5
       50,640,650,930,940:GOSU
       B 950:GOTO 240
SX 250 PRINT"STARTING AT";:GOS
       UB 260:IF(AD<>0)OR(A$=N
       L$)THEN RETURN:ELSE 250
BG 260 A$=NL$:INPUT A$:IF LEN(
       A$)=4 THEN AD=DEC(A$)
PP 270 IF AD=0 THEN BEGIN:IF A
       $<>NL$ THEN 300:ELSE RE
       TURN:BEND
MA 280 IF AD<SA OR AD>EA THEN
       {SPACE}300
PM 290 IF AD>511 AND AD<65280
       {SPACE}THEN PRINT BE$;:
       RETURN
SQ 300 GOSUB 950:PRINT"{RVS} I
       NVALID ADDRESS {DOWN}
       {BLK}":AD=0:RETURN
RD 310 CK=FNHB(AD):CK=AD-Z4*CK
       +Z5*(CK>Z7):GOTO 330
DD 320 CK=CK*Z2+Z5*(CK>Z7)+A
AH 330 CK=CK+Z5*(CK>Z5):RETURN
QD 340 PRINT BE$;"{RVS} ENTER
       {SPACE}DATA ":GOSUB 250
       :IF A$=NL$ THEN 220
JA 350 BANK 0:PRINT:F=0:OPEN 3
       ,3
BR 360 GOSUB 310:PRINT HEX$(AD
       )+":";:IF F THEN PRINT
       {SPACE}L$:PRINT"{UP}
       {5 RIGHT}";
QA 370 FOR I=0 TO 24 STEP 3:B$
       =SP$:FOR J=1 TO 2:IF F
       {SPACE}THEN B$=MID$(L$,
       I+J,1)
PS 380 PRINT"{RVS}"B$+LF$;:IF
       {SPACE}I<24 THEN PRINT"
       {OFF}";
RC 390 GETKEY A$:IF (A$>"/" AN
       D A$<":") OR(A$>"@" AND
        A$<"G") THEN 470
AC 400 IF A$="+" THEN A$="E":G
       OTO 470
QB 410 IF A$="-" THEN A$="F":G
       OTO 470
FB 420 IF A$=RT$ AND ((I=0) AN
       D (J=1) OR F) THEN PRIN
       T B$;:J=2:NEXT:I=24:GOT
       O 480
RD 430 IF A$="{HOME}" THEN PRI
       NT B$:J=2:NEXT:I=24:NEX
       T:F=0:GOTO 360
XB 440 IF (A$="{RIGHT}") AND F
        THEN PRINT B$+LF$;:GOT
       O 470
JP 450 IF A$<>LF$ AND A$<>DL$
       {SPACE}OR ((I=0) AND (J
       =1)) THEN GOSUB 950:GOT
       O 390
PS 460 A$=LF$+SP$+LF$:PRINT B$
       +LF$;:J=2-J:IF J THEN P
       RINT LF$;:I=I-3
GB 470 PRINT A$;:NEXT J:PRINT
       {SPACE}SP$;
HA 480 NEXT I:PRINT:PRINT"{UP}
       {5 RIGHT}";:L$="
       {27 SPACES}"
DP 490 FOR I=1 TO 25 STEP 3:GE
       T#3,A$,B$:IF A$=SP$ THE
       N I=25:NEXT:CLOSE 3:GOT
       O 220
BA 500 A$=A$+B$:A=DEC(A$):MID$
       (L$,I,2)=A$:IF I<25 THE
       N GOSUB 320:A(I/3)=A:GE
       T#3,A$
AR 510 NEXT I:IF A<>CK THEN GO
       SUB 950:PRINT:PRINT"
       {RVS} ERROR: REENTER LI
       NE ":F=1:GOTO 360
DX 520 PRINT BE$:B=BS+AD-SA:FO
       R I=0 TO 7:POKE B+I,A(I
       ):NEXT I
XB 530 F=0:AD=AD+8:IF AD<=EA T
       HEN 360
CA 540 CLOSE 3:PRINT"{DOWN}
       {BLU}** END OF ENTRY **
       {BLK}{2 DOWN}":GOTO 650
MC 550 PRINT BE$;"{CLR}{DOWN}
       {RVS} DISPLAY DATA ":GO
       SUB 250:IF A$=NL$ THEN
       {SPACE}220
JF 560 BANK 0:PRINT"{DOWN}
       {BLU}PRESS: {RVS}SPACE
       {OFF} TO PAUSE, {RVS}RE
       TURN{OFF} TO BREAK{4}
       {DOWN}"
XA 570 PRINT HEX$(AD)+":";:GOS
       UB 310:B=BS+AD-SA
DJ 580 FOR I=B TO B+7:A=PEEK(I
       ):PRINT RIGHT$(HEX$(A),
       2);SP$;:GOSUB 320:NEXT
       {SPACE}I
XB 590 PRINT"{RVS}";RIGHT$(HEX
       $(CK),2)
GR 600 F=1:AD=AD+8:IF AD>EA TH
       EN PRINT"{BLU}** END OF
        DATA **":GOTO 220
EB 610 GET A$:IF A$=RT$ THEN P
       RINT BE$:GOTO 220
QK 620 IF A$=SP$ THEN F=F+1:PR
       INT BE$;
XS 630 ON F GOTO 570,610,570
RF 640 PRINT BE$"{DOWN}{RVS} L
       OAD DATA ":OP=1:GOTO 66
       0
BP 650 PRINT BE$"{DOWN}{RVS} S
       AVE FILE ":OP=0
DM 660 F=0:F$=NL$:INPUT"FILENA
       ME{4}";F$:IF F$=NL$ THE
       N 220
RF 670 PRINT"{DOWN}{BLK}{RVS}T
       {OFF}APE OR {RVS}D{OFF}
       ISK: {4}";
SQ 680 GETKEY A$:IF A$="T" THE
       N 850:ELSE IF A$<>"D" T
       HEN 680
SP 690 PRINT"DISK{DOWN}":IF OP
        THEN 760
EH 700 DOPEN#1,(F$+",P"),W:IF
       {SPACE}DS THEN A$=D$:GO
       TO 740
JH 710 BANK 0:POKE BS-2,FNLB(S
       A):POKE BS-1,FNHB(SA):P
       RINT"SAVING ";F$:PRINT
MC 720 FOR A=BS-2 TO BS+EA-SA:
       PRINT#1,CHR$(PEEK(A));:
       IF ST THEN A$="DISK WRI
       TE ERROR":GOTO 750
GC 730 NEXT A:CLOSE 1:PRINT"
       {BLU}** SAVE COMPLETED
       {SPACE}WITHOUT ERRORS *
       *":GOTO 220
RA 740 IF DS=63 THEN BEGIN:CLO
       SE 1:INPUT"{BLK}REPLACE
        EXISTING FILE [Y/N]{4}
       ";A$:IF A$="Y" THEN SCR
       ATCH(F$):PRINT:GOTO 700
       :ELSE PRINT"{BLK}":GOTO
        660:BEND
GA 750 CLOSE 1:GOSUB 950:PRINT
       "{BLK}{RVS} ERROR DURIN
       G SAVE: {4}":PRINT A$:G
       OTO 220
FD 760 DOPEN#1,(F$+",P"):IF DS
        THEN A$=DS$:F=4:CLOSE
       {SPACE}1:GOTO 790
```

