Word Search: Puzzle Program Inside For Commodore, Atari, Apple, IBM, & TI

COMPUTE!

$2.95
September
1985
Issue 64
Vol. 7, No. 9
$3.75 Canada
02193
ISSN 0194-357X

The Leading Magazine Of Home, Educational, And Recreational Computing

THE AMIGA FROM COMMODORE: An In-Depth Review

Programs Inside:

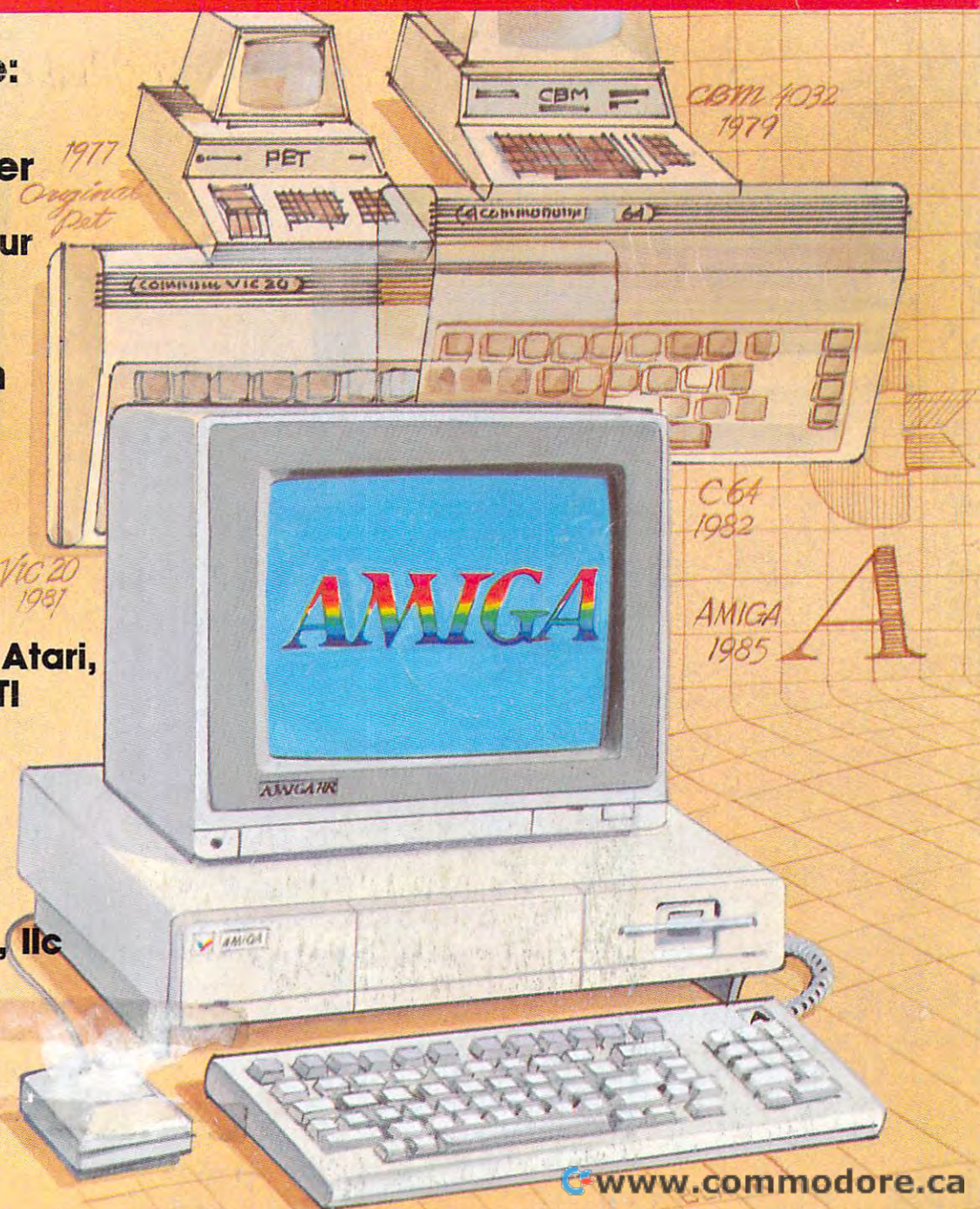
Disk Commander
Add 18 Powerful Commands To Your Commodore 64

Atari Animation

All About IBM Batch Files

Jump Search
For Commodore, Atari, Apple, IBM, And TI

Easy Apple Screen Editing
Enhanced BASIC Line Editor
For Apple II+, IIe, IIc



With Commodore 128's instead of Apple IIc's, these kids would be on computers instead of in line.

Meet the Commodore 128.™ The new personal computer that's already destined to be at the head of its class. It not only outsmarts the Apple® IIc in price, it comes out way ahead in performance.

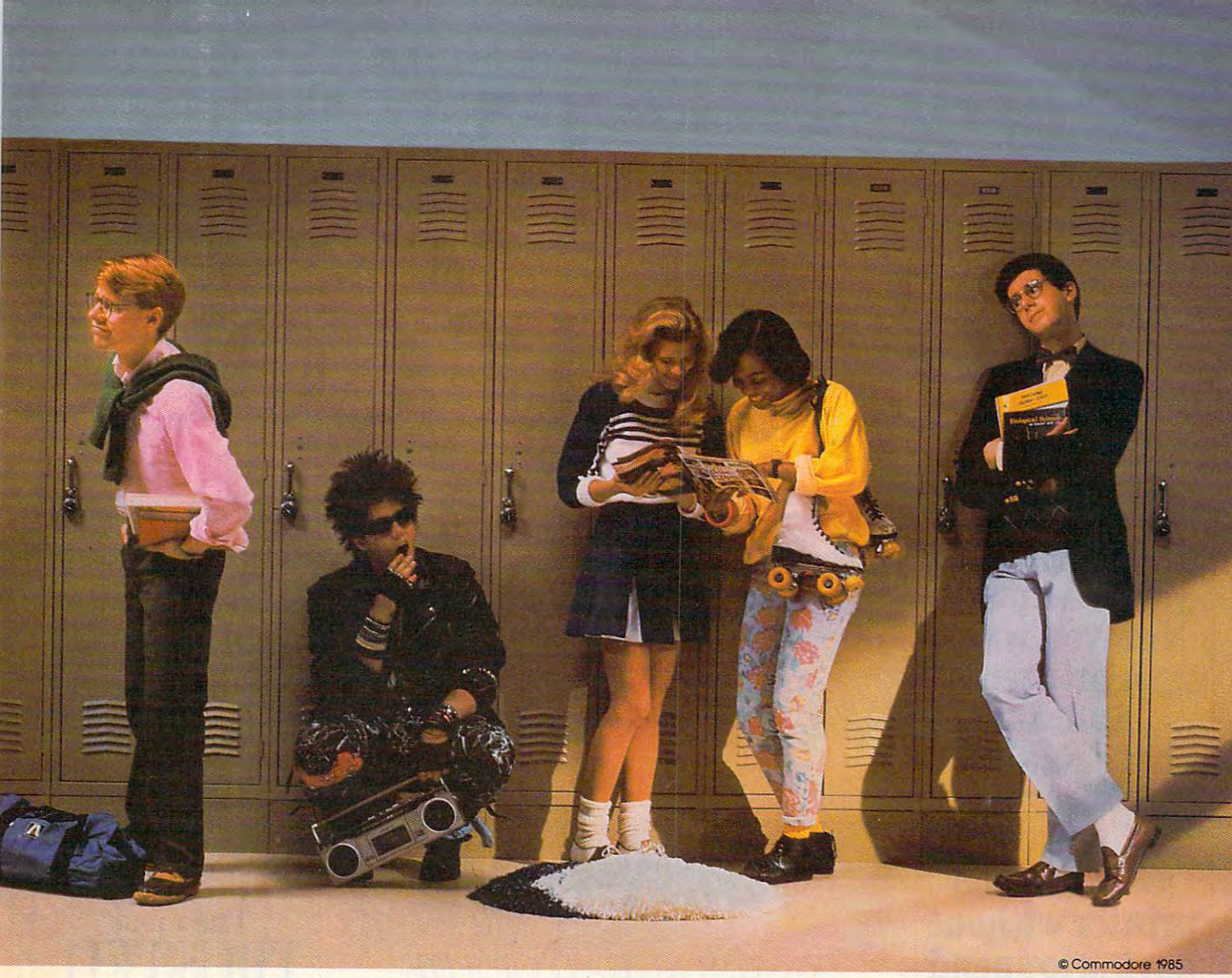
A lower price is welcome news to any tight school budget. But it's not the only way the 128 saves you money. You only need to buy one external disk drive to network eight Commodore 128's from a teacher's desk. The Apple IIc has a built-in disk drive. That adds a built-in extra cost for a feature your classroom doesn't really need. There's even more to the Commodore 128 than being able to put more students on computers for less money.

There's more intelligence.

As your students grow smarter, so does the Commodore 128. It's a computer they'll find more useful because it's capable of expanding from 128K to 512K memory. The Apple IIc doesn't expand, making it less versatile. It also doesn't have the Commodore 128's expanded keyboard that offers more commands for easier programming and more varied use of graphics and text. Or a numeric keypad that's a real necessity when using a computer in math or science classes.

More software.

The Commodore 128 is one new personal computer you won't have to wait for software

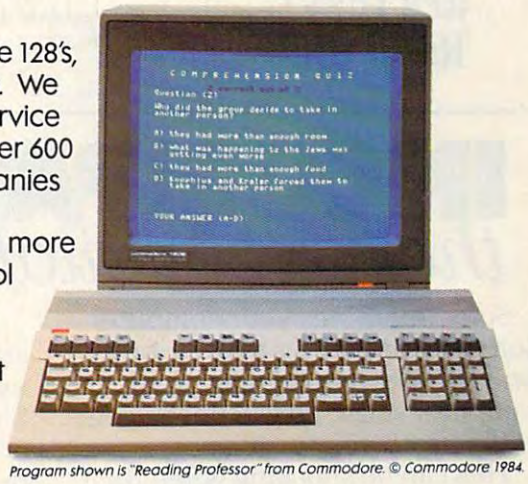


to catch up to. It's compatible with over 3,000 programs and peripherals designed for the Commodore 64™, many of which your school may have. And in the 128 and CP/M® modes, there are programs for students who want to get down to business and learn spreadsheets, file management or other professional uses.

More convenience.

Here's a feature you'll also find intelligent. Should anything go wrong with your Commodore 128's, just call toll-free 800-247-9000. We have an exclusive school service agreement with RCA and over 600 independent service companies to give you fast repair.

When it comes to fitting more computers into a tight school budget, come size up the Commodore 128. For more intelligence, and a price that makes sense, you'll find it's one sure way to lower the cost of a higher education.



Program shown is "Reading Professor" from Commodore. © Commodore 1984.

COMMODORE 128™ PERSONAL COMPUTER

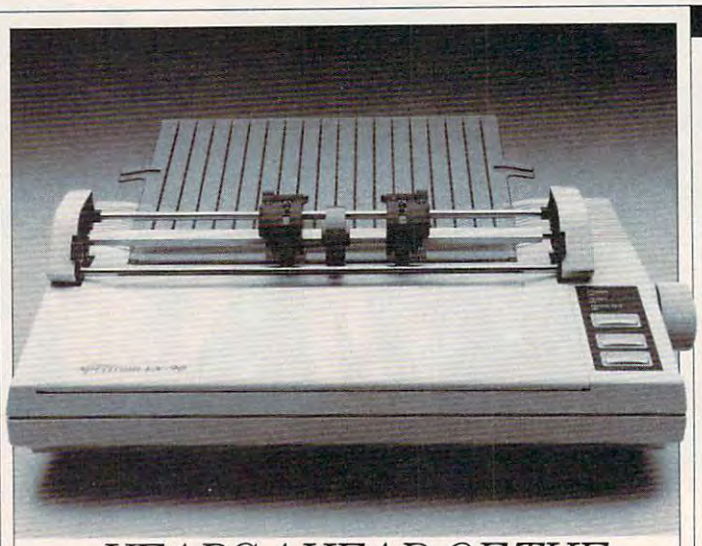
A Higher Intelligence

®Apple is a registered trademark of Apple Computer, Inc.
®CP/M is a registered trademark of Digital Research, Inc.

www.commodore.ca

Epson *all your conc*

You have your computer. Now you want a printer. And you have natural concerns. Can you get a serious printer without having to spend serious money? What about compatibility? How reliable are they? Read on.



YEARS AHEAD OF THE
COMPETITION

While most other printers only offer standard 90-day warranties, we stand behind Epson printers for a full year.

Of course, the warranty is one benefit of owning an Epson that very few people ever have the opportunity to take advantage of.

Another benefit you probably won't use is the industry's strongest local service network. But it's nice to know it's there.

Is it an Epson ????? OR ????? Isn't it an Epson?

For over a decade, major computer companies have staked their own reputations on the reliability and performance of Epson printers by coming to us for the printers they sell with *their* computer systems.

The most TRUSTED printers in the WORLD

Today, more homes and businesses rely on Epson printers than any other printers in the world. And the reason is this. Day to day, year to year, you can count on them. Epson printers are built for years of worry-free service. In fact, most of our printers have been working longer than most of our competitors have been in the business.

You know you need a printer to get the best from your computer. And, if you're going to spend your good money, you shouldn't have to settle for something unproven or less than reliable.

Check around. Now you can afford an Epson. Why settle for anything else?

INCOMPATIBILITY: *the irreconcilable difference.*

Software drives your computer. Software also drives your printer. And it's very possible that your computer software will not be compatible with your printer.

This can be shocking when the p's on your monitor come out as q's on your printer.

But Epson printers are the standard to which virtually all major computer software is written. So whether you have an IBM® PC or the newest name on the market, there are Epson printers for virtually any hardware/software combination.

answers erns in print.

HARD COPY DECISIONS MADE EASY.

For almost any computer. With the Epson Spectrum™ LX-80 printer, and interface cable, you can work with almost any computer. Most often used with the Apple IIe® and IBM PC, it prints out 100 characters per second in draft mode. And 16 characters per second, Near Letter Quality. With SelecType push-button typestyle control and excellent graphic capabilities, it offers a lot of performance for not a lot of money. Available through Epson authorized computer specialty stores.

"So the question is: 'Which Epson?'"

For the Apple IIc® and IBM® PCjr. The Epson Spectrum™ LX-90 is the perfect match for the Apple IIc and IBM PCjr. Also printing 100 characters per second, draft, and 16 characters per second in Near Letter Quality mode, the Spectrum LX-90 features friction and tractor feeds for both plain paper and continuous form computer paper. This makes it better for business applications. It uses a Printer Interface Cartridge (PIC™) to assure maximum compatibility with your computer.

Epson dot matrix printers have very graphic advantages. Besides the versatility of draft and Near Letter Quality and the corresponding speed options, they can draw some very convincing pictures, including true circles.

ONLY EPSON PRINTERS COULD GENERATE PRINT LIKE THIS

"Thank you for making such a rugged yet so fantastic piece of hardware."
L.F.G., Martinez, CA

"...a relief in this industry often faulted by products designed to last about ninety days."
John Cameron, Medical Director, Raleigh Hills Hospital

"I have had my Epson about six weeks... my only regret is that I didn't replace my old printer a year ago."
L.A.R., 29 Palms, CA

Recently in Martinez, California, two cats knocked an Epson printer 4 feet onto a tile/concrete floor. Although it didn't land on its feet, the printer is still running.

NOW THAT YOU CAN AFFORD AN EPSON, WHAT ARE YOU WAITING FOR?

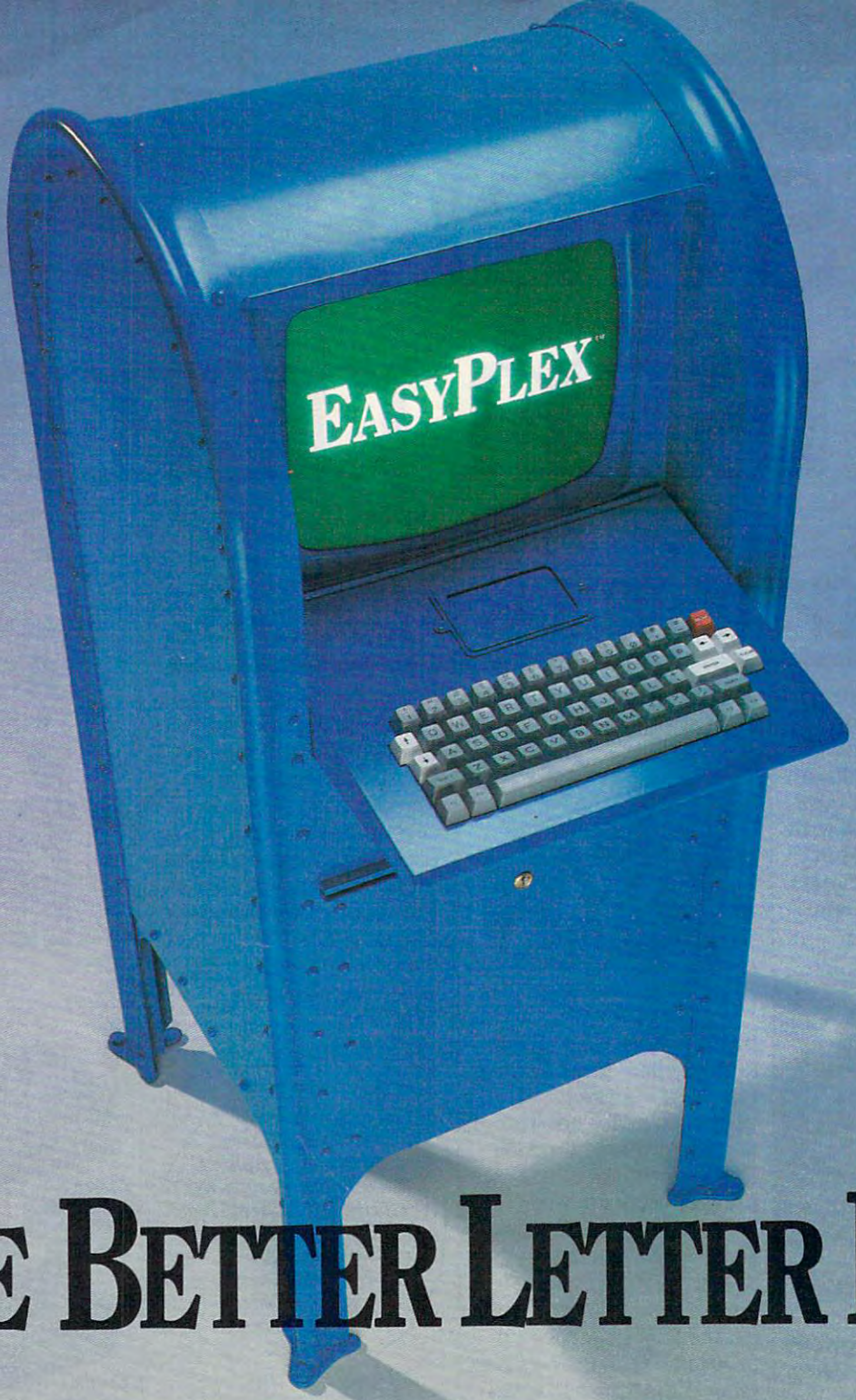
Epson Spectrum LX-80

Epson Spectrum LX-90

For more details, call (800) 421-5426 or in California (213) 539-9140.

EPSON

Number One.
And built like it.



THE BETTER LETTER BOX

Introducing EasyPlex.™ The new, easy-to-use electronic mail system from CompuServe.

Finally! Electronic Mail that's so easy to use you can start composing and sending messages the first time you get online.