```
PX 77Ø GET#1,A$,B$:CLOSE 1:AD=
       ASC(A$)+256*ASC(B$):IF
       {SPACE}AD<>SA THEN F=1:
       GOTO 79Ø
KB 78Ø PRINT"LOADING ";F$:PRIN
       T:BLOAD(F$),BØ,P(BS):AD
       =SA+FNAD(174)-BS-1:F=-2
       *(AD<EA)-3*(AD>EA)
RQ 79Ø IF F THEN 8ØØ:ELSE PRIN
       T"{BLU}** LOAD COMPLETE
       D WITHOUT ERRORS **":GO
       TO 22Ø
ER 8ØØ GOSUB 95Ø:PRINT"{BLK}
       {RVS} ERROR DURING LOAD
       : {4}":ON F GOSUB 81Ø,8
       2Ø,83Ø,84Ø:GOTO22Ø
QJ 81Ø PRINT"INCORRECT STARTIN
       G ADDRESS (";HEX$(AD);"
       )":RETURN
DP 82Ø PRINT"LOAD ENDED AT ";H
       EX$(AD):RETURN
EB 83Ø PRINT"TRUNCATED AT ENDI
       NG ADDRESS ("HEX$(EA)")
       ":RETURN
FP 84Ø PRINT"DISK ERROR ";A$:R
       ETURN
KS 85Ø PRINT"TAPE":AD=POINTER(
       F$):BANK 1:A=PEEK(AD):A
       L=PEEK(AD+1):AH=PEEK(AD
       +2)
XX 86Ø BANK 15:SYS DEC("FF68")
       ,Ø,1:SYS DEC("FFBA"),1,
       1,Ø:SYS DEC("FFBD"),A,A
       L,AH:SYS DEC("FF9Ø"),12
       8:IF OP THEN 89Ø
FG 87Ø PRINT:A=SA:B=EA+1:GOSUB
        92Ø:SYS DEC("E919"),3:
       PRINT"SAVING ";F$
AB 88Ø A=BS:B=BS+(EA-SA)+1:GOS
       UB 92Ø:SYS DEC("EA18"):
       PRINT"{DOWN}{BLU}** TAP
       E SAVE COMPLETED **":GO
       TO 22Ø
CP 89Ø SYS DEC("E99A"):PRINT:I
       F PEEK(2816)=5 THEN GOS
       UB 95Ø:PRINT"{DOWN}
       {BLK}{RVS} FILE NOT FOU
       ND ":GOTO 22Ø
GQ 9ØØ PRINT"LOADING ...{DOWN}
       ":AD=FNAD(2817):IF AD<>
       SA THEN F=1:GOTO 8ØØ:EL
       SE AD=FNAD(2819)-1:F=-2
       *(AD<EA)-3*(AD>EA)
JD 91Ø A=BS:B=BS+(EA-SA)+1:GOS
       UB 92Ø:SYS DEC("E9FB"):
       IF ST>Ø THEN 8ØØ:ELSE 7
       9Ø
XB 92Ø POKE193,FNLB(A):POKE194
       ,FNHB(A):POKE 174,FNLB(
       B):POKE 175,FNHB(B):RET
       URN
CP 93Ø CATALOG:PRINT"{DOWN}
       {BLU}** PRESS ANY KEY F
       OR MENU **":GETKEY A$:G
       OTO 22Ø
MM 94Ø PRINT BE$"{RVS} QUIT
       {4}";RT$;"ARE YOU SURE
       {SPACE}[Y/N]?":GETKEY A
       $:IF A$<>"Y" THEN 22Ø:E
       LSE PRINT"{CLR}":BANK 1
       5:END
JE 95Ø SOUND 1,5ØØ,1Ø:RETURN
AF 96Ø IF ER=14 AND EL=26Ø THE
       N RESUME 3ØØ
MK 97Ø IF ER=14 AND EL=5ØØ THE
       N RESUME NEXT
KJ 98Ø IF ER=4 AND EL=78Ø THEN
        F=4:A$=DS$:RESUME 8ØØ
DQ 99Ø IF ER=3Ø THEN RESUME:EL
       SE PRINT ERR$(ER);" ERR
       OR IN LINE";EL
```

©

MLX Machine Language Entry Program For Apple

Tim Victor

To make it easier to enter machine language programs into your computer without typos, COMPUTE! is introducing its "MLX" entry program for the Apple II series. It's our best MLX yet. It runs on the II, II+, IIe, and IIc, and with either DOS 3.3 or ProDOS.

A machine language (ML) program is usually listed as a long series of numbers. It's hard to keep your place and even harder to avoid making mistakes as you type in the listing, since an incorrect line looks almost identical to a correct one. To make error-free entry easier, COMPUTE! generally lists ML programs for Commodore and Atari computers in a format designed to be typed in with a utility called "MLX." The MLX program uses a checksum system to catch typing errors almost as soon as they happen.

Apple MLX checks your typing on a line-by-line basis. It won't let you enter invalid characters or let you continue if there's a mistake in a line. It won't even let you enter a line or digit out of sequence. Best of all, you don't have to know anything about machine language to enter ML programs with MLX. Apple MLX makes typing ML programs almost foolproof.

Using Apple MLX

Type in and save some copies of Apple MLX on disk (you'll want to use MLX to enter future ML programs in COMPUTE!). It doesn't matter whether you type it in on a disk formatted for DOS 3.3 or ProDOS. Programs entered with Apple MLX, however, must be saved to a disk formatted with the same operating system as Apple MLX itself.

If you have an Apple IIe or IIc, make sure that the key marked CAPS LOCK is in the down position. Type RUN. You'll be asked for the starting and ending addresses of the ML program. These values vary for each program, so they're given at the beginning of the ML program listing and in the program's accompanying article. Find them and type them in.

Invalid Characters Banned

Apple MLX is fairly flexible about how you type in the numbers. You can put extra spaces between numbers or leave the spaces out entirely, compressing a line into 18 keypresses. Be careful not to put a space between two digits in the middle of a number. Apple MLX will read two single-digit numbers instead of one two-digit number (F 6 means F and 6, not F6).

You can't enter an invalid character with Apple MLX. Only the numerals 0–9 and the letters A–F can be typed in. If you press any other key (with some exceptions noted below), nothing happens. This safeguards against entering extraneous characters. Even better, Apple MLX checks for transposed characters. If you're supposed to type in A0 and instead enter 0A, Apple MLX will catch your mistake.

The next thing you'll see is a menu asking you to select a function. The first is (E)NTER DATA. If you're just starting to type in a program, pick this. Press the E key, and the program asks for the address where you want to begin entering data. Type the first number in the first line of the program listing if you're just starting, or the line number where you left off if you've already typed in part of a program. Hit the RETURN key and begin entering the data.

Once you're in Enter mode, Apple MLX prints the address for each program line for you. You then type in all nine numbers on that line, beginning with the first two-digit number after the colon (:). Each line represents eight bytes and a checksum. When you enter a line and hit RETURN, Apple MLX recalculates the checksum from the eight bytes and the address. If you enter more or less than nine numbers, or the checksum doesn't exactly match, Apple MLX erases the line you just entered and prompts you again for the same line.

Apple MLX also checks to make sure you're typing in the right line. The address (the number to the left of the colon) is part of the checksum recalculation. If you accidentally skip a line and try to enter incorrect values, Apple MLX won't let you continue. Just make sure you enter the correct starting address; if you don't, you won't be able to enter any of the following lines. Apple MLX will stop you.

Editing Features

Apple MLX also includes some editing features. The left- and right-arrow keys allow you to back up and go forward on the line that you are entering, so you can retype data. Pressing the CONTROL (CTRL) and D keys at the same time (*delete*) removes the character under the cursor, shortening the line by one character. Pressing CTRL-I (*insert*) puts a space under the cursor and shifts the rest of the line to the right, making the line one character longer. If the cursor is at the right end of the line, neither CTRL-D nor CTRL-I has any effect.

When you've entered the entire listing (up to the ending address that you specified earlier), Apple MLX automatically leaves Enter mode and redisplays the functions menu. If you want to leave Enter mode before then, press the RETURN key when Apple MLX prompts you with a new line address. (For instance, you may want to leave Enter mode to enter a program listing in more than one sitting; see below.)

Display Data

The second menu choice, (D)ISPLAY DATA, examines memory and shows the contents in the same format as the program listing. You can use it to check your work or to see how far you've gotten. When you press D, Apple MLX asks you for a starting address. Type in the address of the first line you want to see and hit RETURN. Apple MLX displays program lines until you press any key or until it reaches the end of the program.

Save And Load

Two more menu selections let you save programs on disk and load them back into the computer. These are (S)AVE FILE and (L)OAD FILE. When you press S or L, Apple MLX asks you for the filename. The first time you save an ML program, the name you assign will be the program's filename on the disk. If you press L and specify a filename that doesn't exist on the disk, you'll see a disk error message.