Designed for various experience levels, EasyPlex has a menu mode with simple, easy-to-follow directions for beginners, and it lets experienced users save time by working in the prompt or command modes. With EasyPlex, you can compose, edit, send, file, and take advantage of sophisticated options previously available only with more expensive services.

CompuServe's EasyPlex lets friends and relatives, associations and club members communicate any time of the day or night. And small business owners, real estate professionals, insurance agents, lawyers, writers, etc. can communicate quickly and simply—either interoffice or interstate. "It's Easy." "Just Plex it!"

Best of all, EasyPlex is available to all CompuServe subscribers. And, along with EasyPlex, you get hundreds of valuable and entertaining computing options. Plus the assurance of belonging to the largest, fastest growing computer information service in the world and the premier supplier of business information to FORTUNE 500 companies.

Start communicating! To buy a CompuServe Subscription Kit, see your nearest computer dealer. To receive our informative brochure or to order direct, call or write:

CompuServe®

Information Services, P.O. Box 20212
5000 Arlington Centre Blvd., Columbus, Ohio 43220

800-848-8199

In Ohio, call 614-457-0802

An H&R Block Company

COMPUTE!

SEPTEMBER 1985
VOLUME 7
NUMBER 9
ISSUE 64

FEATURES

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 70 before typing in programs.

TOLL FREE Subscription Order Line
800-334-0868 (In NC 919-275-9809)

AP Apple, **Mac** Macintosh, **AT** Atari, **V** VIC-20, **64** Commodore 64, **+4** Commodore Plus/4, **16** Commodore 16, **128** Commodore 128, **P** PET/CBM, **TI** Texas Instruments, **PC** IBM PC, **PCjr** IBM PCjr, **CC** Radio Shack Color Computer.
*General interest.

COMPUTE! Publications, Inc.
One of the ABC Publishing Companies:
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019

Address all inquiries to:
P.O. Box 5406, Greensboro, NC 27403

COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published monthly by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. Send subscription orders or change of address (P.O. form 3579) to **COMPUTE!** Magazine, P.O. Box 10954, Des Moines, IA 50340. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. ISSN 0194-357X.

Editors Notes

Last month we mentioned some apparent communication problems regarding access to the new Amiga from Commodore. We're happy to report that comments in our editorial became moot before they reached print. Commodore's new senior management team moved quickly and smoothly to see that we, along with other magazines in the industry, received even-handed treatment in access to information.

The Amiga is an important product. We see a significant, lasting change in the way personal computers will be used and programmed and, thus, in the ways we cover computers. With the introduction of the Amiga (see the story on page 16) and the ST from Atari, consumer computing will never be the same again.

Among other things, BASIC now faces its first serious challenge as the language of popular computing. When you turn on these new computers, you don't see the familiar BASIC greeting "READY." Instead, you see a Macintosh-like "desktop" screen with icons, etc. This manager is called Intuition on the Amiga, GEM on the ST. BASIC is only one of several options, several languages you could load into the computer from disk. A simple command, however, exits this environment and lands you in an IBM PC-like AmigaDOS, said to be quite like Unix, an operating system first developed for large minicomputers. The Atari ST's TOS will be similar. Both are command-rich systems, nearly languages in themselves.

COMPUTE! expects to continue to publish the majority of its programs in BASIC. The new machines' BASICs are large and fast. They include a generous set of graphics and sound instructions. Above all, everyone who buys an ST or an Amiga will have BASIC. That language is being shipped with, though not built into, these computers.

Interestingly, most commercial software announced so far for the ST and Amiga is not being written in machine language. Instead, it is being written in C, a language popular among professional programmers which has a reputation for portability between computers. Some have argued that this spells the end of assemblers, the end of writing machine language programs. We do not find that argument compelling.

The argument goes like this: The new machines are faster (because the microprocessor, the 68000, is more efficient) and thus maximizing speed of execution by using machine language is no longer necessary. Compiled languages like C run sufficiently quickly. *Lotus 1-2-3* is written in C. Also, some new BASICs and operating systems are largely C.

The other factor in favor of machine language, its conservation of memory, is now less critical, too. Compilers can use up computer memory rapidly. Amiga BASIC, written mostly in C, is about 96K large; Commodore 64 BASIC, written entirely in machine language, uses up only 8K. Instead of having to fit everything into 64K, the maximum memory which can be easily accessed by the older 8-bit chips, the new computers can access megabytes of memory. Tecmar, an Ohio company, is developing an expansion board for the Amiga which adds up to two megabytes of memory. Hence, bulky, compiled programs don't cause much of a problem. There's memory to spare. However, even though the Amiga and ST each have 192K of ROM space, both machines' operating systems—written largely in C—have to be supplied on disk with early models. The compiled C is too big to be built into ROM until programmers can optimize and condense the code.

C has its advantages, but one fact is overlooked: Machine language is the computer's language. All other languages are compromises, less direct ways of telling the computer what you want it to do. This indirection slows the computer down for many of the same reasons that you would be slowed down in a foreign country. No matter how similar the two languages, from time to time you would be forced to resort to hand signals, symbols, even to looking things up in a dictionary. Likewise, a compiled programming language results in a more or less indirect communication with the computer. Even the best compilers produce bulkier and less efficient programs than does pure machine language.

Something similar to the current popularity of C happened when home computers were first introduced. BASIC was then the most common language for commercial programs. Spreadsheets, word processors, and games were sold which were entirely BASIC. They were slow, had few features, and used up much of the available memory space.

Now that there is a transition from 64K to 512K, quadruple the processing speed, and far better graphics and sound—most any good program is going to be impressive. The new machines make their software look good in the same way that calculators made the early 8K Commodore PET look good. It's a whole new level of power and control. But the shock of the new doesn't last. Software companies will compete along the classic lines: They will all try to offer the fastest product with the most features. Once again we are likely to see a migration to machine language as programmers vie with each other to take their machines to the limit.

The 68000 is not a new chip, but it is new to home computers. Introduced by Motorola in 1981, it cost over $200 until recently. It is the chip in the Apple Macintosh, and sales of that computer have helped drive down the price to its current $20, making it affordable as the new consumer CPU. How does the 68000 differ from the 6502, the chip in most current popular computers (Apple, Atari, Commodore, etc.)? Essentially, things like multiplying large numbers are easier to do, fetching and storing is faster and more efficient, what took several steps to accomplish in the 6502 can now be done in a single operation.

Of course, we won't see the ultimate software the minute the new hardware is introduced. It will take time for programmers to investigate the new territory. But judging from the preliminary software we've seen, the new computers offer stunning opportunities for creative programming and—whatever languages are used—the resulting software will take us far beyond what we've experienced on today's home computers. We plan to bring you some of that stunning programming in the pages of COMPUTE! in the coming years.

Robert Lock

Editor in Chief

Richard Mansfield

Senior Editor

SUMMER GAMES II.™
EIGHT NEW WAYS TO GO FOR THE GOLD.

Sure Summer Games was great, but why stop there? Let Summer Games II take you even farther with eight new events including cycling, fencing, kayaking, triple jump, rowing, high jump, javelin and even equestrian. They can all be played by up to eight players and some, like cycling, rowing and fencing challenge you with realistic head-to-head competition.

First, you decide which of the 18 different countries you're going to represent. Then, in true Olympic fashion, you will need the proper strategy and mental toughness, not just speed and agility to excel in each event. It's so realistic, there's even an opening and closing ceremony along with medal presentations after each event.

It's not too early to get ready for 1988. With the right diet, proper training and hours of practice you just might make it. In the meantime, put on your sweatsuit, grab that joystick and let Summer Games II give you eight new ways to **Go For The Gold!**

EPYX™ COMPUTER SOFTWARE

Strategy Games for the Action-Game Player

Publisher	Gary R. Ingersoll
Editor In Chief	Robert C. Lock
Director of Administration	Alice S. Wolfe
Senior Editor	Richard Mansfield
Managing Editor	Kathleen Martinek
Editor	Tom R. Halfhill
Assistant Editor	Philip Nelson
Production Director	Tony Roberts
Production Editor	Gail Cowper
Editor, COMPUTE!'s GAZETTE	Lance Elko
Technical Editor	Ottis R. Cowper
Assistant Technical Editors	John Krause, George Miller
Program Editor	Charles Brannon
Features Editor	Selby Bateman
Assistant Editor, COMPUTE!'s GAZETTE	Todd Heimarck
Feature Writer	Kathy Yakal
Programming Supervisor	Patrick Parrish
Editorial Programmers	Tim Victor, Kevin Mykytyn, Kevin Martin
Submissions Reviewer	Mark Tuttle
Programming Assistant	David Florance
Copy Editors	Joan Rouleau, Ann Davies
Executive Assistant	Susan Young
Administrative Assistants	Julia Fleming, Iris Brooks, Jan Kretlow
Associate Editors	Jim Butterfield, Toronto, Canada Harvey Herman, Greensboro, NC Fred D'Ignazio, Roanoke, VA David Thornburg, Los Altos, CA
Contributing Editor	Bill Wilkinson
COMPUTE!'s Book Division Editor	Stephen Levy
Assistant Editors	Gregg Keizer, J. Blake Lambert
Administrative Assistant	Laura MacFadden
Director, Book Sales & Marketing	Steve Voyatzis
Assistant	Carol Dickerson
Production Manager	Irma Swain
Art & Design Director	Janice R. Fary
Assistant Editor, Art & Design	Lee Noel
Mechanical Art Supervisor	De Potter
Artists	Debbie Bray, Dabney Ketrow
Typesetting	Terry Cash, Carole Dunton
Illustrator	Harry Blair
Director of Advertising Sales	Ken Woodard
Production Coordinator	Patti Stokes
Administrative Assistant	Kathleen Hanlon
Promotion Assistant	Caroline Dark
Customer Service Manager	Philippa King
Dealer Sales Supervisor	Gail Jones
Assistants	Debi Goforth, Liz Krusenstjerna, Rhonda Savage
Individual Order Supervisor	Judy Taylor
Assistants	Betty Atkins, Gayle Benbow, Mary Hunt, Jenna Nash, Chris Patty
Warehouse Manager	Lonnie Arden
Staff	Harold Ayers, Steve Bowman, Larry O'Connor, David Hensley
Data Processing Manager	Leon Stokes
Assistant	Chris Cain
Vice President, Finance & Planning	Paul J. Megliola
Director, Finance & Planning	R. Steven Vetter
Accountant	Robert L. Bean
Financial Analyst	Karen K. Rogalski
Staff	Dale Branch, Jill Pope
Credit Manager	Barry L. Beck
Staff	Sybil Agee, Anne Ferguson, Pat Fuller, Doris Hall, Linda Miller, Mary Waddell, Jane Wiggs
Purchasing Manager	Greg L. Smith

Gary R. Ingersoll, President
Paul J. Megliola, Vice President, Finance and Planning
Debi Nash, Executive Assistant
Anita Armfield, Assistant

MPA
Magazine Publishers Association

Member
ABC
Audit Bureau of Circulations

Coming In Future Issues

Save With Replace: Debugged At Last

In-Depth Review: Atari 520ST

Amiga's Amazing Graphics

The Witching Hour: Strategy Game For Commodore, Atari, Apple, IBM, and TI

Apple II Pull-Down Menus

Lightning Renumber For Atari

64 Multicolor Graphics Made Easy

COMPUTE! Publications, Inc. publishes:

COMPUTE!
COMPUTE!'s GAZETTE
COMPUTE! Books
COMPUTE!'s GAZETTE DISK

Corporate office:
324 West Wendover Avenue
Suite 200
Greensboro, NC 27408 USA

Mailing address: COMPUTE!
Post Office Box 5406
Greensboro, NC 27403 USA

Telephone: 919-275-9809

Subscription Orders

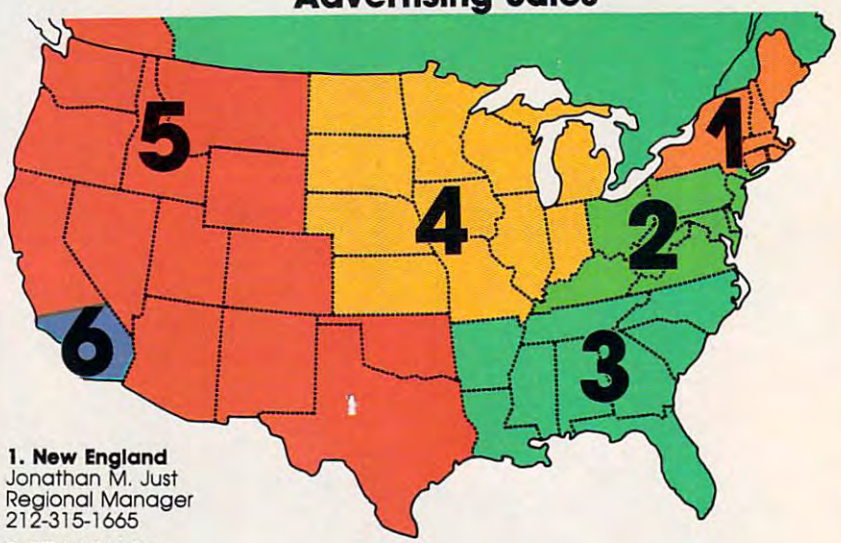
COMPUTE! Circulation Dept.
P.O. Box 914
Farmingdale, NY 11737

TOLL FREE Subscription Order Line
800-334-0868
In NC 919-275-9809

COMPUTE! Subscription Rates (12 Issue Year):

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

Advertising Sales



1. New England
Jonathan M. Just
Regional Manager
212-315-1665

2. Mid Atlantic
John Saval
Eastern Advertising Manager
212-315-1665
Kathy Hicks
Marsha A. Gittelman
215-646-5700
Brian S. Rogers
212-674-0238

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

4. Midwest
Gordon Benson
312-362-1821

5. Northwest/ Mountain/Texas
Phoebe Thompson
408-354-5553

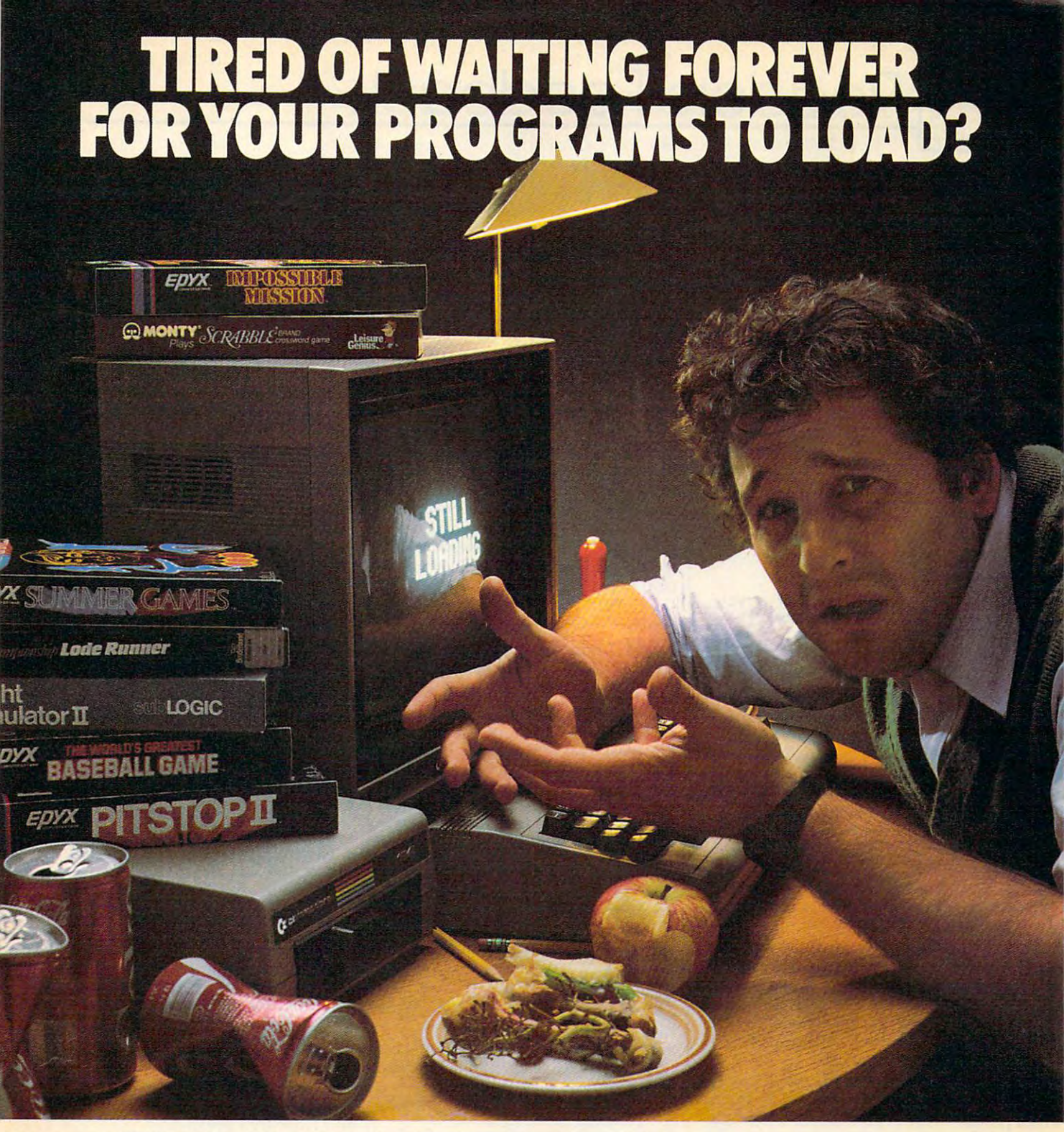
6. Southwest
Ed Winchell
213-378-8361

Director of Advertising Sales
Ken Woodard

COMPUTE! Home Office 919-275-9809.

Address all advertising materials to:
Patti W. Stokes
Advertising Production Coordinator
COMPUTE! Magazine
324 West Wendover Avenue,
Greensboro, NC 27408

TIRED OF WAITING FOREVER FOR YOUR PROGRAMS TO LOAD?



INTRODUCING THE FAST LOAD CARTRIDGE FROM EPYX.

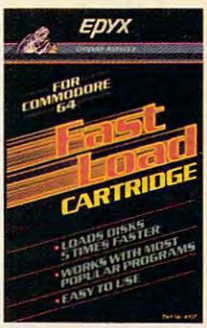
You're tired of waiting forever for your Commodore 64 programs to load. But it's no use glaring at your disk drive. Calling it names won't help, either. It was born slow — a lumbering hippo. You need the FAST LOAD CARTRIDGE from EPYX. FAST LOAD transforms your Commodore 64 disk drive from a lumbering hippo into a leaping gazelle. With FAST LOAD, programs that once took minutes to load are booted up in a matter of seconds.

FAST LOAD can load, save and copy your disks five times faster than normal. It plugs into the cartridge port of your Commodore 64 and goes to work automatically, loading your disks with ease. And that's only the beginning. You can copy a single file, copy the whole disk, send disk commands, and even list directories without erasing programs stored in memory.

And unlike other products, the FAST LOAD CARTRIDGE works with most programs, even copy protected ones, including the most popular computer games.

The FAST LOAD CARTRIDGE from Epyx. Easy to insert, easy to use and five times faster. So why waste time waiting for your disks to load?

Speed them up with FAST LOAD!



EPYX™

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.

Relational Operators

I recently typed in the TI-99/4A game "Circus" (COMPUTE!, February 1984) and noticed the following statement in line 50:

```
SC=SC+(H=120)*-50+(H=112)*-7
5+(H=104)*-100+((H=128)*(M1=
1)*250)
```

How does this statement work?

Dan Schwarz

Although your question concerns a TI program, the answer applies to BASIC programming on a wide variety of computers. The complex statement that has you puzzled calculates the game score (variable SC) by using the equal sign (=) as a relational operator. *Though its syntax looks odd, it efficiently takes the place of several IF-THEN statements.*

In "Circus" the balloon (variable H) popped by the clown can be in the bottom row (character number 120), in the middle row (character 112), or the top row (104). Character 128 signifies the bonus balloon. A bottom row balloon scores 50 points, the middle row scores 75, the top row is worth 100, and a bonus balloon scores 250 points provided its color is yellow (M1=1; see line 80 of the program).

The expression (H=120) doesn't change the value of H. Instead, it performs a logical test similar to IF. When H equals 120—when you pop a bottom-row balloon—this expression returns a value of −1. Any expression that evaluates to −1 is considered to be true. *When H equals any other number, the computer returns 0 to show the expression is* false. *(TI, Commodore, and IBM PC/PCjr computers evaluate true expressions to −1; Apple, Atari, and Timex/Sinclair computers use 1 rather than −1.)*

Say that the clown pops a balloon in the bottom row. Since H equals 120, the expression (H=120) is true and evaluates to −1. This value is multiplied by −50 to add 50 to the score (multiplying two negative numbers produces a positive number). Since H=120 is true, the other expressions (H=112, H=104, and H=128) are false, so the multiplications yield 0 and the score doesn't change. The remaining expressions in the example increment the score when you pop balloons in the middle and upper rows or pop the bonus balloon (character 128) when it's yellow. Other relational operators include <, >, AND, OR, and NOT (if available in your dialect of BASIC). String expressions work as well as numeric expressions, and relational operations are particularly efficient when combined with ON-GOTO or ON-GOSUB statements.

Atari Tape-To-Disk Transfer

When I bought a disk drive for my Atari system, I was faced with retyping all the machine language programs (like *SpeedScript*, COMPUTE!, May 1985) I had previously saved on tape. Instead, I found a way to use "Atari MLX" to load a machine language program from tape, and then either save it as a binary disk file or make a boot disk. To make a binary file, change line 390 of MLX as follows:

```
390 IF N=-19 THEN MEDIA= ASC("
    D"):DTYPE=70:GOTO 720
```

Change line 390 as follows to make a boot disk:

```
390 IF N=-19 THEN MEDIA= ASC("
    D"):GOTO 720
```

After that's done, run MLX and follow the instructions, loading from tape and saving to disk when appropriate.

David L. Pettite

Thank you for the information. Readers should note that this temporary change to line 390 is only for converting tape files to disk files. It is not *a correction to MLX, and should not be permanently incorporated into your copy of Atari MLX.*

64 Key Beeper

Is there a program for the Commodore 64 that will cause a beep when a key is pressed?

Jeffrey Gurr

The following program adds audible feedback to the keyboard of your 64, as found on Atari computers. (Ironically, owners of Atari 400s and 800s frequently write us for a way to turn off *the built-in keyboard beep.) The program puts a short, interrupt-driven machine language routine in an unused memory area (679–760), activates the beep routine, then erases itself. Be sure to save a copy of the program* before *running it, and turn up the volume on your TV or monitor. This routine is designed to be used in direct mode (while you're typing a program, etc.) rather than in program mode (while a program is running). It doesn't interfere with most BASIC operations, but any program that creates other sounds, changes the hardware interrupt vector, or alters locations 3–4 and 679–760 may disrupt the beep or cause other problems. You should always disable the beep (press RUN/STOP–RESTORE) before running other programs. Enter SYS 679 to turn it back on.*

```
1 S=679:N=S
2 READQ:IFQ=256THEN4
3 POKEN,Q:N=N+1:CK=CK+Q:GOTO2
4 IFCK<>9233THENPRINT"ERROR IN
   DATA":END
5 SYS(S):NEW
6 DATA 120,169,206,141,20,3,16
  9,2,141,21,3
7 DATA 162,0,138,157,0,212,232
  ,224,25,208,248
8 DATA 169,15,141,24,212,169,6
  7,141,5,212,169
9 DATA 17,141,1,212,88,96,165,
  197,201,64,240
10 DATA 30,197,3,208,6,165,4,2
   40,2,208,24
11 DATA 169,32,141,4,212,169,3
   3,141,4,212,165
12 DATA 197,133,3,169,1,133,4,
   208,4,169,0
13 DATA 133,4,76,49,234,256
```

Simpler IBM Unprotection

On CompuServe's PC-SIG disk #184 you can find a simpler procedure for unlocking protected IBM BASIC programs (see "Unlocking IBM BASIC Programs" by Peter Nicholson, COMPUTE!, June 1985). Written by Todd Pollock, this method uses BSAVE and BLOAD commands to restore the portion of RAM that is disabled by a protected program. First, type in any two- or three-line BASIC program such as this:

```
10 PRINT "HELLO"
```

Flight Simulator II

For Apple, Atari®, & Commodore 64™

Put yourself in the pilot's seat of a Piper 181 Cherokee Archer for an awe-inspiring flight over realistic scenery from New York to Los Angeles. High speed color-filled 3D graphics will give you a beautiful panoramic view as you practice takeoffs, landings, and aerobatics. Complete documentation will get you airborne quickly even if you've never flown before. When you think you're ready, you can play the World War I Ace aerial battle game. Flight Simulator II features include ■ animated color 3D graphics ■ day, dusk, and night flying modes ■ over 80 airports in four scenery areas: New York, Chicago, Los Angeles, Seattle, with additional scenery areas available ■ user-variable weather, from clear blue skies to grey cloudy conditions ■ complete flight instrumentation ■ VOR, ILS, ADF, and DME radio equipped ■ navigation facilities and course plotting ■ World War I Ace aerial battle game ■ complete information manual and flight handbook.

See your dealer . . .

or write or call for more information. For direct orders enclose $49.95 plus $2.00 for shipping and specify UPS or first class mail delivery. American Express, Diner's Club, MasterCard, and Visa accepted.

Order Line: 800 / 637-4983

subLOGIC Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

```
20 GOTO 10
30 END
```

Save the program by entering this line: BSAVE "UNPRO.CIM",&H400,&H7F. To unprotect a protected program, load the protected program into memory, then enter this line: BLOAD "UNPRO.CIM". I suspect that Nicholson's procedure may be required on some compatibles, since Pollock's does not simply query a standard location for standard information. A quick test on my friend's Sperry PC-compatible showed that it disables the BLOAD command while a protected program is in memory. However, Pollock's procedure does have the advantage of requiring much less typing.

Guy R. Winters

We tested this method on the PC and PCjr and found that you need to BSAVE only one byte of memory. Type in any one-line program such as 10 END. Then enter this command: BSAVE"UN.PRO",1124,1. The BSAVE command saves one byte of memory at location 1124 (&H464 hexadecimal). Now load a protected program (one that was saved with SAVE"filename",P), and load the one-byte file with BLOAD"UN.PRO". On the PC/PCjr, the protection evaporates and you can list, edit, or save the program as usual. Also, PEEK and POKE are reenabled in direct mode.

The PC and PCjr use location 1124 as a flag: It contains 0 when an unprotected program is in memory and 254 after you load a protected program. The BSAVE shown above saves location 1124 at a time when we know the flag is set to 0. The BLOAD simply loads the 0 back into location 1124, resetting the flag to signify no protection. As you found by testing your friend's Sperry, "compatibility" is a relative concept. Evidently one of the Sperry designers knew or anticipated this trick, and prevented it by disabling BLOAD.

Although program protection disables POKE and PEEK in immediate mode, both commands are still legal in program *mode (at least on the PC/PCjr). Thus, a protected program can unprotect itself while running (for instance, if you enter a password) and an unprotected program can protect itself as well. The PCs we tested put a 254 in location 1124 to indicate protection, but in fact any nonzero value seems to set the protection flag: Editing, listing, PEEKing, and POKEing are ruled out, and you can resave the program only in protected format.*

Disabling Apple's Break Key

According to your answer to Alex Tarlecky's letter in December 1984, the RESET key can be disabled on the Apple IIc with the command POKE 1012,PEEK(1012) AND 10. But is there a way to also disable the CONTROL-C function to keep people from breaking out of my programs?

Mike Sanders

Yes, there is. After Applesoft BASIC executes a program statement, it checks for any errors that might have occurred. At the same time, it checks to see if CTRL-C was pressed. If so, Applesoft responds as it does when it encounters a syntax error or illegal quantity error. Normally, it stops the program and displays an appropriate error message (BREAK IN line#).

The secret to trapping CTRL-C is an instruction that changes the way Applesoft handles such errors—the ONERR statement. For instance, once the computer executes a statement such as ONERR GOTO 1000, it responds to any error—including the CTRL-C function—by transferring control to line 1000 (or any other line you specify with ONERR). Make sure, however, that the line specified in the ONERR statement actually exists in your program. Otherwise, Applesoft searches for an undefined line when an error happens, causing another error. The result is an endless loop and a locked-up computer.

You should put an error-handling routine starting at the line number referred to by ONERR. This routine should PEEK location 222, which contains an error code. If this location contains 255, then CTRL-C was pressed. The best way to deal with CTRL-C is to have your error routine GOTO the program's main menu or some other predictable location, so that CTRL-C still causes a break but doesn't stop the program.

If PEEK(222) isn't 255, then CTRL-C wasn't *pressed—an actual error occurred. This could be a disk error (wrong disk in the drive, no disk, disk full, etc.) or an error in your program. It is usually easier to let Applesoft handle the errors that you aren't expecting. You can do this by POKEing memory location 216 with 0 to cancel the ONERR trap. Then use the Applesoft RESUME instruction, which re-executes the statement that caused the error in the first place. Since the instruction didn't finish the first time, you should get the same error, but this time the program halts with an appropriate error message.*

TI Supplies

Just after I purchased a TI-99/4A computer, the company went out of business. Does this mean I won't be able to purchase anything for my computer? I would like to purchase Extended BASIC, a printer, and other peripherals.

Kathy Armstrong

Texas Instruments is still very much in business; it has simply stopped manufacturing home computers such as the TI-99/4A. Fortunately, TI-99/4A products are still available. The following firms carry software, hardware, and peripherals (this is the most complete and accurate list we were able to compile at time of publication):

Triton Products
P.O. Box 8123
San Francisco, CA 94128
1-800-227-6900

Unisource Electronics, Inc.
P.O. Box 64240
Lubbock, TX 79464
1-800-858-4580

MSW Computers & Electronics
22 East Tioga Street
Tunkahannock, PA 18657
1-800-233-3266

Tenex Computer Express
P.O. Box 6578
South Bend, IN 46660
219-259-7051

Reader Cynthia Becker informs us that hardware and software are also available through the TI-99/4A National Assistance Group. After paying a $10 membership fee, you are entitled to purchase TI products from this organization and receive its newsletter as well:

TI-99/4A National Assistance Group
P.O. Box 290812
Ft. Lauderdale, Florida 33329
(305) 583-0467

Commodore 16 Conversions

I have found that programs written for the VIC-20 Super Expander will run on the Commodore 16 as well if you add the BASIC 3.5 statement SCALE 1=1023*1023 to the beginning of the program. The 16 uses different tokens for graphics keywords like DRAW, POINT, and so on. But the programs will load without any problem from disk or tape. After you load the program, edit the lines that contain those keywords and save it again. It should run just fine.

John Elliot

Thanks for the information.

Trapping IBM's Break Key

I own an IBM PC and have been trying to trap the Ctrl-Brk keys. I have looked in a tremendous number of books, but still couldn't find anything about it. I haven't been able to scan the keyboard for the information I need. How can I trap those keys?

Patrick McGarry

Since many readers have asked this question, we'll show you two techniques that work with BASICA or Cartridge BASIC on either the PC or PCjr. The following program traps both Ctrl-Break (break) and Ctrl-Alt-Del (reboot).

CLOSE ENCOUNTERS OF THE FANTASY KIND

QUESTRON.™ *A role-playing game so spell-binding, it suspends reality and conjures up an enchanted world whose only hope against the forces of evil is you.*

GEMSTONE WARRIOR.™ *A lightning-fast action strategy game that will leave you breathless with excitement as it takes you through the labyrinths of the Netherworld on a noble quest.*

Open them up and step through the gateway to your fantasies come true.
At your local computer/software or game store today.

QUESTRON™ is available on 48K diskette for APPLE®II with Applesoft ROM, Apple II+, IIe, and IIc ($49.95). On 40K diskette for ATARI® home computers ($49.95). On 64K diskette for COMMODORE 64™ ($39.95).

GEMSTONE WARRIOR™ is available on 48K diskette for APPLE®II with Applesoft ROM, Apple II+, IIe, and IIc ($34.95). Also on 64K diskette for COMMODORE 64™ ($34.95).

If there are no convenient stores near you, VISA & M/C holders can order direct by calling 800-227-1617, ext. 335 (toll free). In California, 800-772-3545, ext. 335. Please specify computer format and add $2.00 for shipping and handling.

To order by mail, send your check to: STRATEGIC SIMULATIONS INC, 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043. (California residents, add 7% sales tax.) All our games carry a "14-day satisfaction or your money back" guarantee.

WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES.

```
10 CLS:PRINT "Try to use Brea
   k or Ctrl-Alt-Del"
20 B$=CHR$(4)+CHR$(70):C$=CHR
   $(12)+CHR$(83)
30 KEY 15,B$:KEY (15) ON:ON K
   EY (15) GOSUB 80
40 KEY 16,C$:KEY (16) ON:ON K
   EY (16) GOSUB 90
50 FOR J=1 TO 9999:NEXT:PRINT
    "Break & Ctrl-Alt-Del wor
   k now"
60 KEY (15) OFF:KEY (16) OFF
70 GOTO 70
80 PRINT "Break has no effect
    right now.":RETURN
90 PRINT "Rebooting is a very
    bad idea.":RETURN
```

Once the key trap is set (lines 20–40 above), the system checks for a trap between every statement of the main program. When the right keys are pressed, execution diverts immediately to the trapping subroutine, no matter what the main program is doing at the time. Since the trap can be sprung between any two statements in the program, strange results may occur if you don't anticipate the possible diversion. Of course, the trapping subroutine doesn't have to print a message (or do anything else except end with RETURN). You can also disable Break by changing the computer's break interrupt vector at locations 108–112 (&H6C–&H6F), as shown here:

```
10 DEF SEG=0:FOR J=0 TO 3:A(J
   )=PEEK(108+J):NEXT
20 POKE 108,64:POKE 109,1:POK
   E 110,112:POKE 111,0
30 PRINT "Try to use Ctrl-Brk
    (PC) or Fn-Brk (PCjr)
40 FOR J=1 TO 9999:NEXT:PRINT
    "Brk key works again"
50 FOR J=0 TO 3:POKE 108+J,A(
   J):NEXT
60 GOTO 60
```

This program diverts the system's normal break routine to a do-nothing IRET (return) instruction in ROM (Read Only Memory). Don't forget to restore the normal vector when the program ends (line 50). These examples are drawn from Russ Davies' Mapping the IBM PC and PCjr *(published by COMPUTE! Books), which contains additional information on keyboard programming from DOS and machine language.*

Commodore ML Addresses

I own a Commodore 64. How can I find the beginning and ending addresses of a machine language program stored on disk?

Eric Adams

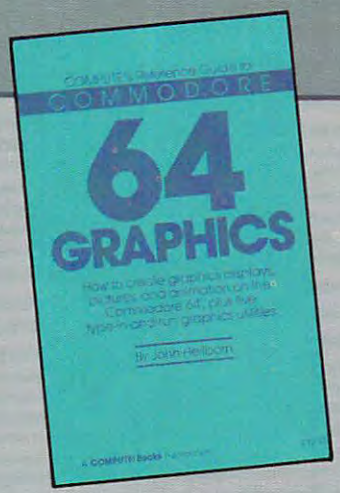
The following program does the job on any Commodore computer with a disk drive (except the 128 in CP/M mode). The first two bytes of a disk program file contain the load address in low byte/high byte format. This program finds the beginning, then reads to the end of the file. The end address equals the start address plus the number of bytes read. (Of course, a disk data *file—which holds data rather than a program—has no load address.)*

```
1 INPUT"FILENAME";F$:A$="0:"+F
  $+",P,R":OPEN 2,8,2,A$
2 GET#2,A$:GOSUB 5:L=A:GET#2,A
  $:GOSUB 5:SA=L+256*A:PRINT"S
  TART";SA
3 GET#2,A$:IF ST=0 THEN SA=SA+
  1:GOTO 3
4 PRINT"END";SA:CLOSE 2:END
5 IF A$="" THEN A$=CHR$(0)
6 A=ASC(A$):RETURN
```

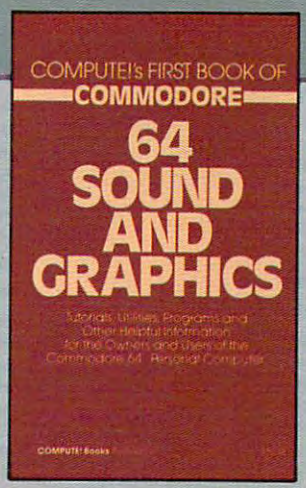
Tape users can find beginning and ending addresses with only two program lines. The following routine runs as listed on the Commodore 64, VIC-20, and PET. Plus/4 and 16 users should subtract 10 from the four addresses in line 2 (replace 829 with 819, 830 with 820, and so on). Commodore 128 users (in 128 mode) should replace the same four addresses with 2817, 2818, 2819, and 2820. The header data stored at the beginning of a tape file contains the program's starting and ending addresses. The method shown here simply OPENs the file to read the header into the tape buffer, then PEEKs the addresses from the buffer.

```
1 INPUT"FILENAME";F$:OPEN 2,1,
  0,A$:CLOSE 2
2 PRINT"START";PEEK(829)+256*P
  EEK(830);CHR$(13);"END";PEEK
  (831)+256*PEEK(832)
```

GREAT
Sound and Graphics



COMPUTE!'s Reference Guide to Commodore 64 Graphics, John Heilborn, $12.95
A complete tutorial on creating graphics displays, pictures, and animation on the Commodore 64.



COMPUTE!'s First Book of Commodore 64 Sound and Graphics, Edited, $12.95
Clear, useful explanations of the 64's sound and graphics capabilities including tutorials and example programs.

COMPUTE!'s Beginner's Guide to Commodore 64 Sound, John Heilborn, $12.95
Complete descriptions and numerous examples show you how to program sound and music on the 64.

COMPUTE! Books offers you three fascinating, informative books on sound and graphics for your COMMODORE 64 at a special savings.

Buy all three books for only $29.00. That's 25 percent off the retail price!

Limited Time Only

These books give you up-to-date, easy-to-understand tutorials, programs, and examples to help you maximize the sound and graphics features on your Commodore 64. Order today to receive all three books for only $29.00. This offer expires October 10, 1985.

Order directly from COMPUTE! Books by mailing the attached form with your payment to COMPUTE! Books, P.O. Box 5058, Greensboro, NC 27403. For the fastest service, call toll-free 800-334-0868 (in NC call 919-275-9809).

Order Form

Please send me:

____ *COMPUTE!'s Beginner's Guide to Commodore 64 Sound* (54-X), $12.95 _____
____ *COMPUTE!'s First Book of Commodore 64 Sound and Graphics* (21-3), $12.95 _____
____ *COMPUTE!'s Reference Guide to Commodore 64 Graphics* (29-9), $12.95 _____
____ SPECIAL SAVINGS—ALL THREE BOOKS FOR $29.00 _____

Subtotal _____
NC residents add 4.5% tax _____
Shipping charges* _____

*($2.00 per book in U.S. and surface mail; outside U.S. $5.00 airmail per book)

Total paid _____

ALL ORDERS MUST BE PREPAID IN U.S. FUNDS

☐ Payment enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ Visa ☐ American Express

Account No. _____ Exp. Date _____ (Required)

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

Please allow 4–6 weeks for delivery 7591171

COMPUTE! Books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England.