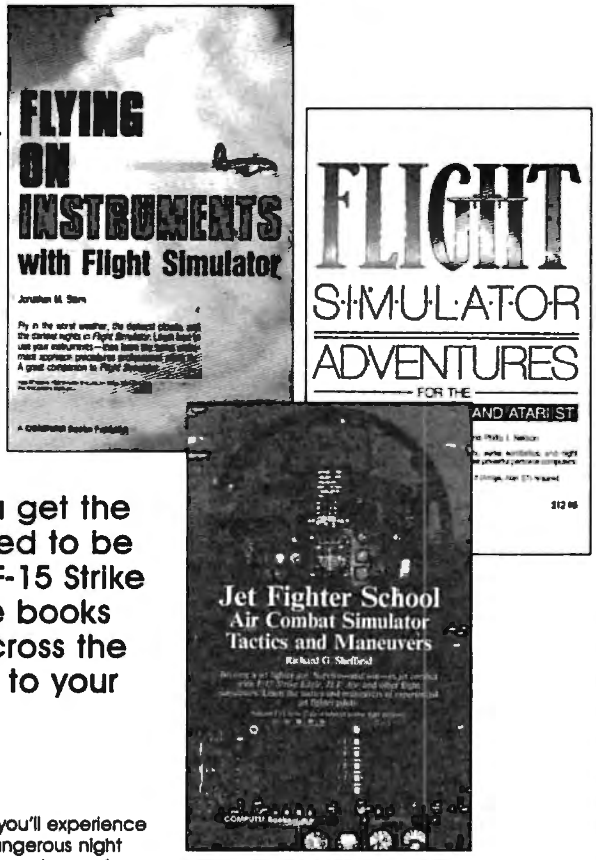
If you're not sure why a disk error has occurred, check the drive. Make sure there's a formatted disk in the drive and that it was formatted by the same operating system you're using for Apple MLX (ProDOS or DOS 3.3). If you're trying to save a file and see an error message, the disk might be full. Either save the file on another disk or quit Apple MLX (by pressing the Q key), delete an old file or two, then run Apple MLX again. Your typing should still be safe in memory.

Apple MLX: Machine Language Entry Program

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
35 100 N = 9: HOME : NORMAL : PRINT CHR$ (17);"APPLE MLX V1.1": POKE 34,2: ONERR GOTO 610
CC 110 VTAB 1: HTAB 20: PRINT "START ADDRESS";: GOSUB 530 : IF A = 0 THEN PRINT CHR$ (7): GOTO 110
8C 120 S = A
E3 130 VTAB 2: HTAB 20: PRINT "END ADDRESS ";: GOSUB 530 : IF S > = A OR A = 0 THEN PRINT CHR$ (7): GOTO 130
20 140 E = A
85 150 PRINT : PRINT "CHOOSE: (E)NTER DATA";: HTAB 22: PRINT "(D)ISPLAY DATA": HTAB 8: PRINT "(L)OAD FILE  (S)AVE FILE  (Q)UIT": PRINT
AE 160 GET A$: FOR I = 1 TO 5: IF A$ < > MID$ ("EDLSQ",I,1) THEN NEXT : GOTO 160
93 170 ON I GOTO 270,220,180,200 : POKE 34,0: END
AF 180 INPUT "FILENAME: ";A$: IF A$ < > "" THEN PRINT CHR$ (4);"BLOAD";A$;",A";S
A1 190 GOTO 150
60 200 INPUT "FILENAME: ";A$: IF A$ < > "" THEN PRINT CHR$ (4);"BSAVE";A$;",A";S;",L";E - S
92 210 GOTO 150
C2 220 GOSUB 590: IF B = 0 THEN 150
9E 230 FOR B = B TO E STEP 8:L = 4:A = B: GOSUB 580: PRINT A$;": ";:L = 2
85 240 FOR F = 0 TO 7:V(F + 1) = PEEK (B + F): NEXT : GOSUB 560:V(9) = C
F2 250 FOR F = 1 TO N:A = V(F): GOSUB 580: PRINT A$" ";: NEXT : PRINT : IF PEEK (49152) < 128 THEN NEXT
94 260 POKE 49168,0: GOTO 150
CC 270 GOSUB 590: IF B = 0 THEN 150
48 280 FOR B = B TO E STEP 8
A6 290 HTAB 1:A = B:L = 4: GOSUB 580: PRINT A$;": ";: CALL 64668:A$ = "":P = 0: GOSUB 330: IF L = 0 THEN 150
F9 300 GOSUB 470: IF F < > N THEN PRINT CHR$ (7);: GOTO 290
27 310 IF N = 9 THEN GOSUB 560: IF C < > V(9) THEN PRINT CHR$ (7);: GOTO 290
72 320 FOR F = 1 TO 8: POKE B + F - 1,V(F): NEXT : PRINT : NEXT : GOTO 150
8E 330 IF LEN (A$) = 33 THEN A$ = O$:P = O: PRINT CHR$ (7);
22 340 L = LEN (A$):O$ = A$:O = P:L$ = "": IF P > 0 THEN L$ = LEFT$ (A$,P)
E8 350 R$ = "": IF P < L - 1 THEN R$ = RIGHT$ (A$,L - P - 1)
55 360 HTAB 7: PRINT L$;: FLASH : IF P < L THEN PRINT MID$ (A$,P + 1,1);: NORMAL : PRINT R$;
78 370 PRINT " ";: NORMAL
E6 380 K = PEEK (49152): IF K < 128 THEN 380
C1 390 POKE 49168,0:K = K - 128
58 400 IF K = 13 THEN HTAB 7: PRINT A$;" ";: RETURN
A7 410 IF K = 32 OR K > 47 AND K < 58 OR K > 64 AND K < 71 THEN A$ = L$ + CHR$ (K) + R$:P = P + 1: GOTO 330
C7 420 I = FRE (0): IF K = 4 THEN A$ = L$ + R$
5F 430 IF K = 9 THEN A$ = L$ + " " + MID$ (A$,P + 1,1) + R$
0A 440 IF K = 8 THEN P = P - (P > 0)
93 450 IF K = 21 THEN P = P + (P < L)
9D 460 GOTO 330
37 470 F = 1:D = 0: FOR P = 1 TO LEN (A$):C$ = MID$ (A$,P,1): IF F > N AND C$ < > " " THEN RETURN
BB 480 IF C$ < > " " THEN GOSUB 520:V(F) = J + 16 * (D = 1) * V(F):D = D + 1
5F 490 IF D > 0 AND C$ = " " OR D = 2 THEN D = 0:F = F + 1
08 500 NEXT : IF D = 0 THEN F = F - 1
17 510 RETURN
85 520 J = ASC (C$):J = J - 48 - 7 * (J > 64): RETURN
AB 530 A = 0: INPUT A$:A$ = LEFT$ (A$,4): IF LEN (A$) = 0 THEN RETURN
6F 540 FOR P = 1 TO LEN (A$):C$ = MID$ (A$,P,1): IF C$ < "0" OR C$ > "9" AND C$ < "A" OR C$ > "Z" THEN A = 0: RETURN
2D 550 GOSUB 520:A = A * 16 + J: NEXT : RETURN
28 560 C = INT (B / 256):C = B - 254 * C - 255 * (C > 127):C = C - 255 * (C > 255)
28 570 FOR F = 1 TO 8:C = C * 2 - 255 * (C > 127) + V(F):C = C - 255 * (C > 255): NEXT : RETURN
0A 580 I = FRE (0):A$ = "": FOR I = 1 TO L:T = INT (A / 16):A$ = MID$ ("0123456789ABCDEF",A - 16 * T + 1,1) + A$:A = T: NEXT : RETURN
1F 590 PRINT "FROM ADDRESS ";: GOSUB 530: IF S > A OR E < A OR A = 0 THEN B = 0: RETURN
80 600 B = S + 8 * INT ((A - S) / 8): RETURN
86 610 PRINT "DISK ERROR": GOTO 150
```