The AMIGA: An In-Depth Review

Tom R. Halfhill, Editor

Three years in the making, Commodore's new Amiga personal computer was finally introduced at a lavish media event in New York this summer. Commodore says the new machine should be available by the end of August. This report was compiled from sessions with the Amiga prior to its release.

Commodore's Amiga is much more than just another new computer. It's a pivotal machine that may well shatter the traditional boundaries and prejudices which for years have divided the microcomputer marketplace. It defies classification as simply a home computer, game computer, business computer, or hacker's computer. In fact, the Amiga's power, versatility, and ease of use may qualify it as the first true personal computer.

The Amiga is not a me-too clone, or a cautious step sideways, or an incremental step forward. It's a genuine leap to a new generation of advanced personal computers. The Amiga will be the yardstick by which all other new computers over the next few years will be measured.

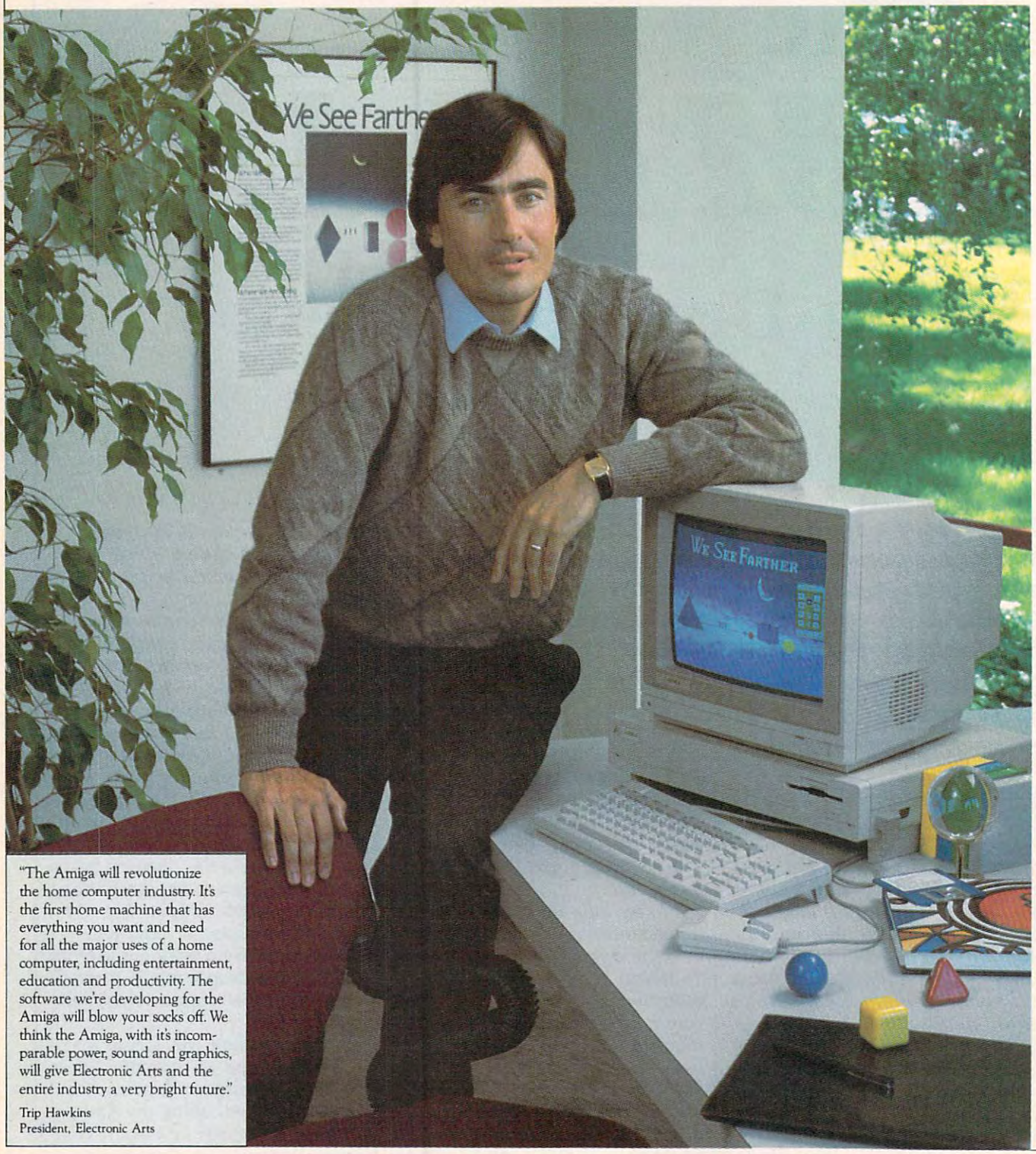
What sets the Amiga apart is that no other computer on the market can do so many things so well. To match its power as a business computer, you'd have to go all the way to a $4,000 IBM AT or even a minicomputer; to surpass its graphics and animation capabilities, you'd have to invest in a $10,000 dedicated graphics terminal; to surpass its sound and music features, you'd have to buy a music synthesizer. The Amiga is that rare example of a general-purpose machine that excels at specialized applications.

This versatility transcends the traditional computer categories taken for granted over the years. For example, although it's certainly possible to use a machine such as a Commodore 64 as a business computer, or a machine such as an IBM PC as a home computer, some compromises are usually inevitable. But the Amiga should prove to be equally suitable for the most demanding business people, home users, programmers, educators, children, video artists, and electronic musicians. In addition, it's easy enough for a beginner to learn quickly, yet deep enough to fascinate the most impassioned late-night hacker.

Commodore, too, senses that it has a new kind of computer on its hands. The company is going out of its way to avoid calling the Amiga a business computer or a home computer. Furthermore, Commodore is

A message from a leading software publisher.

WHY ELECTRONIC ARTS



"The Amiga will revolutionize the home computer industry. It's the first home machine that has everything you want and need for all the major uses of a home computer, including entertainment, education and productivity. The software we're developing for the Amiga will blow your socks off. We think the Amiga, with it's incomparable power, sound and graphics, will give Electronic Arts and the entire industry a very bright future."

Trip Hawkins
President, Electronic Arts

IS COMMITTED TO THE AMIGA.

In our first two years, Electronic Arts has emerged as a leader of the home software business. We have won the most product quality awards—over 60. We have placed the most *Billboard* Top 20 titles—12. We have also been consistently profitable in an industry beset by losses and disappointments.

Why, then, is Electronic Arts banking its hard won gains on an unproven new computer like the Amiga?

The Vision of Electronic Arts.

We believe that one day soon the home computer will be as important as radio, stereo and television are today.

These electronic marvels are significant because they bring faraway places and experiences right into your home. Today, from your living room you can watch a championship basketball game, see Christopher Columbus sail to the New World, or watch a futuristic spaceship battle.

The computer promises to let you do much more. Because it is interactive you get to participate. For example, you can play in that basketball game instead of just watching. You can actually be Christopher Columbus and feel firsthand what he felt when he sighted the New World. And you can step inside the cockpit of your own spaceship.

But so far, the computer's promise has been hard to see. Software has been severely limited by the abstract, blocky shapes and rinky-dink sound reproduction of most home computers. Only a handful of pioneers have been able to appreciate the possibilities. But then, popular opinion once held that television was only useful for civil defense communications.

A Promise of Artistry.

The Amiga is advancing our medium on all fronts. For the first time, a personal computer is providing the visual and aural quality our sophisticated eyes and ears demand. Compared to the Amiga, using some other home computers is like watching black and white television with the sound turned off.

The first Amiga software products from Electronic Arts are near completion. We suspect you'll be hearing a lot about them. Some of them are games like you've never seen before, that get more out of a computer than other games ever have. Others are harder to categorize, and we like that.

For the first time, software developers have the tools they need to fulfill the promise of home computing.

Two years ago, we said, "We See Farther." Now Farther is here.

ELECTRONIC ARTS™

Dr J and Larry Bird Go One-On-One

The number one software sports game of all time. Shoot as accurately as Larry Bird, slam dunk like the Doctor, while you're cheered on by the victory chants of the Boston Garden crowd.

Skyfox™

Get in the spaceship and fly. Out your window or on your radar screen you have but split-seconds to appreciate the fierce beauty of enemy jets and tanks.

Seven Cities of Gold™

Be Christopher Columbus and discover the New World. Learn history and geography, or generate your own random new worlds to explore.

Archon

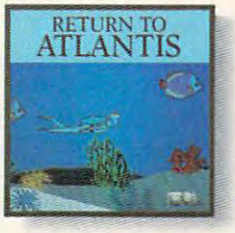
A new kind of computerized board game, like chess with wizards and dragons for pieces. But when one lands on another, they have to fight a white-knuckled action battle.

Deluxe Video Construction Set™

Be your own video director for business presentations or just for fun. Set up special effects, animated computer graphics, sound effects and titles – even record them to videotape for use with a VCR.

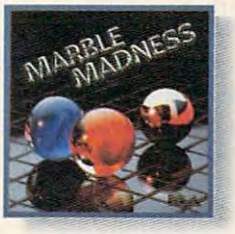
Arcticfox™

You command the advanced and deadly tank of the future – the Arcticfox. A first person tank combat game with all the stunning graphics and sound of the best 3-D simulations.



Return to Atlantis™

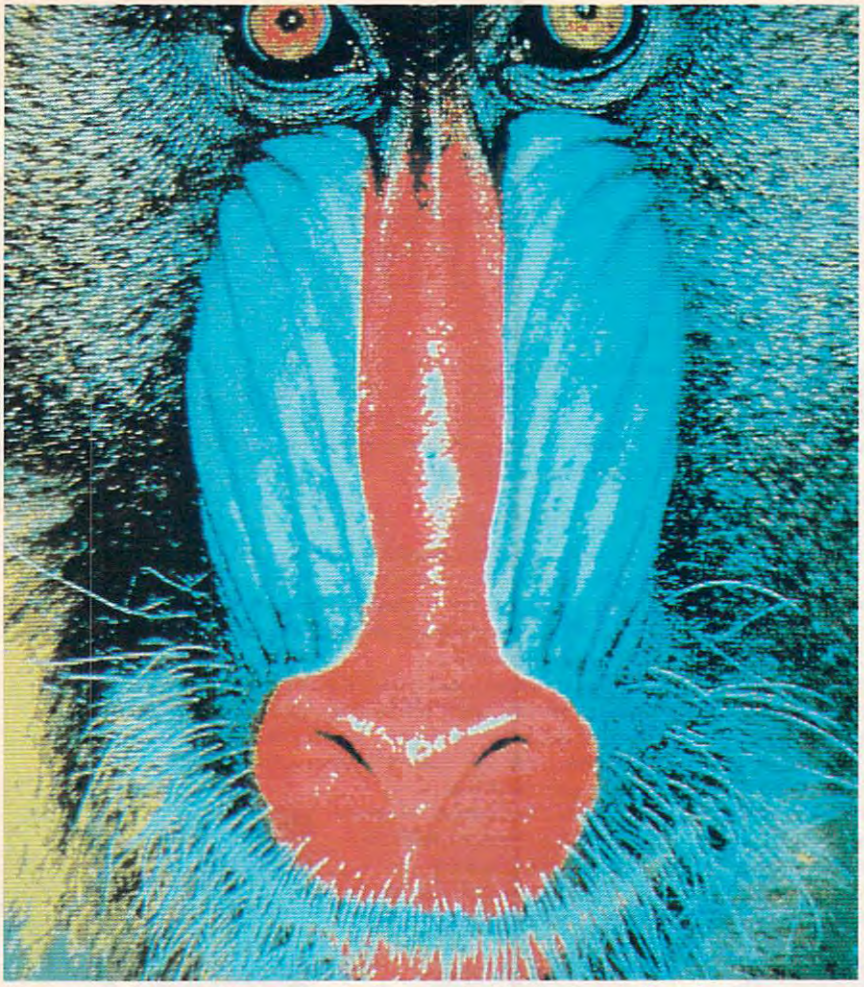
Play Indiana Cousteau, oceanic hero, in this three dimensional simulation under the seven seas.



Marble Madness™

For the first time, the home version of a coin-op arcade game is just as good as the original. Same graphics. Same sound. And you can play it in your bathrobe.

For details about availability, see your Amiga software dealer or call us at (415) 572-ARTS. For a product catalog send $.50 and a stamped, self-addressed envelope to: Electronic Arts, Amiga Catalog Offer, 2755 Campus Drive, San Mateo, CA 94403. Amiga is a trademark of Commodore Business Machines. Skyfox, Seven Cities of Gold, Deluxe Video Construction Set, Arcticfox, Return to Atlantis and Electronic Arts are trademarks of Electronic Arts. Marble Madness is a trademark of Atari Games, Inc.



High-resolution graphics on the Amiga are startlingly close to broadcast-quality TV pictures. This image of a mandrill was digitized directly from a photograph and reproduced on the Amiga's 640 × 400-pixel screen.

trying to disassociate the Amiga from its earlier line. The label on the computer, peripherals, and company-branded software says "Amiga," not "Commodore"; and one Commodore executive has asked writers to refer to the computer as the "Amiga from Commodore" rather than the "Commodore Amiga." Apparently, Commodore doesn't want potential buyers to prejudge the Amiga by Commodore's previous products. Although the best-selling VIC-20 and Commodore 64 have earned well-deserved reputations as powerful computers for the price, they are dismissed by some as "game computers" or "toy computers." But now there's an under-$1,500 personal computer which can comfortably outperform much more expensive business computers as well as the best arcade machines.

More than old technology may be rendered obsolete by computers like the Amiga. The new generation may also change a lot of old-fashioned thinking.

Here's a quick review of the Amiga's major features:

• Motorola 68000 chip for the central processing unit. This 16/32-bit microprocessor is also found in the Apple Macintosh and Atari ST series.

• Three special integrated chips nicknamed Portia, Daphne, and Agnes. Portia handles sound and input/output; Daphne handles the video; Agnes controls memory access and also contains two special devices, blitter and copper (short for *coprocessor*), which work together to produce stunning animation and graphics.

• 256K of Random Access Memory (RAM) standard. A clip-on memory board that hides behind a plastic cover on the front of the system unit adds another 256K; further expansion up to six megabytes (6,144K) is possible by adding boards onto the side expansion bus (see below).

• 192K of Read Only Memory (ROM) containing operating system routines. Most of the operating system, however, is loaded from disk into RAM on early model Amigas. This leaves about 130K RAM free on a 256K system. The operating system won't be burned into ROM chips until later. Commodore hasn't decided if upgrade ROMs will be available for early purchasers.

• Built-in microfloppy disk drive. This double-sided drive squeezes 880K of data on a single hardshell 3½-inch disk. Four external drives can be daisy-chained to a port on the back panel.

• Two-button mouse controller. This plugs into one of the two joystick ports on the side of the machine.

• Detached typewriter-style keyboard with separate cursor keys, numeric keypad, and ten special function keys. Interestingly, the keyboard not only returns a value when a key is pressed, but also when the key is released—a highly unusual feature. Also, Commodore says the Amiga can be operated completely from the keyboard, even if you unplug the mouse and hurl it across the room by its wire tail.

• Two-level operating system—AmigaDOS and Intuition, a Macintosh-style user interface that uses a mouse, icons, pull-down menus, screen windows, and multiple screens.

• Multitasking. The Amiga can run several application programs *simultaneously*, and AmigaDOS can even perform several DOS functions at once in different screen windows.

• Four sound channels with stereo output. The sound capabilities are the best of any personal computer available—a wide variety of musical instruments can be simulated with fidelity approaching that of professional-quality synthesizers. A pair of phono jacks on the rear panel sends two sound channels to each auxiliary input jack on your stereo, or they can be plugged into a mono sound system. There are also provisions for digital sound sampling with optional equipment.

ONLINE IN THE U.S.A.

Be anything you want on American PEOPLE/LINK, where people from across the country get together every night.

They are people with interests in sports, movies, music, games, and just meeting other people . . . people like you. Whatever your interests or fantasies, you'll find someone to share them with on PEOPLE/LINK.

See for yourself why so many men and women are getting together and partying online with us. Not screen after screen of boring data . . . just lots of fun and friendly people.

And our low, low rates won't turn your online dreams into nightmarish monthly charges. To discover our affordable videotex entertainment, call or write us today.

PEOPLE/LINK . . . the network that says "Thank You."

AMERICAN PEOPLE LINK

American PEOPLE/LINK
Arlington Ridge Office Center
3215 N. Frontage Road – Suite 1505
Arlington Heights, IL 60004

1-800-524-0100 (Toll Free) 1-312-870-5200 (Illinois)

This picture was created on the 320 × 200 graphics screen by an artist at Island Graphics, an Amiga software developer.

• Outputs for analog RGB (red-green-blue) monitors, composite color and monochrome monitors, and TV sets. Commodore is selling its own fine-pitch RGB monitor under the Amiga brand name. An RGB monitor is highly recommended for the Amiga, because the higher-resolution graphics modes exceed the capabilities of composite monitors and TVs.

• Centronics-standard parallel port for printers and other peripherals.

• RS-232 serial port for printers, modems, and other peripherals. Tecmar, Inc., of Cleveland, Ohio, is introducing a 2400 bits-per-second modem for this port.

• Expansion port that carries every line on the system bus. This port, on the right side of the system unit, is extremely versatile and will be used for memory expansion beyond 512K RAM, among other things. Tecmar is introducing a 20-megabyte hard disk drive and an expansion board that adds a battery-backed-up clock/calendar, a second RS-232 port, and up to two megabytes of RAM. Coprocessors are another possibility.

• A total of 4,096 colors, far surpassing any other personal computer on the market. Up to 16 or 32 colors can be displayed simultaneously in the standard graphics modes, and all 4,096 can be shown onscreen in a special mode called *hold and modify.*

• Graphics modes of 640×400 with 16 colors; 640×200 with 32 colors; 320×400 with 16 colors; and 320×200 with 32 colors. The screen display system bears a closer resemblance to 8-bit Atari computers than to existing Commodores—not surprising, since some of the Amiga designers were among those who built the original Atari 800 in the late 1970s. For example, a series of memory registers—not color memory—determines which colors will be selected onscreen. Among other things, that means that the 16 or 32 colors displayable in the graphics modes can be any of the 4,096 possible hues, and that changing a color register instantly changes the color of everything previously drawn in that color.

• Eight multicolor sprites. The sprites can be reused on various parts of the screen to create even more moving objects. In some ways, they resemble Atari player/missile graphics instead of Commodore 64-style sprites—they aren't square blocks, but rather tall strips which extend the full height of the screen. Unlike Atari players or Commodore sprites, however, the Amiga's sprites are 16 pixels wide and can display four colors simultaneously with resolution equivalent to the 320 × 200 mode. By overlaying sprites, up to 16 colors can be displayed per object.

• Text modes of 40, 60, or 80 columns. Actually, the Amiga has no true text modes in the conventional sense; all characters are displayed in high-resolution graphics. This makes possible a wide variety of onscreen type styles.

• Speech synthesis as a standard feature. This is simulated in software, not built into the hardware. The male voice seems to have a foreign accent and definitely sounds like a computer, but is more understandable than most speech synthesizers. English text-to-speech conversion is included.

• BASIC on disk. Two BASIC interpreters are in the final stages of development—ABasiC (Amiga BASIC) and a Microsoft BASIC which Commodore says resembles Microsoft BASIC for the Macintosh. According to Commodore, the Amiga will be shipped with the Microsoft BASIC, and ABasiC will be optional. Both are very powerful languages with support for graphics, animation, sound, operating system calls, and the Intuition user interface. Other interpreters, compilers, and assemblers (including Pascal, Forth, and C) will be available soon after the Amiga is introduced.

Although prices still haven't been firmed up at this writing, it appears the basic system unit with 256K RAM, built-in disk drive, detached keyboard, mouse controller, operating system software, and BASIC will cost $1,000 to $1,500. The same system with 512K RAM and a high-resolution RGB color monitor will cost about $2,000.

As personal computers have grown more powerful over the years, designers have wrestled with a dilemma: ease of use versus full flexibility. Beginners and casual users need a computer that's simple to learn and operate, while advanced users don't want to be bogged down with distractions.

The Amiga designers have worked out a compromise by offering an operating system that can be used both ways. With Intuition, the Macintosh-like interface, you can manipulate the system simply by pointing to menu items or icons representing the functions you want. For example, to call a disk directory on a Commodore 64, you

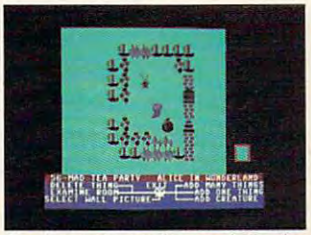
[This isn't just hype. We really do have the money.]

WANTED:

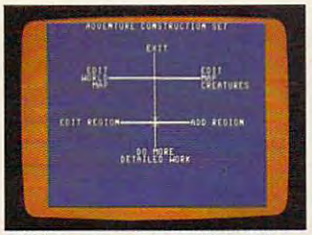
WORLD'S GREATEST ADVENTURES.

REWARD: $1000 *and* FAME. CALL (800) 227-6703.

ACS is your tool kit. More than 7000 parts to build from. Power to spare!

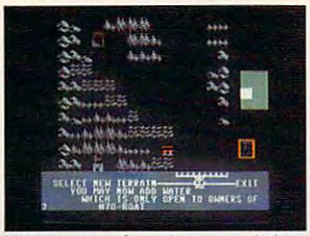


Or make your own parts from scratch. Draw 'em. Give 'em properties. Go for it!

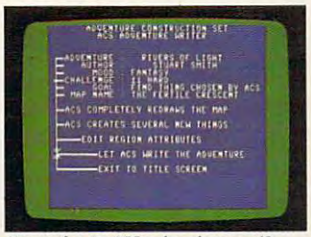


ACS is menu driven. It works with a positive "stick shift" controller. Easy as pie!

Ready-made segments of sci-fi, mystery & dragon-type adventures included. A real bargain!



Mountains, oceans, forests & etc. are pre-built. Or make your own landscapes. You're the boss!



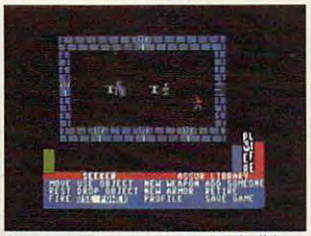
Amazing fact #1: ACS makes adventures (& finishes parts of yours) by itself. This is No Lie!

Music modules & sound effects included. Score your adventures like movies. Be a maestro!

No arms control here! Endless weapons of your design. (Endless armor & cures, fortunately.)

Complete, top-notch epic adventure included: *Rivers of Light* by Stuart Smith, author of *Ali Baba!*

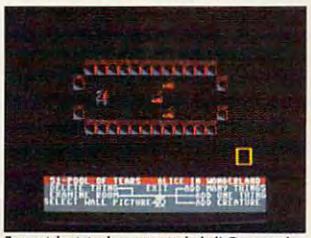
500 obnoxious creatures thrown in! Or make your own. Point 'em at the population! See their priorities!



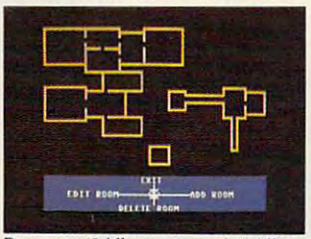
You get magic! 31 variations on 15 Big Spells! You get technology. You get religion. All right!

Amazing fact #2: ACS integrates text and graphics. All other products wimp out in this regard.

Characters? How many you want? Make 'em. Break 'em. Cast them into the world without a dime!



7 tutorial mini-adventures included! Generously survivable for those tentative first timers!



Design up to 15 different regions with 16 different rooms in each one. Space out!

When you've made a zippo-zocko GREAT ADVENTURE, send it here. You could win 1,000 smackers!

ADVENTURE CONSTRUCTION SET™

from ELECTRONIC ARTS™

HOW TO WIN: Send your adventure entries to Electronic Arts by 12/30/85. 3 winners will be selected by our judges. Their decision: final. Your adventure becomes "public domain" which means you can still get famous even if you don't get rich. **HOW TO ORDER:** Visit your retailer or call 800-227-6703 for direct VISA or MasterCard orders **(in CA call 800-632-7979).** To buy by mail, send check or money order to Electronic Arts, P.O. Box 306, Half Moon Bay, CA 94019. Add $3 to cover insured shipping and handling. Indicate desired machine version. Allow 1-4 weeks for delivery. *All EA products purchased direct have a 14 day "satisfaction or your money back" guarantee.* **PRICE & MACHINE AVAILABILITY:** C-64 $39.95. Apple IIe/c $49.95. Coming soon to IBM PC/PCjr $49.95. To receive our complete product catalog and direct order form, send 50¢ and a self-addressed, stamped envelope to Electronic Arts, Catalogue Offer, 2755 Campus Drive, San Mateo, CA 94403.

ANNOUNCING THE FAMILY SOFTWARE SPECTACULAR.™

BUY 3, GET 1 FREE.

CHOOSE FROM OVER 50 LEADING TITLES.

What do you call an offer that brings together, for the first time ever, five leading brands of home software?

The Family Software Spectacular!

And it's your opportunity to choose a *free* software program from the best titles on the market today. All you have to do is buy any combination of three titles from any participating brands during this special offer period. And we'll send you another one of your choice – *free!*

It's a one-of-a-kind value your family shouldn't miss.

FISHER-PRICE.® A NAME YOU CAN TRUST.

For years Fisher-Price has been known for helping children develop skills as they play. With Fisher-Price™ Learning Software, your child can develop skills in five key areas: math, language, creativity, basic learning and computer literacy.

Your child will love developing and testing language skills with the help of PETER RABBIT READING, a program that uses voice to teach sound and letter recognition in words.

WINDHAM CLASSICS.™ CLASSIC NOVELS COME TO LIFE.

Pick up Windham Classics, graphic interactive fiction based on some of the greatest adventures of all time.

They're great for kids, yet fun for all ages.

Enjoy "THE WIZARD OF OZ.™" You are Dorothy, and each of your decisions determines your adventure. There's more than one adventure to be had as you meet old friends and new in the land of Oz.

WORK BETTER WITH BETTER WORKING.™

Constantly erasing and retyping? Unsure of your spelling? WORD PROCESSOR WITH SPELLCHECKER is exactly what you need. This full featured word processor includes the 50,000 word AMERICAN HERITAGE DICTIONARY™ spellchecker which detects over 99% of misspelled words.

The full line also includes FILE & REPORT and SPREADSHEET – titles that represent the best values on the market today.

SPINNAKER® THE LEADER IN FAMILY LEARNING SOFTWARE.

From pre-school counting to high-school compositions, Spinnaker offers the high quality educational programs you have come to expect. For example, there's the HOMEWORK HELPER™ series which provides step-by-step methods for improving writing and math skills and helps kids tackle otherwise troublesome school assignments.

TELARIUM.™ INTERACTIVE FICTION AT ITS BEST.

Here's graphic interactive fiction by famous writers of science fiction, fantasy, and mystery like Ray Bradbury, Arthur C. Clarke and Erle Stanley Gardner. Each story unfolds with you in the center of the action – and completely in control. In PERRY MASON: THE CASE OF THE MANDARIN MURDER,™ you are the world famous criminal lawyer. Challenge the evidence. Cross-examine the witness. Only you can prove your client's innocence. And time is running out.

A SOFTWARE OFFER FOR ALL AGES.

Any software offer including something for everyone in the family is certainly fantastic. But combine that with a free product offer and what you have is something spectacular. Look for the specially marked boxes of The Family Software Spectacular at your retailer.

The Family Software Spectacular™

TURN PAGE FOR MORE INFORMATION ON THE FAMILY SOFTWARE SPECTACULAR. ⟶

have to type LOAD"$",8 and then LIST—hardly mnemonic or intuitive. But on the Amiga, you can call a directory simply by rolling the mouse to point at a disk icon; the files on the disk will appear on-screen as file folder icons. To delete a disk file, you no longer have to type OPEN15,8,15,"S0:filename":CLOSE15. Instead, you just point to a file icon and drag it into an icon of a trash can.

With Intuition, you can shrink any screen into a window and layer several such windows on the screen at once. In effect, the computer screen resembles a desktop on which papers can be shuffled around or pushed aside. Windows can be opened, closed, resized, and moved about. You can even display multiple screens on top of each other, all with their own windows.

More advanced users haven't been forgotten, however. Below this shell of windows and menus lies the core operating system, AmigaDOS—perhaps the most powerful disk operating system offered on any personal computer. It's a command-line interpreter patterned after Unix, and it also resembles PC-DOS and CP/M. A large number of advanced functions—including batch files and multitasking DOS commands—are available by typing keyboard commands at the AmigaDOS screen prompt. In fact, AmigaDOS even qualifies as a small programming language. It has commands for IF-THEN comparisons, branching to labels, and looping, so you can construct batch files to run the computer automatically.

Furthermore, AmigaDOS was designed from the ground up as a multitasking operating system. Although it is difficult to pick the Amiga's most impressive feature, multitasking is a top candidate. In effect, it's like having a mainframe computer with several terminals all to yourself. You can run several programs at once, in multiple windows and screens, without noticeably affecting performance.

For instance, you can run a word processor, spreadsheet, and database manager simultaneously, flipping between the three windows as needed. Or you can print out a document with a word processor in one window while writing another document in a different window. Or you can work on several files at once—and even several versions of the same file—by running a single application program in several windows. Programmers can test-run a program in one window while editing the code in another. Even AmigaDOS itself can be running in multiple windows, processing a number of DOS commands simultaneously.

An example of blitter animation. In this demo, the ball spins and bounces around the screen, with sound effects in stereo (see text).

The limit on this kind of multitasking depends on the complexity of the application programs and the amount of available memory. In a test using small BASIC programs, Commodore claims that AmigaDOS has handled 50 windows running 50 programs at once. After that point, they lost track of what was happening.

Part of the secret behind the Amiga's multitasking is its trio of custom chips. Like a team of busy assistants, they free the 68000 microprocessor for more important jobs, sometimes to a startling degree. For instance, a graphics demo on the Amiga features a bouncing ball (see photo). The large checkered ball rotates on its axis in simulated 3-D while bouncing off the bottom and sides of the screen; the shadow it casts is transparent, partially obscuring the background text over which it passes; and bouncing sounds echo realistically from the left and right stereo speakers each time the ball hits a surface. Yet, while all this is happening, the 68000 is doing nothing but calculating the bounce angles, working at only 8 percent capacity.

The blitter and copper are capable of cartoon-quality animation.

Another low-resolution screen created by Island Graphics. The artist used GraphiCraft, *a drawing program designed by the company that will be sold under the Amiga brand name.*

In fact, blitter animation is so good that Commodore hardly talks about the Amiga's sprite graphics. The blitter can move a screen object of any size, shape, and color at least as fast as a sprite. It even has such sprite-like features as proximity detection and display priorities. One Amiga demo shows a futuristic street scene with moving objects passing behind and in front of each other on five levels—all without sprites. If you do choose to write a program with sprites and use up all eight, the blitter can simulate extra sprites to give you as many independent objects as you want.

Another fascinating feature of the Amiga is its ability to superimpose multiple screens, referred to as *playfields*. You can think of a playfield as a giant sprite that covers the entire screen. By cutting holes in the playfield, you can see the other playfield which lies below it. Each playfield can be independently scrolled vertically and horizontally. In combination with sprites and blitter objects, this feature could lead to incredible 3-D games and other graphics effects. Intuition uses playfields to let you slide one screen away to reveal another beneath it, like a sliding chalkboard.

Even more interesting things become possible when you add an optional video board (about $200). This lets you feed standard video signals into the Amiga and mix them with graphics. The video signals can originate from a video camera, videocassette recorder, laserdisc player, TV receiver with video output, or another computer. Island Graphics of Sausalito, California, which is developing graphics software for the Amiga, used video mixing to reproduce the

This Edgar Degas painting was carefully copied onto the Amiga's low-resolution screen by Island Graphics (see text). Although the 320 × 200 resolution in this mode is no greater than that found on today's home computers, the Amiga's extensive color palette allows it to do more justice to the original.

Degas painting seen in the accompanying screen photo. First, the painting was displayed onscreen as a video image; next, a drawing program was superimposed; then, pixel by pixel, an artist traced the image in computer graphics by manipulating the mouse.

When the optional video board is finished, this process will be automated by a feature called the *frame grabber*. As the term implies, the frame grabber can digitize an incoming video image automatically. You could capture any scene with a video camera, digitize it, modify it with a graphics program if desired, and then dump the image to a graphics printer. The Diablo color inkjet printer, with an Amiga printer driver, can closely reproduce any Amiga screen. We've also heard that work is underway on a laser printer capable of reproducing any screen image in color.

Equally remarkable are the Amiga's sound capabilities. On most computers, four sound channels mean you're limited to four-part harmony or four-note chords. But because the Amiga creates sounds by simulating complex waveforms, it can play chords using only one sound channel. As a result, the Amiga can simulate a wide variety of musical instruments, often with uncanny realism. We've experimented with pipe organ sounds that would grace a cathedral, drum sounds that could hammer out a hot rap rhythm, and heavy-metal electric guitar chords that could blow you out of the room.

The sound demo program we used lets you tinker with the synthesized instruments merely by pulling down menus and selecting options with the mouse. No PEEKs, POKEs, programming skills, or computer knowledge is required. For instance, one menu contained parameters for the sound envelopes, such as attack, decay, sustain, and release. Submenus for each parameter presented such choices as "very slow" to "very fast." By readjusting the electric guitar envelope for a very slow attack and very fast release, we created a backwards guitar sound reminiscent of 1960s records by Jimi Hendrix or the Beatles.

On other computers, custom sounds can only be created by laborious programming. But with an optional accessory (price unannounced), the Amiga provides a shortcut—digital sound sampling. Just as the frame grabber lets you digitize a picture, sampling lets you capture and digitize any sound fed into the Amiga from an outside source. Want to simulate a saxophone? Just play a sax into a sound system that's plugged into the Amiga, or even hook up your stereo to the computer and pipe in some music from a favorite record, tape, or compact disc. We've also heard demos of digitally sampled speech—not to be confused with synthesized speech—that sound as good as tape recordings.

Commodore says several companies are working on music keyboards that will turn the Amiga into a full-blown synthesizer. By using the computer's memory as a sequencer, the Amiga could become a multitrack recording studio for the additional cost of only a few hundred dollars.

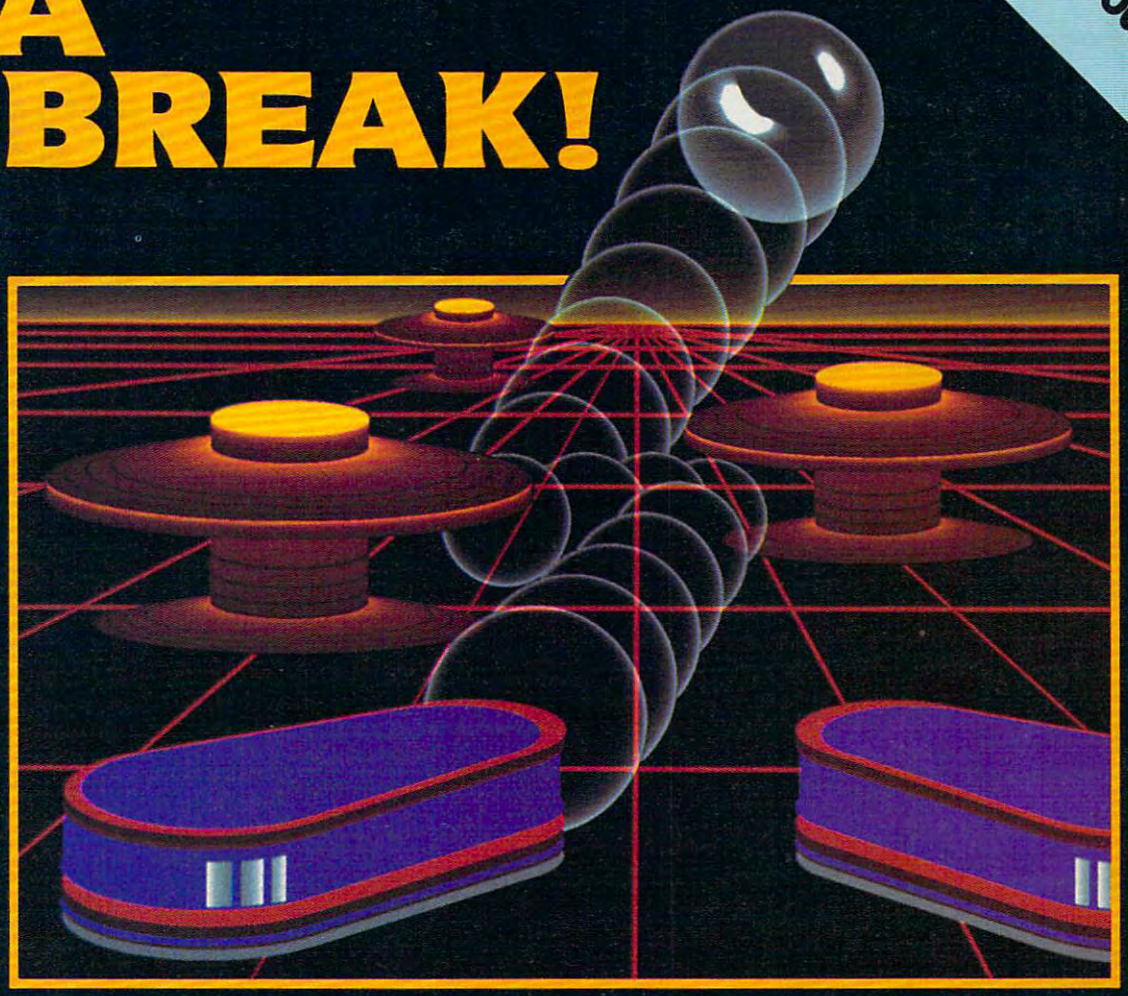
This report only scratches the surface. A complete set of technical manuals for the Amiga resembles a stack of Manhattan phone books—it will be months, perhaps years, before they're fully explored by programmers and software manufacturers. People are still developing new techniques on computers which have been available for years, and the Amiga is a whole order of magnitude more advanced.

A significant number of companies are now programming for the Amiga, and it appears that about two dozen packages will be available around the time the computer hits the stores. These include everything from word processors to business-graphics programs to games.

Looking toward the future, Commodore says this computer is just the first in a series of Amigas, and that this one represents the low end. What's to follow? Commodore isn't saying. Perhaps the best thing about the Amiga is that it stretches our imaginations a little bit further. ©

TAKE A BREAK!

For Apple, Atari®, & Commodore 64™



WITH NIGHT MISSION PINBALL

You deserve the best. You've earned it. Now reward yourself with a session of **Night Mission PINBALL**, the most realistic and challenging arcade simulation ever conceived! ■ Stunning graphics and dazzling sound effects put **Night Mission PINBALL** in a class by itself. Game features: multi-ball and multi-player capabilities, ten different professionally designed levels of play, and an editor that lets you create *your own* custom modes. ■ So take a break with **Night Mission PINBALL** from SubLOGIC. Winner of *Electronic Games* magazine's 1983 Arcade Award for Best Computer Audio/Visual Effects.