SOAR to new heights with flight simulation books from COMPUTE!



This collection of flight guidebooks helps you get the most out of flight simulation software. Designed to be used with Flight Simulator, Flight Simulator II, F-15 Strike Eagle, and other fighter-jet simulations, these books contain scores of scenarios that lead you across the U.S., around the world, into battle, and back to your favorite landing strips.

Flying on Instruments with *Flight Simulator*
Jonathan M. Stern — 251 pages
Here's fascinating hands-on instruction on flying with instruments in *Flight Simulator* and *Flight Simulator II*. How to fly in all kinds of weather—when the cloud ceiling is low or the visibility is reduced. How to read and fly the instrument approach charts that instrument-rated pilots use. How to take off, fly to another airport, and land—all while the clouds are a mere 200 feet off the ground. And how to navigate by using the impressive instruments available in *Flight Simulator* and *Flight Simulator II*. More than 130 actual instrument approach procedure charts. For all versions of *Flight Simulator* and *Flight Simulator II* (Apple, Amiga, Atari, Atari ST, Commodore, IBM PC and compatibles, Macintosh). *Flight Simulator* or *Flight Simulator II* required.
Perfect-bound edition:
$9.95 ISBN 0-87455-091-2
Split-wire-bound edition:
$12.95 ISBN 0-87455-103-X

COMPUTE!'s Flight Simulator Adventures for the Macintosh, Amiga, and Atari ST
David Florance, Tom R. Halfhill, and Philip I. Nelson — 227 pages
Fly through mists into mystery. This collection of 48 exciting, new, customized simulator scenarios for the Macintosh, Commodore Amiga, and Atari ST personal computers puts you in the pilot's seat. The new generation of *Flight Simulator* and *Flight Simulator II* offers better graphics, easier-to-use controls, and more airports. With this book, you'll experience hair-raising rescue missions, dangerous night landings, sightseeing side trips, and one-of-a-kind flights of fancy. Make *COMPUTE!'s Flight Simulator Adventures* your copilot and thrill to hours of flight.
Flight Simulator or *Flight Simulator II* required.
$12.95 ISBN 0-87455-100-5

Learning to Fly with *Flight Simulator*
John Rafferty
Learn to fly like a simulator expert in the imaginary world of *Flight Simulator* and *Flight Simulator II*. You'll learn all the fundamentals—from takeoffs to landings—in a single, detailed flight. Then you'll develop those flying skills by practicing standard rate turns, executing a flight plan, navigating from point A to point B, and more. Learn how to make precision landings and use the more advanced instruments in your aircraft. To cap your training—and turn you into a professional simulator pilot—you'll learn how to read and use the actual FAA approach charts commercial pilots fly with. This complete flight-instruction tutorial was written by a professional pilot.
$12.95 ISBN 0-87455-115-3

40 Great Flight Simulator Adventures
Charles Gulick — 152 pages
This bestselling volume offers 40 exciting, customized flight simulator scenarios for *Flight Simulator* and *Flight Simulator II* on the Apple II, IBM, Commodore 64, and Atari personal computers.
$10.95 ISBN 0-87455-022-X

40 More Great Flight Simulator Adventures
Charles Gulick — 236 pages
Forty more challenging flight simulator scenarios for Microsoft's *Flight Simulator* and SubLOGIC's *Flight Simulator II*. Explore the world from the air. For the Apple II, IBM, Commodore 64, and Atari personal computers.
$12.95 ISBN 0-87455-043-2

Jet Fighter School: Air Combat Simulator Tactics and Maneuvers
Richard G. Sheffield — 177 pages
Learn actual combat maneuvers and much more with this hands-on book for popular jet-combat simulators like *F-15 Strike Eagle, Jet,* and *Ace*. Information about jet fighter pilots and their aircraft, especially the F-15, provides an excellent introduction to the subject. There's even detailed data about the weapons available to jet-combat simulator pilots. The heart of the book demonstrates, step-by-step, numerous offensive and defensive combat flight maneuvers, from the Hi Yo-Yo to the Defensive Spiral Drive. Tactical tips on *F-15 Strike Eagle* scenarios round out the book. *F-15 Strike Eagle* or other jet-combat flight simulator required.
Perfect-bound edition:
$9.95 ISBN 0-87455-092-0
Split-wire-bound edition:
$12.95 ISBN 0-87455-104-8

Ask for COMPUTE! Books at your local book or computer store.
To order direct, call toll free **800-346-6767** (in N.Y. call 212-887-8525) or write **COMPUTE! Books**, P.O. Box 5038, F.D.R. Station, New York, NY 10019.
Customer Service hours are 10:00 am–12:30 pm and 1:30 pm–3:00 pm EST.