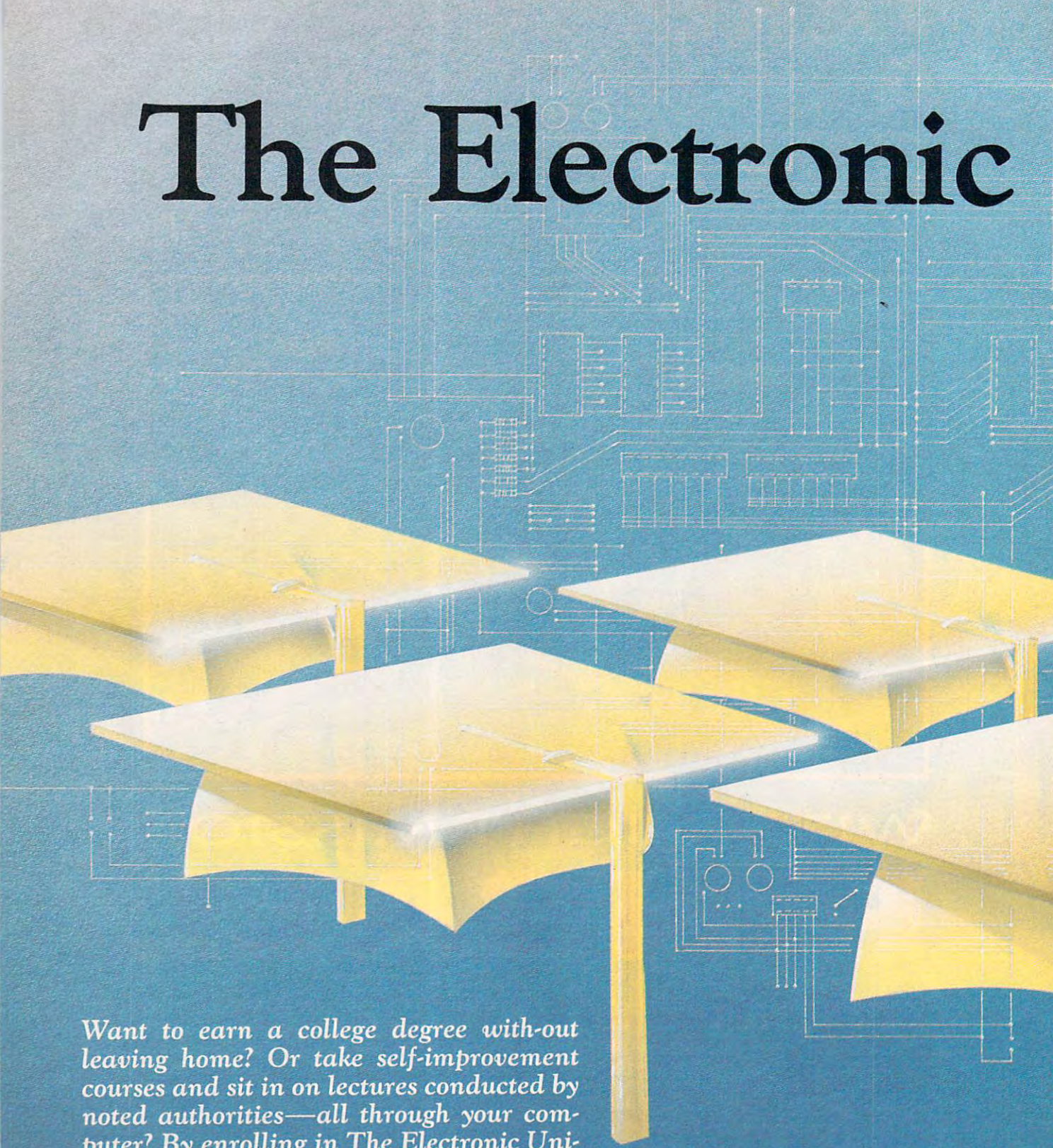
See your dealer . . .

or write or call for more information.

Order Line: 800 / 637-4983

subLOGIC Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

The Electronic



Want to earn a college degree with-out leaving home? Or take self-improvement courses and sit in on lectures conducted by noted authorities—all through your computer? By enrolling in The Electronic University—a computerized correspondence school—you can do all that, and much more.

University

Sharon Darling

Leeann Pearce calls The Electronic University a "miracle." As she sits at the Commodore SX-64 in her home in West Des Moines, Iowa, Pearce is working toward a degree in computer technology to be granted by Thomas A. Edison College in New Jersey. Although she lives a thousand miles away and suffers from multiple sclerosis, Pearce is gaining the benefits of a college education by using an online educational system designed to work with home computers. Her husband, Frank, is using the same system after he comes home from work at night to earn a master's degree in business. And their eight-year-old daughter, Katie, who used to have trouble with math in school, has boosted her grades by taking an online math tutoring class. Katie is also halfway through a computer programming course and is registering for a class in literary arts this fall. One of the family's biggest problems now is arranging schedules so that each has enough time with the computer.

ble to attending local colleges. But what really made the difference was the ability to take courses without leaving home. Because classes proceed at the student's own pace, Pearce was able to undergo surgery six months ago without interrupting her coursework. And academically, they find the classes as worthwhile as those taken the traditional way.

"I would say the courses are challenging enough," says Pearce. "They're like peanuts—you keep wanting to come back for more. And to bat around ideas with a Ph.D. is really wonderful to me."

What began as a project to teach people how to use modems has grown into a telecommunications network which allows students to use computers to earn high school and college degrees, take noncredit self-improvement courses, and "attend" seminars conducted by noted authorities. Graduate degrees in business administration have even been added to The Electronic University, which was developed by TeleLearning Systems, Inc. of San Francisco, a company founded in 1983 by entrepreneur Ron Gordon.

Close to 15,000 students are now taking classes and seminars in subjects ranging from economics to the subtleties of California wines. And the number of colleges and universities participating in The Electronic University has topped 1,700—all of which offer credit for courses taken through EU. Among the major institutions participating in EU are Cornell University, American University in Washington, D.C., Boston University, Virginia Tech, the New York Institute of Technology, Brigham Young University, the California State University system, the State University of New York, and many other state university systems. If enough coursework is completed to obtain a degree, the diploma is issued by the participating institution, not EU. It's up to students to make sure they meet the requirements of the college from which they want to receive the credit. EU has counseling services, however, to guide students through a degree program.

Close to 15,000 students are now taking classes and seminars in subjects ranging from economics to the subtleties of California wines. And the number of colleges and universities participating in The Electronic University has topped 1,700.

All it takes to enroll in EU is a computer (the system is compatible with the Commodore 64, IBM PC/PCjr, and Apple II series), a modem, and an enrollment package from EU. The package, a one-time investment, costs $79.95 for the Commodore 64 and $149.95 for Apple and IBM computers. If you don't own a modem, TeleLearning will sell you one for about $100.

Tuition ranges from $12 for a seminar up to $295 for some courses leading to a degree. In addition, students pay connect-time fees to participate in seminars and to access the more than 60 online databases. These fees range from about 17 to 80 cents per minute, depending on which database is accessed and when the call is placed. (A $15 monthly minimum is required.) To avoid long-distance charges, the phone calls are made to a local network number.

EU offers seven degree programs, including associate degrees in science, management, and the arts; bachelor's degrees in business administration and the arts; and three master of business administration (MBA) degrees—a general MBA and two specialized MBAs in technology/engineering management and individual financial planning.

Courses for college credit and self-improvement aren't the only services available. The enrollment package also offers tutoring programs for children, an electronic library with more than eight million books, counseling services, and courses in business and professional skills. Once students receive the enrollment package, they can sign up for whatever services they want. Credit courses begin every 60 days.

After students register, they're mailed an information packet on the courses they selected. The packet includes assignment outlines, a list of textbooks and other required materials, and the procedures of the institution delivering the course.

Students also receive a floppy disk containing a general introduction and a series of lessons. A typical lesson might include onscreen instruction, a textbook reading assignment, or other outside activities assigned by the instructor. Periodically, students must use their computer to transmit a progress report to their instructor via electronic mail (E-mail). They can also send questions about the course material and receive answers from the instructor by E-mail. Instructors respond to E-mail messages within 24 hours. In addition, students can schedule an online conference with the professor during designated office hours.

Some courses feature online exchanges with the instructor and even electronic forums with other students—a kind of class discussion via computer. Seminars also employ realtime conferences. Students sign on with their computers at the appropriate time, and the entire discussion session is carried out online.

Roughly 50 percent of a course's contents call for responses from the instructor. A typical class has 10 or 12 lessons; of those, half usually require students to write a response and send it to the instructor via modem, while the other half are "read-write" lessons. In that mode, students read material and type responses on the screen, but the results are not sent to the instructor. However, the instructor has the option of testing students on read-write material to check their progress.

Get the jump on the weather-man by accurately forecasting the local weather yourself!

A scientifically proven way to develop an awesome memory.

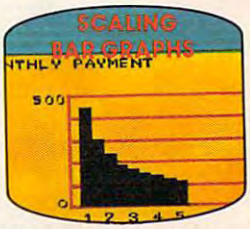
You are trapped in a five-story, 125-room structure made entirely of ice. Find the exit before you freeze!

Take control of your personal finances in less than one hour a month.

The beautiful princess is held captive by deadly dragons. Only a knight in shining armor can save her now!

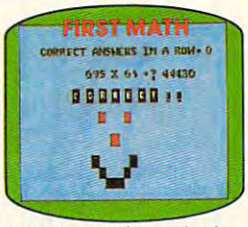
Cut your energy costs by monitoring your phone, electric and gas bills.

Computerize car maintenance to improve auto performance, economy and resale value.

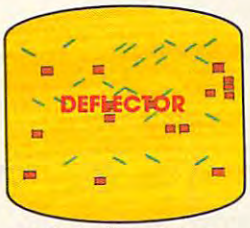


Create multi-colored bar graphs with a surprisingly small amount of memory.

A time-saving organizer for coupons, receipts and more.



School-age and pre-school children are rewarded for right answers, corrected on their wrong ones.



A real brainflexer. Deflect random balls into targets on a constantly changing playfield.

A fun way to dramatically increase typing speed and accuracy.

Get up to 30 new programs and games for less than 15 cents each— every month in COMPUTE!

Every month, COMPUTE! readers enjoy up to 30 brand-new, ready-to-run computer programs, even arcade-quality games.

And when you subscribe to COMPUTE!, you'll get them all for less than 15 cents each!

You'll find programs to help you conserve time, energy and money. Programs like Cash Flow Manager, Retirement Planner, Coupon Filer, Dynamic Bookkeeping.

You'll enjoy games like Air Defense, Boggler, Slalom, and High Speed Mazer.

Your children will find learning fast and fun with First Math, Guess That Animal, and Mystery Spell.

Looking for a challenge? You can write your own games. Customize BASIC programs. Even make beautiful computer music and pictures.

It's all in COMPUTE!. All ready to type in and run on your Atari, Apple, Commodore, PET/CBM, TI-99/4A, Radio Shack Color Computer, IBM PC or IBM PCjr.

What's more, you get information-packed articles, product reviews, ideas and advice that add power and excitement to all your home computing.

And when it's time to shop for peripherals or hardware, check COMPUTE! first. Our product evaluations can save you money and costly mistakes. We'll even help you decide what to buy: Dot-matrix or daisy-wheel printer? Tape storage or disk drive? What about modems? Memory expansion kits? What's new in joysticks, paddles and track balls?

Order now! Mail the postpaid card attached to this ad and start receiving every issue of COMPUTE!.

For Faster Service Call Toll-Free 1-800-334-0868

COMPUTE! P.O. Box 914, Farmingdale, NY 11737

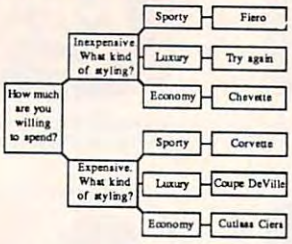
SUPER PRODUCTIVITY

POWER PLAN

Names	Wage per /hr	Mo	Tu
Andrew	6.00 $	7.50	8.10
Bosen	7.50 $	7.70	3.50
Carter	4.70 $	6.80	3.50
Damien	5.90 $	1.90	10.60
Gerrit	13.00 $	11.50	10.00
Higgins	9.10 $	6.50	7.80
Mc Donald	7.20 $	9.00	10.40
Nimitz	8.99 $	9.20	4.40
Smith 1	15.90 $	4.40	13.10
Smith 2	15.00 $	10.10	4.40
Wimpy	6.00 $	9.20	11.60
Minimum		1.90	3.50
Average		7.62	7.95
Maximum		11.50	13.10

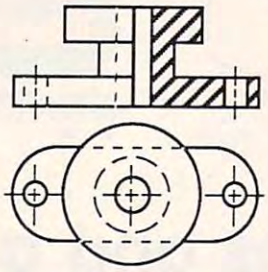
Powerful spreadsheet *plus* builtin graphics - display your important data visually as well as numerically. You'll learn fast with the 90+ HELP screens. Advanced users can use the shortcut commands. For complex spreadsheets, you can use POWER PLAN's impressive features: cell formatting, text formatting, cell protection, windowing, math functions, row and column sort, more. Then quickly display your results in graphics format in a variety of 2D and 3D charts. Includes system diskette and user's handbook. **$49.95**

XPER - expert system



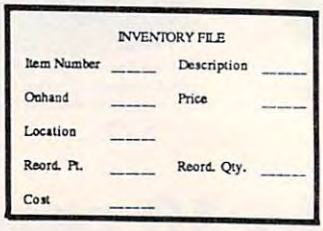
XPER is the first *expert system* - a new breed of intelligent software for the C-64 & C-128. While ordinary data base systems are good at reproducing facts, **XPER** can help you make decisions. Using its simple entry editor, you build the information into a *knowledge base*. XPER's very efficient searching techniques then guide you through even the most complex decision making criteria. Full reporting and data editing. Currently used by doctors, scientists and research professionals. **$59.95**

CADPAK Revised Version



CADPAK is a superb design and drawing tool. You can draw directly on the screen from keyboard or using optional lightpen. POINTs, LINEs, BOXes, CIRCLEs, and ELLIPSEs; fill with solids or patterns; free-hand DRAW; ZOOM-in for intricate design of small section. Mesuring and scaling aids. Exact positioning using our *AccuPoint* cursor positioning. Using the powerful **OBJECT EDITOR** you can define new fonts, furniture, circuitry, etc. Hardcopy to most printers. **$39.95**
McPen lightpen, optional **$49.95**

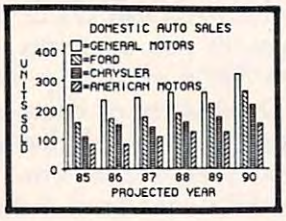
DATAMAT - data management



"Best data base manager under $50"
RUN Magazine

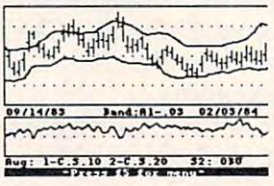
Easy-to-use, yet versatile and powerful features. Clear menus guide you from function to function. Free-form design of data base with up to 50 fields and 2000 records per diskette (space dependent). Simple data base design. Convenient and quick data entry. Full data editing capabilities. Complete reporting: sort on multiple fields and select records for printing in your specific format. **$39.95**

CHARTPAK



Make professional quality charts from your data in minutes. Quickly enter, edit, save and recall your data. Then interactively build pie, bar, line or scatter graph. You can specify scaling, labeling and positioning and watch **CHARTPAK** instantly draw the chart in any of 8 different formats. Change the format immediately and draw another chart. Incudes statistical routines for average, deviation, least squares and forecasting. Hardcopy to most printers. **$39.95**
CHARTPLOT-64 for 1520 plotter **$39.95**

TAS - technical analysis



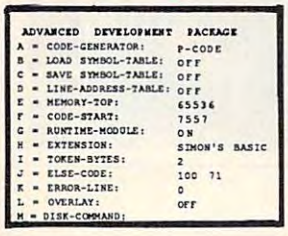
Technical analysis charting package to help the serious investor. Enter your data at keyboard or capture it through **DJN/RS** or Warner Services. Track high, low, close, volume, bid and ask. Place up to 300 periods of information for 10 different stocks on each data diskette. Build a variety of charts on the split screen combining information from 7 types of moving averages, 3 types of oscillators, trading bands, least squares, 5 different volume indicators, relative charts, much more. Hardcopy to most printers. **$84.95**

SUPER LANGUAGES

The most advanced **C** development package available for the C-64 or C-128 with very complete source editor; full K&R compiler (w/o bit fields); linker (binds up to 7 separate modules); and set of disk utilities. Very complete editor handles search/replace, 80 column display with horizontal scrolling and 41K source files. The I/O library supports standard functions like printf and fprintf. Free runtime package included. For C-64/C-128 with 1541/1571 drive. Includes system diskette and user's handbook. **$79.95**

Not just a compiler, but a complete development system. Rivals Turbo Pascal© in both speed and features. Produces fast 6510 machine code. Includes advanced source file editor; full Jensen & Wirth compiler with system programming extensions, new high speed DOS (3 times faster); builtin assembler for specialized requirements. Overlays, 11-digit arithmetic, debugging tools, graphics routines, much more. Free runtime package. Includes system diskette and complete user's handbook. **$59.95**

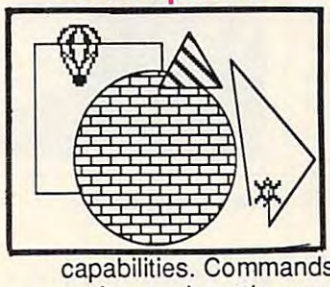
BASIC-64 full compiler



```
ADVANCED DEVELOPMENT PACKAGE
A = CODE-GENERATOR:      P-CODE
B = LOAD SYMBOL-TABLE:   OFF
C = SAVE SYMBOL-TABLE:   OFF
D = LINE-ADDRESS-TABLE:  OFF
E = MEMORY-TOP:          65536
F = CODE-START:          7557
G = RUNTIME-MODULE:      ON
H = EXTENSION:           SIMON'S BASIC
I = TOKEN-BYTES:         2
J = ELSE-CODE:           100  71
K = ERROR-LINE:          0
L = OVERLAY:             OFF
M = DISK-COMMAND:
```

The most advanced **BASIC** compiler available for the C-64. Our bestselling software product. Compiles to superfast 6510 machine code or very compact speedcode. You can even mix the two in one program. Compiles the complete BASIC language. Flexible memory management and overlay options make it perfect for all program development needs. **BASIC 64** increases the speed of your programs from 3 to 20 times. Free runtime package. Includes system diskette and user's handbook. **$39.95**

VIDEO BASIC development



The most advanced graphics development package available for the C-64. Adds dozens of powerful commands to standard BASIC so that you can use the hidden graphics and sound capabilities. Commands for hires, multicolor, sprite and turtle graphics, simple and complex music and sound, hardcopy to most printers, memory management, more. Used by professional programmers for commerical software development. Free runtime package. Includes system diskette and user's handbook. **$39.95**

FORTH Language

```
SCR # 20
 0 P ( RANDOM NUMBER TESTER ?RND )
 1 P FORTH DEFINITIONS DECIMAL
 2 P : ?RND
 3 P       ( INITIALIZE FIRST SCREEN)
 4 P       1024 1000 ASCII 0 FILL
 5 P       BEGIN
 6 P          1000 RND   ( RANDOM 0.999)
 7 P          40 /MOD    ( COLUMN, LINE)
 8 P          SWAP       ( EXCHANGE)
 9 P          2DUP 58    ( CHARACTER)
10 P          1+ -ROT    ( ADD 1)
11 P          5!         ( SAVE)
12 P       ?TERMINAL UNTIL
13 P
```

Our **FORTH** language is based on the Forth 79 standard, but also includes much of the 83 level to give you 3 times vocabulary of fig-Forth. Includes full-screen editor, complete Forth-style assembler, set of programming tools and numerous sample programs to get you deeply involved in the FORTH language. Our enhanced vocabulary supports both hires and lores graphics and the sound synthesizer. Includes system diskette with sample programs and user's handbook. **$39.95**

Other software also available!
Call now for free catalog and the name of your nearest dealer. Phone: **616/241-5510.**

Abacus Software

P.O. Box 7211 Grand Rapids, MI 49510 616/241-5510

For fast service call 616/241-5510. For postage and handling, include $4.00 per order. Foreign orders include $8.00 per item. Money orders and checks in U.S. dollars only. Mastercard, Visa and Amex accepted.