Please include $2.00 shipping and handling in the U.S. or $5.00 airmail. NC residents add 5 percent sales tax and NY residents add 8.25 percent sales tax. Please allow 4–6 weeks for delivery.

COMPUTE! Publications, Inc. abc
A Capital Cities/ABC, Inc. Company

COMPUTE! books are available outside the United States from subsidiaries of McGraw-Hill International Book Company.

Classified

SOFTWARE

COMMODORE: TRY BEFORE YOU BUY. Best selling games, utilities, educational, + classics and new releases. 100's of titles. Visa/MC. Free brochure. RENT-A-DISC, Frederick Bldg. #345, Hunt'n, WV 25701 (304) 529-3232

DISCOUNT SOFTWARE for most computers. FREE CATALOG. Sale: 5.25" DSDD Disks 25 for $13.95 ppd. WMJ DATA SYSTEMS-C, 4 Butterfly Dr., Hauppauge, NY 11788

FREE SOFTWARE for C64, C128, IBM & CPM send SASE for info (specify computer) to: PUBLIC DOMAIN USERS GROUP PO Box 1442-A1, Orange Park, FL 32067

MILLION $ "LOTTO BUSTER" PROGRAM
SCIENTIFIC. HITS JACKPOTS. NO RN/G $49.95 Ap II, Atari, MS/DOS, C64. OH res add 6% s.tax. Also: FANTASTIC DAILY NUMBER FORECASTER! Gets Straight Hits Weekly! $42.95. Both: $79.95. Large SASE gets info. Z-Way, POB 9017-C, Canton OH 44711 (216) 875-2512

More than 200 great ML routines for 64 and 128, ready to add to your own programs, in COMPUTE! Books' MACHINE LANGUAGE ROUTINES FOR THE COMMODORE 64/128. Explanations, uses, commented source code. 585 pages, $18.95. Check your local bookstore or call (800) 346-6767

FREE PUBLIC DOMAIN SOFTWARE
Request free catalog or send $2 for sample disk & catalog (refundable). Apple, C64/128 (specify) CALOKE IND., Box 18477, KC, MO 64133

IBM PUBLIC DOMAIN SOFTWARE $3 PER DISK
Send stamp for catalog (on disk). Hundreds of disks to choose from. Excellent service. Two disk sizes now available: 5¼" @ $3.00 and 3½" @ $5.00. Send for your list. Now accepting MasterCard and Visa.
JDX/C, P.O. BOX 1561, CORONA, CA 91718

C64/128 FINEST PUBLIC DOMAIN PROGRAMS Pretested quality programs *Most $1.50* *On Disk* YOU pick the programs that YOU want!!! Free diskfull of programs with first order! For a list and description, send a SASE to: JLH Co., Dept. H. Box 67021, Topeka, KS 66667

IBM CROSSWORD POWER
Up to 80 words in each puzzle. Make single puzzles or automatically create and print up to 15 different puzzles from each list of words and clues. Reuse words & clues in new puzzles without re-typing. Great for games or vocabulary review. 256 KB. Check/MO $69. Add $2 S/H. WISCO COMPUTING, 2821 Sampson Street, Wisconsin Rapids, WI 54494

FREE* IBM SOFTWARE *FREE
5¼" AND 3½" FORMATS
FREE CATALOG: AP-JP, INC.,
BOX 1155, W. BABYLON, NY 11704

IBM GRADE BOOK POWER
Enter either letter or number grades. Prints 28 different types of reports for students, teachers, parents and administrators. Most comprehensive student reports of any computer grade book program. Check/MO $59. Add $2 S/H. WISCO COMPUTING, 2821 Sampson Street, Wisconsin Rapids, WI 54494

ATTENTION!
IBM & COMPATIBLE
CURRENT AND *FUTURE* OWNERS
★
Join and support the WORLD'S LARGEST computer assistance group Now serving over 25,000 members worldwide with the BEST in technical assistance, service and products for your computer. To become a member and receive newsletters, catalog, technical assistance and membership package, send $10 (for a one year membership) to:
THE INTERNATIONAL PC COMPUTER GROUP
2269 S. University Dr. #118C
Ft Lauderdale, FL 33324
For more info call (305) 962-9019 (24 hrs)

FREE APPLE AND IBM SOFTWARE
Over 2500 Public Domain Programs on over 150 diskettes. $5 each plus $1 for shipping per order, refundable with order.
C&H ENTERPRISES
PO Box 29243, Memphis, TN 38127

Cheap Software / IBM, PCjr, Compatibles. 5.25 or 3.5 disk. For catalog of diskettes available for education, games, business, etc. Write: Morning Star Industry, Box 3095, Ann Arbor, MI 48106

$$Win with Horseracing, Greyhound, Football, Baseball & Basketball handicapping software - $52.95 each. Software Exchange, Box 5382cp, W. Bloomfield, MI 48033 Info/V/MC 800-527-9467

TALKING SOFTWARE FOR YOUR COMMODORE! Speaking Operating System Utilities. Uses BASIC "SAY" command and SPEECH 64 (or compatible) in Expansion Port. Only $17 (includes S/H Send to: Don Hunt, P.O. Box 3283, El Segundo, CA 90245

TI-99/4A Software/Hardware bargains. Hard-to-find items. Huge selection. Fast service. Free catalog. D.E.C., Box 690, Hicksville, NY 11801

MISCELLANEOUS

PERSONAL COMPUTER OWNERS CAN EARN $1000 to $5000 monthly selling simple services part time. Free list of 100 best services. A.I.M.L.U. PO Box 60369, San Diego, CA 92106

MAKE MONEY WITH YOUR PC!
Full or Part-Time! Send SASE to: Harbor Investments, Post Office Box 69, Essexville, Michigan 48732

NO HOURLY CHARGES! Chat with 11 other callers on our MODEM PARTY LINE! (300/1200 BAUD) 1-818-842-3322

PCjr LOVERS!!!!!
Subscribe to THE JUNIOR REPORT and receive continued support for your PCjr. 28 pages packed full of articles just for junior. The PCjr Club, Dept. T, POB 59067, Schaumburg, IL 60195-0067 $18/yr (12 issues). In Canada $24 US. Write for a free issue.

DUST COVERS CUSTOMIZED FOR ANY Computer/keyboard/printer/disk drive/monitor/vcr. Nylon disk wallets. All water repellant and anit-static. $5–$12. Keyskins $16. Write Future Logics, PO Box 403, Dracut, MA 01826. Call 617-782-5371/617-957-5145

COMPUTE! Classified is a low-cost way to tell over 350,000 microcomputer owners about your product or service.

Rates: $25 per line, minimum of four lines. Any or all of the first line set in capital letters at no charge Add $15 per line for boldface words, or $50 for the entire ad set in boldface (any number of lines.) Inquire about display rates
Terms: Prepayment is required. Check, money order, American Express, Visa, or MasterCard is accepted. Make checks payable to COMPUTE! Publications.
Form: Ads are subject to publisher's approval and must be either typed or legibly printed. One line equals 40 letters and spaces between words Please underline words to be set in boldface.
General Information: Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers Ad will appear in next available issue after receipt.
Closing: 10th of the third month preceding cover date (e.g., June issue closes March 10th). Send order and remittance to. Harry Blair, Classified Manager, COMPUTE!, P.O Box 5406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 275-9809.
Notice: COMPUTE! Publications cannot be responsible for offers or claims of advertisers, but will attempt to screen out misleading or questionable copy.

Advertisers Index

Train for the Fastest Growing Job Skill in America

Only NRI teaches you to service all computers as you build your own, fully IBM-PC compatible microcomputer

The biggest growth in jobs between now and 1995, according to Department of Labor estimates, will occur in the computer service and repair business, where demand for trained technicians will actually double.

You can cash in on this opportunity—either as a full-time corporate technician or an independent service person—once you've learned all the basics of computers the NRI way.

Get inside the newest, fully IBM-PC compatible Sanyo Microcomputer

As an NRI student, you'll get total hands-on training as you actually build the latest model Sanyo 880 Series computer from the keyboard up. It's fully IBM PC compatible, and, best of all, it runs programs almost twice as fast as an IBM PC. As you assemble the Sanyo 880, you'll perform demonstrations and experiments that will give you a total mastery of computer operation and servicing techniques.

Understanding you get only through experience

You need no previous knowledge to succeed with NRI. You start with the basics, rapidly building on the fundamentals of electronics with bite-size lessons. You perform hands-on experiments with your NRI Discovery Lab® and then move on to master advanced concepts like digital logic, microprocessors, and computer memories.

Learn at home in your spare time

You train in your own home at your own convenience, backed at all times by your own NRI instructor and the entire NRI staff of educators and student service support people. They're always ready to give you guidance, follow your progress, and help you over the rough spots to keep you moving toward your goal.

Get all the facts from NRI's 100-page catalog.

Your incomparable total systems training includes all this:

NRI's unique Discovery Lab® to let you design and modify circuits, diagnose and repair faults • NRI's digital multimeter featuring LCD and four key functions, complete with "talk-you-through" instruction on audio tape to teach you how pros use meters • A digital logic probe that lets you visually examine computer circuits • The latest Sanyo 880 dual speed computer with "intelligent" keyboard • 360K double-density, double-sided disk drive • 8K ROM, 256K RAM • Bundled software, including GW BASIC, MS-DOS, WordStar, CalcStar • Sanyo reference manuals to give you programming guidelines and schematics.

See other side for highlights of NRI's "hands-on" computer training →

SEND CARD TODAY FOR FREE NRI CATALOG

COMPUTER ELECTRONICS

training prepares you to service all computers as you build your own 16-bit, IBM PC-compatible computer. Total systems training includes disk drive, test equipment, bundled software, and NRI Discovery Lab®.

☑ **Check one FREE catalog only.**

- ☐ Computer Electronics
- ☐ TV, Audio, and Video Servicing
- ☐ Electronic Music Technology
- ☐ Basic Electronics

Name (Please Print) Age

Street

City State Zip

For career courses approved under G.I. Bill ☐ Check for details

230-028

COMPUTE!'s
FREE Reader Information Service

Use these cards to request FREE information about the products advertised in this issue. Clearly print or type your full name and address. Only one card should be used per person. Circle the numbers that correspond to the key number appearing in the advertisers index.

Send in the card and the advertisers will receive your inquiry. Although every effort is made to insure that only advertisers wishing to provide product information have reader service numbers, COMPUTE! cannot be responsible if advertisers do not provide literature to readers.

Please use these cards *only* for subscribing or for requesting product information. Editorial and customer service inquiries should be addressed to: COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Check the expiration date on the card to insure proper handling.

Use these cards and this address only for COMPUTE!'s Reader Information Service. Do not send with payment in any form.

COMPUTE!

101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117
118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151
152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168
169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185
186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202
203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219
220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236
237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253

Circle 101 for a one-year new U.S. subscription to COMPUTE!; you will be billed for $24

Please let us know. Do you
own: plan to buy:

- ☐ 270 Apple ________ ☐ 271
- ☐ 272 Atari ________ ☐ 273
- ☐ 274 Commodore ________ ☐ 275
- ☐ 276 IBM ________ ☐ 277
- ☐ 278 TI-99/4A ________ ☐ 279
- ☐ 280 Other ________ (specify model) ☐ 281

Please print or type name and address.
Limit one card per person.