Dealer Inquiries Welcome
More than 1200 dealers nationwide

EU does not administer any final exams. There is a practice exam available to students, but it doesn't replace taking a proctored exam at a nearby library or college, which is given by the school accepting the credits. Students also have the option of taking a CLEP (College Level Examination Program) test for credit, or an ACT PEP exam, which is given by the American College Testing Proficiency Examination Program.

Developing a college course to be taught by computer and keeping the material interesting is quite a challenge, says Tom Copley, an EU professor who formerly taught business courses at Antioch College in Ohio. Copley says he was "immediately intrigued" by the idea of an electronic college when he first read about TeleLearning last spring. Not only has he been a computer buff for the last 10 or 15 years, but he also has taken traditional evening school courses in the past. In addition to teaching classes, he's now deeply involved in developing courses for the online school.

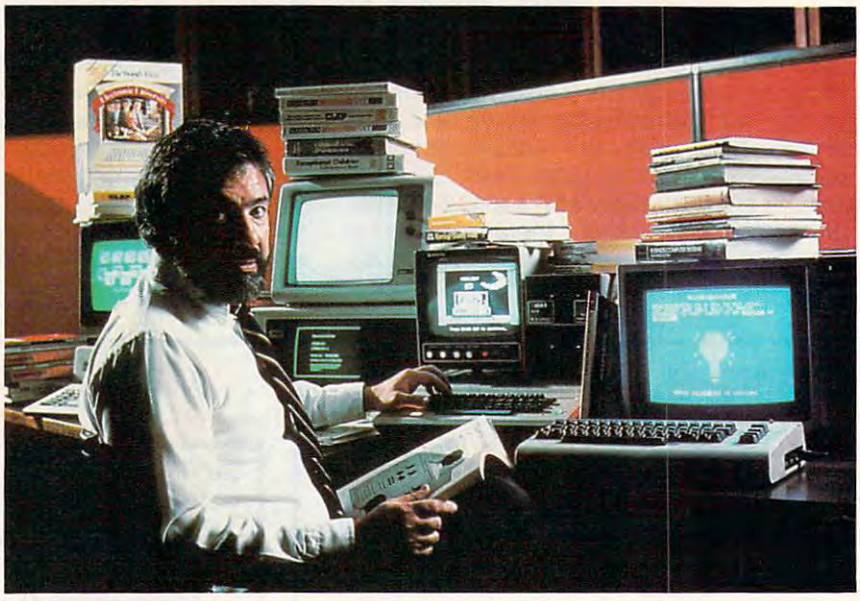
"In the first place, you're working with a totally different media, and in order to be effective, you have to take advantage of its advantages. Unfortunately, the cathode ray tube is not nearly as expressive a medium [as books or lectures]." Therefore, he says, "you have to get high learning impact in a small amount of space."

Copley tries to focus on higher-level questions, the kind in which "the student has to synthesize a lot more information and draw more conclusions. I don't find myself using typical textbook jargon—words like *describe, list, differentiate,* etc. I ask for things that require a little more creative thought."

One less obvious advantage to long-distance learning that Copley has discovered is the opportunity to respond to students on a one-to-one basis by E-mail, even though he never sees the student in person. "So often [while teaching in a traditional college], I've had to respond to so many students at once. This is the opposite extreme. Every student gets an individual response, and it's not something off the top of my head, but a thought-out response."

But there are disadvantages, too. "You lose the group dynamics of working in a class environment; some people find that very stimulating. Of course, a lot of educators are critical of the class environment. They say the students are being spoon-fed, entertained. There is none of that in this system. Alternatively, though, there are a lot of things you can do, like screen layout, to make it interesting."

Tom Copley predicts that alternatives like EU are "the wave of the future." He says the opportunity to take courses on your own time, at your own pace, and at the setting of your choice appeals to certain kinds of students, especially those in remote locations with no colleges nearby.



Ron Gordon, founder of The Electronic University.

Today's EU differs from the original focus of the university, which was to offer noncredit courses for personal improvement. After working with the U.S. Department of Education, TeleLearning realized there was an untapped market of people who could benefit from an alternative to traditional colleges.

When TeleLearning first approached universities with the idea of offering courses by computer, many professors were skeptical. Now, however, the school is gaining acceptance nationwide. By next year, founder Ron Gordon hopes to have 50,000 students enrolled. His ultimate goal is for the system to become the largest of its kind in the world, with millions of students.

EU also tends to attract older students than traditional universities. The usual emphasis on undergraduate students who are 18 to 22 years old doesn't always mesh with "people in their 30s who work maybe ten hours a day and may have a family," explains Copley. "Maybe it's been a lifelong dream of theirs to finish college, or maybe their job depends on them finishing a degree. For them, the traditional college life doesn't fit what they need. They're tired after work, or they want the flexibility they can't get from a regular university."

In the future, Copley is convinced The Electronic University will continue growing as more adults find computerized learning accessible, challenging, and rewarding. "So many marketing people focus on baby boomers, and that's where the market is—adults. And that's what undergraduate schools are finding out."

For more information about The Electronic University, contact TeleLearning Systems, Inc., 505 Beach Street, San Francisco, CA 94133. ©

THE #1 BACKUP COPY SYSTEM
IS NOW BETTER THAN EVER!

DiskMaker

Version 3.3

with MasterKey Modules

NEW! MODULAR SYSTEM

THE AUTOMATIC DISK BACKUP SYSTEM FOR COMMODORE 64 SOFTWARE

- The LATEST breakthrough in copy technology!
- Copies NEW custom G.C.R. code protection
- Accesses & copies NEW no-sync data blocks!
- Now includes NEW long data-block logic
- Copies normal & PROTECTED data up to track 40
- 3 New IMPROVED copiers! Includes FAST Copier, VARI-SECTOR Copier, and AUTO-LOGIC NIbbler
- Actually copies a FULL disk in only 2.75 minutes!
- Includes one FREE Masterkey Module!
- Inexpensive MODULE releases keep you up-to-date
- MODULES available individually or by subscription
- Works with DiskMaker TOOLKIT and FAT-TRACK Kit
- Requires standard 1541 disk drive or SX-64
- Special version available for INDUS GT drive!
- PRODUCT SUPPORT tel. # & future module updates
- You've tried the rest, now get the BEST!!

THE FASTEST NIBBLE COPIER YOU CAN BUY!

ONLY
$49.95
for copier system

To order CALL **805-687-1541 ext. 94**
Technical line 805-682-4000 ext. 55

MasterCard VISA AMERICAN EXPRESS

Order by phone 24 hrs./7 days or send cashier's check/money order payable to BASIX. Visa, MasterCard & American Express include card # and exp. date. CA residents add 6% tax. Allow 2 weeks for clearance of personal and c.o.d. checks. Add $3.50 shipping/handling for continental U.S., $5.50 for UPS air. COD's shipped to 48 states only, add $4.00. AK, HI, APO, FPO, add $7.50. CANADA add $10.00. Other Foreign orders add $15.00, and remit certified U.S. funds only. No credit card orders from outside North America. Dealers & Distributors invited & supported.

DISKMAKER...THE ONLY COPY SYSTEM YOU'LL EVER NEED!

basix

3463 State Street · Suite 454E · Santa Barbara, CA 93105

Word Search

Original Program By Michael B. Williams

This computerized puzzle-maker can provide hours of challenging fun. We've included versions for Commodore, IBM PC/PCjr, Apple II-series, TI-99/4A, and Atari computers. A printer is required.

You're probably familiar with word search puzzles: Certain words are hidden in a rectangle of nonsense letters, and it's your job to hunt them down. "Word Search" lets you create such puzzles on your computer's printer with words of your own choice. Since you design the puzzle, you can make it as easy or as difficult as you want, using up to 100 different words on some computers. Topical puzzles make the game even more interesting. For example, you might include only computer words, the names of foreign cities, or stumpers like "uxorious" and "bougainvillaea." Parents and teachers can make puzzles for children using weekly vocabulary lists.

If you're using an Atari, type in and save Program 8, then skip to the program instructions below. For other computers, we've saved space by listing Word Search in the form of one main program with separate line changes and additions for each specific machine. If you're using a Commodore, Apple, IBM PC/PCjr, or TI-99/4A, the first step is to find the specific listing for your computer. Before typing anything, cross out every line in the main program (Program 1) that has the *same* line number as a line in the listing for your computer. Then type in all the lines listed for your computer, as well as all the lines in Program 1 that haven't been crossed out.

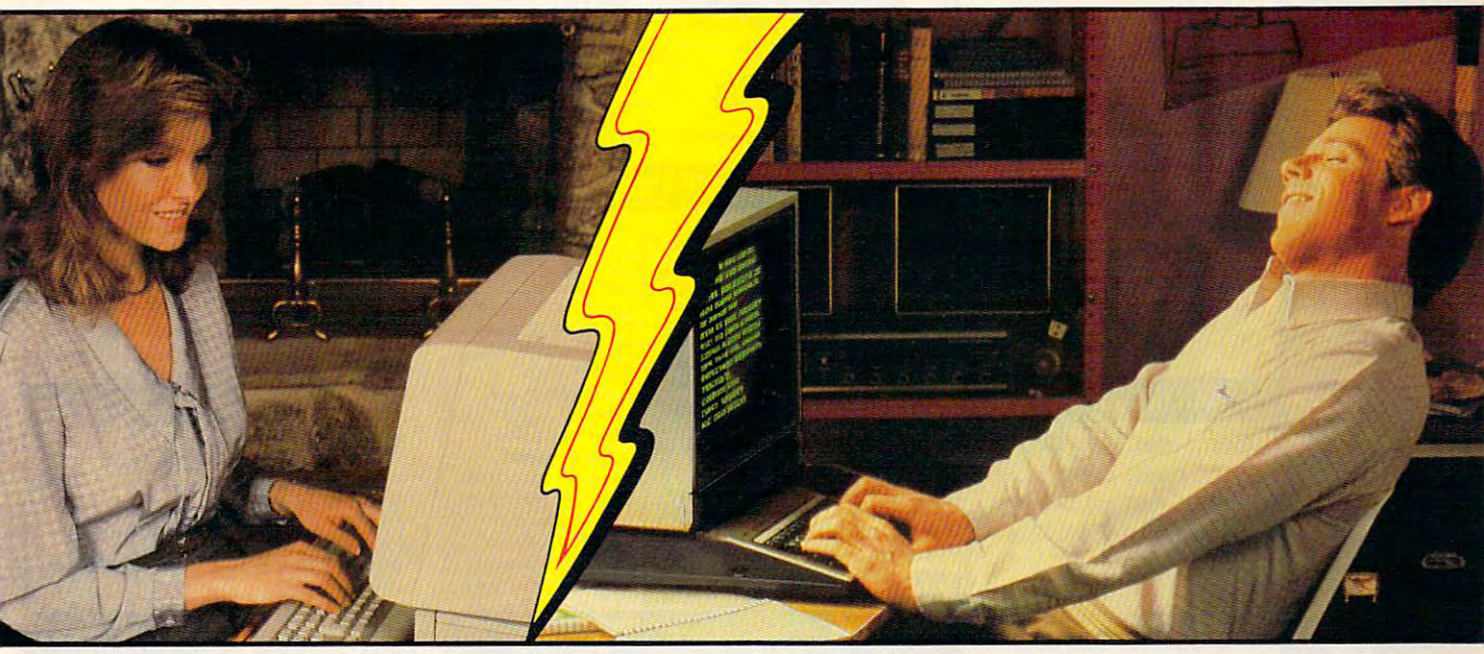
No matter which computer you're using, save a copy of Word Search and refer to the notes below before running the program. The following instructions apply to every version:

Word Search begins by asking you for the number of words to be hidden. When you've answered that question, the computer asks you to choose the number of rows and columns for the puzzle grid. Since the grid must be big enough to hide all the words, the computer tells you when you've made the grid too small and lets you try again.

Next, Word Search lets you enter the words one by one. There's no particular limit on word length, but keep in mind that the words must fit inside the grid. (For example, you can't fit a 12-letter word in a 6 × 6 grid.) Since longer words are harder to fit into the grid, the computer sorts the words by length (from longest to shortest) so it can place the longest words first. When many words are involved, this can take a few minutes, so be patient.

Once the words are sorted, you're allowed to name the puzzle. You also have the option of printing the solution to the puzzle (parents and teachers might want to separate the solution from the puzzle until the puzzle has been tried). After printing one puzzle, you can create another, using the same word list (the words will be rearranged) or entirely new words. Word Search is designed to permit a maximum of 100 words in a 99 × 99 grid (exceptions for certain computers are noted below). However, puzzles of that size can take a long time to create—over an hour in some cases. In addition, many

"Plinkers" Just Want To Have Fun.



Every night, American PEOPLE/LINK subscribers (Plinkers) go online and party with other men and women of all ages and interests from across the country and the world.

What do they do? They do what adults do. They make dates and play trivia, they tell jokes and argue, they form clubs, and even live out their fantasies. But the one thing they all have in common is that they want to have *fun*.

Our subscribers develop friendships and romances as they "talk live" (in groups or in private) and send electronic mail to each other. And our low rates will show you why Plinkers can *afford* to have so much fun.

To discover our affordable videotex entertainment, call or write us today.

PEOPLE/LINK . . . for the *fun* of it.

AMERICAN PEOPLE LINK

American PEOPLE/LINK
Arlington Ridge Office Center
3215 N. Frontage Road – Suite 1505
Arlington Heights, IL 60004

1-800-524-0100 (Toll Free) 1-312-870-5200 (Illinois)

VISA MasterCard Card

USE YOUR VISA, MASTERCARD OR AMERICAN EXPRESS.

COMPUTER MAIL ORDER

THE CMO ADVANTAGE

- THE BEST PRICES!
- Next day shipping on all in stock items.
- Free easy access order inquiry.
- Orders from outside Pennsylvania and Nevada save state sales tax.
- Free technicial support with our factory trained technical staff.
- There is no limit and no deposit on C.O.D. orders.
- There's no extra charge for using your credit card. Your card is not charged until we ship.
- No waiting period for cashiers checks.
- We accept purchase orders from qualified corporations. Subject to approval.
- Educational discounts available to qualified institutions.
- FREE CATALOG MEMBERSHIP.

ORDER LINE

1-800-233-8950
In PA 1-800-242-4215

CUSTOMER SERVICE & TECH SUPPORT
1-717-327-1450
Dept. A209

MAILING ADDRESS

Computer Mail Order
Dept. A209
477 East Third Street
Williamsport, PA 17701

MEMBER DIRECT MARKETING ASSOCIATION

CREDIT CARDS

VISA MasterCard AMERICAN EXPRESS DINERS CLUB

SHIPPING

Add 3%, minimum $5.00 shipping and handling on all orders. Larger shipments may require additional charges.

All items subject to availability and price change.

Returned shipments may be subject to a restocking fee.

CANADIAN ORDERS

1-800-268-3974
Ontario/Quebec

1-800-268-4559
Other Provinces

1-416-828-0866
In Toronto

TELEX: 06-218960

2505 Dunwin Drive,
Mississauga, Ontario
Canada L5L1T1

All prices shown are for U.S.A. orders.
Call The Canadian Office for Canadian prices.

ATARI

COMPUTER CARTRIDGES $2.99

AAB4013 ASTEROIDS
AAB8130 CAVERNS OF MARS
AAB4020 CENTIPEDE
AAB4025 DEFENDER
AAB8026 DIG DUG
AAB8031 DONKEY KONG
AAB8039 EASTERN FRONT
AAB8030 E.T.
AAB4024 GALAXIAN
AAB4012 MISSILE COMMAND
AAB4022 PAC MAN
AAB4027 QIX
AAB4008 SPACE INVADERS
AAB4011 STAR RAIDERS
AAB4006 SUPER BREAKOUT
AAB4010 3-D TIC-TAC-TOE
AAB8045 MUSIC COMPOSER
AAB8045 PENGO
AAB8029 FOOTBALL
AAB8057 TYPO ATTACK

(Repackaged without instructions)

COMPUTERS

Atari 400	$39.99
Atari 800 48K	$69.99
Atari 1200 64K	$69.99

COMPUTER PACKAGES

800, 810, 410, Basic, Disks, DC	$239.00
400, 410, Basic, DC	$59.99
1200, 810, 825, 850	$379.00
400, 20-cart, 482 Educator 822	$119.00

(Warranty packages available)

ACCESSORIES

Flip & File + 8 Cartridges	$34.99
Atari 850 Interface Module	$99.99
Atari 835 Direct Modem	$39.99
Atari 410 Program Recorder	$19.99
Atari 822 Thermal Printer	$19.99
Atari 482 The Educator	$24.99
Atari 483 The Programmer	$19.99
Atari Basic	$14.99
Atari Writer	$14.99
Atari AAB4018 Pilot	$9.99
Atari AAB7097 Logo	$9.99

HOME COMPUTERS

APPLE

APPLE IIe	CALL
APPLE IIc	CALL
MacINTOSH	CALL
IIc LCD Display	CALL

Macintosh Software

Lotus Jazz	CALL
Microsoft Excel	CALL
Microsoft Business Pak	$375.00
Living Videotext ThinkTank 512	$159.00
Manhatten Ready, Set, Go	$79.99
Creighton Development Mac Spell	$69.99
Monogram Dollars & Sense	$99.99
Peachtree Back to Basics - GL	$109.00
PFS File & Report (New Version)	$129.00
Silicon Beach Airborn	$25.99

commodore

C128 Computer	**$NEW**
C1571 (Disk Drive for C128)	**$NEW**
C1902 (RGB 13" Monitor for C128)	**$NEW**
C1670 (Modem for C128)	**$NEW**
SX-64 Portable	CALL
Commodore Plus 4	$199.00
CBM 64	$149.00
C1541 Disk Drive	$199.00
C1530 Datasette	$39.99
M-801 Dot Matrix Printer	$189.00
M-802 Dot Matrix/Serial	$219.00
MCS 803 Dot Matrix	$179.00
C1802 Color Monitor	$199.00
C1660 Auto Modem	$59.99
DPS 1101 Daisy Printer	$339.00

PORTABLE COMPUTERS

HEWLETT PACKARD

41CV	$189.99
41CX	$249.99
HP 71B	$419.99
HP 11C	$62.99
HP 12C/15C/16C	$89.99
HP 75D	$999.99
HPIL Module	$98.99
HPIL Cassette or Printer	$359.99
Card Reader	$143.99
Extended Function Module	$63.99
Time Module	$63.99

We stock the full line of HP calculator products

NEC

PC-8401	CALL
PC-8201 Portable Computer	$289.00
PC-8231 Disk Drive	$599.00
PC-8221A Thermal Printers	$149.00
PC-8281A Data Recorder	$99.99
PC-8201-06 8K RAM Chips	$105.00

SHARP

PC-1350	$159.99
PC-1261	$159.99
PC-1260	$109.99
PC-1500A	$165.99
PC-1250A	$88.99
CE-125 Printer/Cassette	$128.99
CE-150 Color Printer Cassette	$171.99
CE-161 16K RAM	$134.99

DISKETTES

maxell

3½" SS/DD	$39.99
3½" DS/DD	$54.99
5¼" MD-1 w/Hardcase	$17.99
5¼" MD-2 w/Hardcase	$23.99
5¼" MD-2-HD for AT	$44.99

Verbatim

5¼" SS/DD	$21.99
5¼" DS/DD	$29.99
Disk Analyzer	$24.99

Dennison

Elephant 5¼" SS/SD	$13.99
Elephant 5¼" SS/DD	$15.99
Elephant 5¼" DS/DD	$16.99
Elephant Premium DS/DD	$23.99