Name ________

Address ________

City ________

State/Province ________ Zip ________

Country ________

Phone ________

Please Include ZIP Code

Expiration Date 3/31/88

GA0288

SUBSCRIBE TO COMPUTE!

For Fastest Service,
Call Our **Toll-Free**
US Order Line
800-727-6937

☐ $24.00 One Year US Subscription
☐ $45.00 Two Year US Subscription

Name ________

Address ________

City ________ State ________ Zip ________

☐ Payment Enclosed ☐ Bill me
Charge my: ☐ VISA ☐ MasterCard ☐ American Express
Account No. ________ Expires ____ / ____

Your subscription will begin with the next available issue. Please allow 4-6 weeks for delivery of first issue. Subscription price subject to change at any time. Outside the U.S.A. please add $8 for each subscription Basic Rate $24.

J2S29

Place
Stamp
Here

COMPUTE! Reader Service
P.O. Box 2141
Radnor, PA 19089

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 7478 DES MOINES, IA

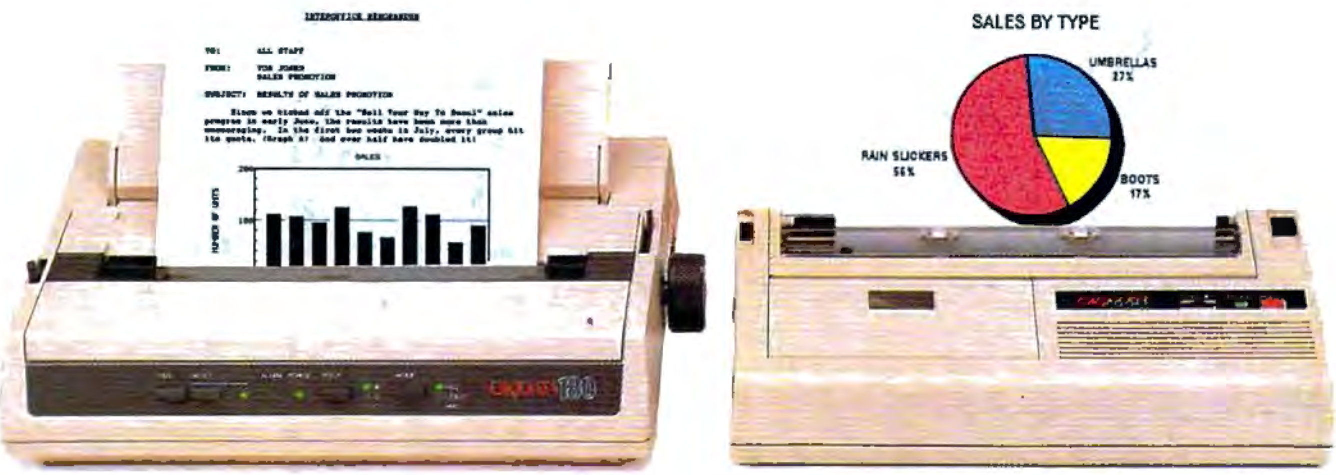
POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!

P.O. Box 10955
Des Moines, IA 50347-0955


Wizardry®
Proving Grounds of the Mad Overlord©
The First Scenario
Adventure with your Apple, IBM PC or compatible, C64/C128 or Macintosh
Knight of Diamonds©
The Second Scenario*
Adventure with your Apple, IBM PC or compatible
Legacy of Llylgamyn©
The Third Scenario*
Adventure with your Apple, IBM PC or compatible
The Return of Werdna©
The Fourth Scenario
Note: Wizardry Expert Level. Adventure with your Apple.
PLAY IT TO THE HILT!
Wizardry's four superior scenarios challenge you to overcome devious tricks and sudden traps, strange and mysterious encounters, maze upon maze of puzzles, stimulating and intricate subplots and sophisticated themes. Ingenious multi-level mind-bending tests abound with vile creatures, unspeakable villains, courageous adventurers and strong allies to hinder and help you achieve ultimate victory.
SIR-TECH
Sir-tech Software, Inc., P.O. Box 245, Ogdensburg, NY 13669 (315) 393-6633
THE INTERNATIONALLY ACCLAIMED WIZARDRY GAMING SYSTEM
* Characters from Proving Grounds of the Mad Overlord required for Knight of Diamonds and Legacy of Llylgamyn. ™ Registered trademark of Sir-tech Software, Inc. © Copyright by Sir-tech Software, Inc. Apple is a registered trademark of Apple Computer, Inc. Macintosh is a registered trademark of Apple Computer, Inc. IBM is a registered trademark of International Business Machines. C64 and C128 are registered trademarks of Commodore International.

All work and some play.



The New OKIDATA 180 and the OKIMATE® 20: Office Quality and Home Versatility.

When the work you do at home needs to look good enough for the office, you need one of these two printers.

Get the new OKIDATA 180 and get crisp near letter quality printing at time-saving print speeds. Enjoy its easy-to-use push-button front panel, easy paper handling and ability to print on anything from letterhead to four-part forms.

And, for pure compatibility, use its dual Commodore® and Epson®-compatible parallel interface.

Take home the OKIMATE 20 and unlock the power of color graphics and 24-element quality.

Give life to your letters with sharp NLQ printing and a choice of built-in fonts. And add impact to your overheads with acetates in hundreds of colors.

The new OKIDATA 180 and the OKIMATE 20 both work beautifully with either your Commodore 64 or 128, your IBM® PC or compatible, Tandy® and most Apple® computers.

These two printers bring home the OKIDATA tradition of toughness you know from the office. *At a price that could convince you to buy both.*

Call 1-800-OKIDATA, Ext. 25, for the name of the retailer nearest you.

OKIDATA®
an OKI AMERICA company