IBM

5¼" DS/DD Floppy Disks (Box of 10) $25.99

DISK HOLDERS

INNOVATIVE CONCEPTS

Flip-in-File 10	$3.99
Flip-in-File 50	$17.99
Flip-in-File 50 w/lock	$24.99
Flip-in-File (400/800 ROM)	$11.99

AMARAY

50 Disk Tub 5¼"	$9.99
30 Disk Tub 3½"	$8.99

SMITH & HILLMAN

Wood Disk Holders CALL

MODEMS

ANCHOR

Volksmodem	$59.99
Volksmodem 300/1200	$189.99
Mark XII (1200 Baud)	$259.00
Signalman Express	$299.00
Lightning 2400 Baud	$399.00

Hayes

Smartmodem 300	$145.00
Smartmodem 1200	$389.00
Smartmodem 1200B	$359.00
Smartmodem 2400	$699.00
Micromodem IIe	$249.00
Smart Com II	$89.99
Chronograph	$199.00
Transet 1000	CALL

AST

Reach 1200 Baud Half Card $399.00

MICROBITS

MPP-1000E AD/AA (Atari)	$79.99
MPP-1064 AD/AA (C-64)	$69.99

Novation

Smart Cat Plus	$319.00
J-Cat	$99.99
Novation 2400	CALL
Apple Cat II	$229.00
212 Apple Cat II	$379.00
Apple Cat 212 Upgrade	$229.00
Macmodem	$319.00

QUADRAM

Quadmodem II
300/1200 $339.00
300/1200/2400 $499.00

TELELEARNING

C64 300 Baud (Closeout) $39.99

GRAPHICS

Koala

IBM	$89.99
Apple/Franklin	$79.99

Polaroid

Palette $1299.00

DRIVES

HARD

ALLOY

PC Stor CALL

IDEAssociates

5 meg Removable/Internal	$1399.00
10 meg Fixed/Internal	$1249.00
15 meg 5 Removable/10 Fixed	$2149.00
25 meg 5 Removable/20 Fixed	$2499.00

i-MEGA

10 Meg Bernoulli Box	$2099.00
20 meg Bernoulli Box	$2599.00
5 meg "MacNoulli"	$1599.00

TALLGRASS TECHNOLOGIES

12, 25, 35, 50, 80 meg (PC) from $1499.00

FLOPPY

INDUS

Apple GT	$209.00
Atari GT	$239.00
C-64 GT	$259.00

μ-SCI

A1.5 Apple	$199.00
A2 Apple	$199.00

USI

SD1 C-64 Single	$219.00
SD2 C-64 Dual	$469.00

Tandon

320K 5¼" (PC) $119.00

TEAC

320K 5¼ $119.00

CALL TOLL-FREE

COMPUTER MAIL ORDER

MONITORS

AMDEK
300 Green	$129.00
300 Amber	$139.00
310 Amber IBM-Plug	$169.00
Color 300 Composite	$239.00
Color 500 Composite/RGB	$389.00
Color 600 Hi-Res (640x240)	$399.00
Color 700 Hi-Res (720x240)	$499.00
Color 710 Long Phosphor	$579.00

NAP
12" Amber/Green Composite	$99.99
12" Amber/Green TTL (ea.)	$119.00

NEC
JB 1260 Green	$59.99
JB 1201/1205 (ea.)	$99.99
JB 1270 Green	$139.00
JB 1275 Amber	$149.00
JB 1280 G TTL/1285 A TTL	$149.00
JC 1410 RGB	$669.00

PRINCETON
MAX-12E Amber	$189.00
HX-9 9" RGB	$469.00
HX-9E Enhanced	$519.00
HX-12 12" RGB	$469.00
HX-12E Enhanced	$559.00
SR-12 Hi-Res	$599.00
SR-12 P Enhanced	$649.00

TAXAN
115 12" Green Mono	$119.00
116 12" Amber Mono	$119.00
121 Green TTL	$139.00
122 Amber TTL	$149.00
420 Hi-Res RGB (IBM)	$429.00
440 Ultra Hi-Res RGB	$559.00

QUADRAM
8400 Quadchrome I	$479.00
8410 Quachrome II	$429.00
8420 Amberchrome	$179.00
8500 Quad Screen	$1499.00

ZENITH
ZVM 122/123	$79.99
ZVM 124 IBM Amber	$149.00
ZVM 130 Color	$269.00
ZVM 131 Color	$249.00
ZVM 133 RGB	$429.00
ZVM 135 RGB/Color	$459.00
ZVM 136 RGB/Color	$599.00
1220, 1230, 1240	CALL

INTERFACES

PRACTICAL PERIPHERALS
Graphcard	$79.99
Seriall Card	$99.99
Microbuffer II+	$169.00
Microbuffer 32K	$189.00

QUADRAM
Microfazer	from $139.00
Efazer (Epson)	from $79.99

Orange Micro
Grappler CD (C64)	$99.99
Grappler + (Apple)	$89.99
Grappler 16K+ (Apple)	$159.00

DIGITAL DEVICES
Ape Face (Atari)	$49.99
U-Print A (Atari)	$54.99
U-A16/Buffer (Atari)	$74.99
U-Call Interface (Atari)	$39.99
U-Print C (C64)	$49.99
P-16 Print Buffer	$74.99

MPP MICROBITS
MPP-1150 Parallel (Atari)	$69.99
MP-1150XL (Atari 1200XL)	$69.99
MicroStuffer 64K Print Buffer	$109.00

PRINTERS

AXIOM
AT-100 Atari Interface Printer	$139.00
AT-550 Atari Dual Mode	$249.00
GP-100 Parallel Interface	$189.00
GP-700 Color Printer	$449.00
GP-550 Parallel Printer	$239.00
Elite 5 Letter Quality	$239.00
Elite 5 C64 Interface	$249.00
Penman 3-pen	$289.00

CITIZEN
MSP-10 (80 col.)	$349.00
MSP-15 (132 col.)	$499.00
MSP-20 (80 col.)	$489.00
MSP-25 (132 col.)	$679.00

C.ITOH
Prowriter 7500	$219.00
Prowriter 8510P	$299.00
Prowriter 8510 NLQ	$329.00
Prowriter 1550P	$469.00
F10-40P Starwriter	$869.00
F10-55 Printmaster	$1049.00
Prowriter 8510-NLQ	$329.00

corona
Lazer LP-300	$2799.00

DIABLO
D25 Daisywheel	$599.00
630-109 Daisywheel	$1749.00
D801F Daisywheel	CALL

daisywriter
2000	$749.00

EPSON
LX-80, LX-90, FX-85, FX-185
JX-80, DX-10, DX-20, SQ 2000
HS-80, Homewriter 10 ..... CALL

JUKI
6000 Letter Quality	CALL
6100 Letter Quality	CALL
6300 Letter Quality	CALL

NEC
8027 Transportable	$299.00
2000 Series	$699.00
3000 Series	$1099.00
8000 Series	$1449.00
ELF 360	$449.00

OKIDATA
84, 182, 192, 193, 2410	CALL
Okimate 10 (Specify C64/Atari)	$189.00
Okimate 20 (IBM)	CALL

OLYMPIA
Needlepoint Dot Matrix	$299.00
Compact RO	$339.00
Compact 2	$369.00

Panasonic
KX1091	$259.00
KX1092	$389.00
KX1093	$479.00

QUADRAM
Quadjet	$399.00

SILVER-REED
500 Letter Quality	$279.00
550 Letter Quality	$419.00
770 Letter Quality	$759.00

STAR
SG-10C (C64 Interface)	$NEW
SB/SD/SG/SR Series	CALL
Powertype Letter Quality	CALL

TOSHIBA
1340 (80 column)	$599.00
P351 (132 column)	$1299.00

PC COMPATIBLES

IBM PC SYSTEMS
Configured to your specification. Call for Best Price!
IBM-PC, IBM-PC II, IBM-XT, IBM-AT

SOFTWARE FOR IBM

ALPHA
Electric Desk IBM-PC	$199.00

ASHTON-TATE
Framework	$369.00
dBase III	$369.00

BORLAND INTERNATIONAL
Turbo Pascal 3.0	$49.99
Sidekick	$39.99

CENTRAL POINT
Copy II PC-Backup	$29.99

DECISION RESOURCES
Chartmaster	$259.00
Signmaster	$179.00

FOX & GELLER
Quickcode III	$169.00

FUNK SOFTWARE
Sideways	$39.99

Harvard Software Inc.
Harvard Project Manager	$209.00
Total Project Manager	$269.00

Human Edge™
Communication Edge	$99.99
Management Edge	$119.00
Negotiation Edge	$139.00
Sales Edge	$119.00

LIFETREE
Volkswriter Duluxe	$159.00

LIVING VIDEOTEXT
Think Tank	$109.00

Lotus
Symphony	$439.00
1-2-3	$309.00

MECA SOFTWARE
Managing Your Money	$109.00

MICROSTUF
Crosstalk XVI	$89.99
Microstuf Remote	$89.99

MICRORIM
R:Base 4000	$259.00
R:Base 5000	$399.00
Clout 2.0	$129.00

MicroPro
WordStar 2000	$249.00
WordStar 2000+	$309.00
WordStar Professional	$299.00

MICROSOFT
Word	$239.00
Mouse	$139.00
Flight Simulator	$39.99
MultiPlan	$129.00

MultiMate
Multi Mate Word Proc	$249.00

NOUMENON
Intuit	$69.99

NORTON
Norton Utilities 3.0	$59.99

Peachtree Software
Peachtext 5000	$169.00
PeachPack (GL/AP/AR)	$219.00

pfs:
IBM/APPLE
Access	$54.99
Write/Graph/File/Plan (ea.)	$79.99
Report	$74.99
Proof	$59.99

Professional Software
Wordplus-PC w/Boss	$249.00

ROSESOFT
Prokey	$89.99

THE SOFTWARE GROUP
Enable	$339.00

SATELLITE SYSTEMS
Word Perfect 4.0	$219.00

SORCIM/IUS
Accounting AP/AR/GL/INV/OE (ea.)	$295.00
SuperCalc III	$195.00
EasyWriter II System	$195.00
Super Project	$195.00

SPI
Open Access	$379.00

THOUGHTWARE
Trigger	$289.00
Sell, Sell, Sell Training	$299.00
Sell, Sell, Sell Application	$179.00

AT&T
Safari (7300)	CALL
6300	CALL

corona
PPC400 Dual Portable	$1799.00
PPCXT 10 meg Portable	$2399.00
PC40022 10 meg Desktop	$1999.00

ITT
ITT X-TRA 256K, 2 Drive System	CALL
256K, 10 meg Hard Drive System	CALL

SANYO
MBC 550-2 Single Drive	$699.00
MBC 555-2 Dual Drive	$969.00
MBC 775 Portable	$1599.00
MBC 511 10 meg	CALL
MBC 675 Portable	CALL
MBC 880 Desktop	CALL

ZENITH
PC-151-21 Single Desktop	CALL
PC-151-52 Dual Desktop	CALL
PC-151-53 10 meg Desktop	CALL
PC-161-21 Single Portable	CALL
PC-161-52 Dual Portable	CALL
Z-200 (AT)	CALL
171 (Portable)	CALL
138 (Transportable) (NEW)	CALL
148 (DeskTop) (NEW)	CALL

MULTIFUNCTION CARDS

AST
Six Pack Plus	$239.00
Mega Plus II	$269.00
I/O Plus II	$139.00
Advantage-AT	$399.00
Graph Pak	$599.00
Monograph Plus	$399.00
Preview Mono	$299.00
PC Net Cards	$379.00
5251/11 On-line	$799.00
5251/12 Remote	$579.00
3780 Emulation Card	$639.00
BSC Bisync	$489.00

dca
IRMA 3270	$879.00
IRMA Print	$999.00

EVEREX
Color Card (Graphics Edge)	$299.00
Magic Card	$199.00

HERCULES
Graphics	$299.00
Color	$159.00

IDEAssociates
IDEAmax - ZPR, 64K, C, S, P.	$229.00
IDEAmini - YPR, C, S, P.	$189.00
IDEAminimax - MPR 128K	$229.00
IDEAshare Software	$219.00
IDEA 5251	$699.00

MYLEX
The Chairman	$489.00

PARADISE
Modular Graphics Card	$274.00
Multi Display Card	$289.00
Five Pack C, S	$159.00

PERSYST
Bob Board	$389.00

PLANTRONICS
Color Plus	$369.00

LEGAMAR
Captain - 64	$239.00
Captain Jr. 128K	$339.00
Graphics Master	$469.00

QUADRAM
Quadport-AT	$119.00
Quadmeg - AT (128K)	$349.00
The Gold Quadboard	$449.00
The Silver Quadboard	$239.00
Expanded Quadboard	$219.00
Quad 512+	$229.00
Liberty	$309.00
QuadSprint	$499.00
QuadLink	$399.00
Quadcolor 1	$199.00
QuadJr. Expansion Chassis	$469.00
Expansion Chassis Memory	$199.00
Quadmem, Jr.	$199.00
Chronagraph	$79.99
Parallel Interface Board	$64.99

```
      1         2
    12345678901234567890
 1  YARRAHPQRZERRNVPSSJQ
 2  GYSJLEJORLIIWBRBOXVC
 3  NCOUAXUXNYRANIBQSNKR
 4  ITRHMADBZMEMORYAPPLE
 5  ROTLODSASETYBAGIGIPC
 6  TAIPCEVARIABLESQYVCA
 7  SZNHLCMHPRINTERSOTCZ
 8  XMGDAINDISKDRIVECIMP
 9  GSOSCMLACLOGOFYHSHQY
10  PGTWSAEONBRBQCFAWCII
11  EBSZALMVCOAOQPBGZLBX
12  ZRBSPMAATOSNPMLLKKWW
13  QCUCODOITIMEFBLOPMRP
14  XFYDGFNGNAHPCIPBASTF
15  KBOLXOVTOURVUONOUZJC
16  VRTAMZUYCEEIBTNCXFMX
17  EJENITUORBUSBWEDXZPZ
18  PKHAVBAVFLOKXGBRETDW
19  VECAFRETNIAYKJKDAPMF
20  MYEAIOZFJTSIZSDKQXZY
```

"Word Search" prints out challenging hidden-word puzzles of various sizes on your printer.

printers can't print more than 80 columns unless you first send the printer a special escape code for condensed type (see your printer manual).

Commodore Versions

The line changes listed as Program 2 are for the Commodore 64, 128, Plus/4, 16, PET, and VIC-20 (with at least 8K expansion). If you're using a VIC with only 8K expansion, type in the line changes shown in Program 2 and also substitute lines 95 and 100 in Program 4. If you're using a Commodore 16, type the line changes from Program 2 and also substitute lines 95 and 100 in Program 3. The VIC with only 8K expansion can hide a maximum of 50 words in a 50 × 50 grid; the 16 is limited to a maximum of 60 words in a 60 × 60 grid. If you're using a PET, you'll have to make similar adjustments, depending on the amount of memory available.

Apple And IBM

The Apple version of Word Search runs on any Apple II-series computer with either DOS 3.3 or ProDOS. Follow the general instructions above, typing in the line changes listed as Program 5. IBM users should enter the line changes in Program 6; this version runs on a PC or PCjr with any memory configuration.

TI Word Search

Program 7 lists the line changes required for TI. The unexpanded TI-99/4A is limited to 50 words in a 50 × 50 grid. However, with memory expansion this number can be increased by changing the value of MC in line 95 from 50 to the desired value. You will also need to increase every occurrence of 50 in line 100 to the same value. Adjust line 2000 for whatever configuration your particular printer requires.

Atari Version

The Atari version of Word Search is complete in itself. Simply type in Program 8, save a copy, and run it. Ataris with 32K or 48K memory can create puzzles with up to 100 words in a 99 × 99 grid. If your Atari has 16K, you're limited to 25 words in a 25 × 25 grid. To run Word Search on a 16K Atari you must make two additional changes in line 100 of Program 8: Change the 99 and the 100 to 25.

Program 1: Word Search (Main Program)

Version By Patrick Parrish, Programming Supervisor

Please refer to the article instructions before entering this listing.

```
95 MC=99
100 DIM FF$(100),S$(99),W$(100
    ),CC(100),RR(100),L(100),E
    $(2,2)
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 READ E$(I+1,J+1)
140 NEXT J
150 NEXT I
160 DATA "NW"," N","NE"," W","
    {2 SPACES}"," E","SW"," S"
    ,"SE"
170 FOR I=1 TO MC
180 G$=G$+" "
190 NEXT I
200 FOR I=1 TO 8
210 READ D(1,I),D(2,I)
220 NEXT I
230 DATA -1,-1,-1,0,-1,1,0,-1
240 DATA 0,1,1,-1,1,0,1,1
250 GOTO 1220
260 REM SHELL SORT
270 PRINT "SORTING..."
280 X=1
290 X=2*X
300 IF X<=W0 THEN 290
310 X=INT(X/2)
320 IF X<>0 THEN 340
330 RETURN
340 FOR Y=1 TO W0-X
350 Z=Y
360 A=Z+X
370 IF L(Z)>=L(A) THEN 460
380 X$=W$(Z)
390 W$(Z)=W$(A)
400 W$(A)=X$
410 B=L(Z)
420 L(Z)=L(A)
430 L(A)=B
440 Z=Z-X
450 IF Z>0 THEN 360
460 NEXT Y
470 GOTO 310
480 REM HIDE WORDS
490 FOR X=1 TO W0
500 FOR Y=1 TO 50
510 R1=INT(RND(1)*R0)
520 C1=INT(RND(1)*C0)
530 D1=INT(RND(1)*8)+1
540 O1=D1
550 DX=D(1,D1)
560 DY=D(2,D1)
570 IF R1+DX*L(X)<1 OR R1+DX*L
    (X)>R0 OR C1+DY*L(X)<1 THE
    N 590
580 IF C1+DY*L(X)<=C0 THEN 630
590 D1=D1*(D1<8)*(1=1)+1
600 IF D1<>O1 THEN 550
610 NEXT Y
620 GOTO 800
630 FOR Z=1 TO L(X)
640 IF MID$(W$(X),Z,1)<"A" OR
    {SPACE}MID$(W$(X),Z,1)>"Z"
     THEN 680
650 R1=R1+DX
660 C1=C1+DY
670 IF MID$(S$(R1),C1,1)<>" "
    {SPACE}AND MID$(S$(R1),C1,
    1)<>MID$(W$(X),Z,1) THEN 5
    90
680 NEXT Z
690 FOR Z=L(X) TO 1 STEP -1
700 IF MID$(W$(X),Z,1)<"A" OR
    {SPACE}MID$(W$(X),Z,1)>"Z"
     THEN 770
710 S$(R1)=MID$(S$(R1),1,C1-1)
    +MID$(W$(X),Z,1)+MID$(S$(R
    1),C1+1)
720 RR(X)=R1
730 CC(X)=C1
740 FF$(X)=E$(DX+1,DY+1)
750 R1=R1-DX
760 C1=C1-DY
770 NEXT Z
780 NEXT X
790 GOTO 890
800 GOSUB 1720
810 PRINT "SORRY, BUT I CAN'T
    {SPACE}FIT WORD NUMBER ";S
    TR$(X);" , ";W$(X);" , ";
820 PRINT "INTO THE GRID. SHOU
    LD I SKIP IT, START OVER,
    {SPACE}OR TRY AGAIN"
830 INPUT X$
840 IF MID$(X$,1,2)="ST" THEN
    {SPACE}1660
850 IF MID$(X$,1,2)="TR" THEN
    {SPACE}500
860 IF MID$(X$,1,2)<>"SK" THEN
     830
870 W$(X)="/"
880 GOTO 780
890 FOR X=1 TO R0
900 FOR Y=1 TO C0
910 IF MID$(S$(X),Y,1)<>" " TH
    EN 930
920 S$(X)=MID$(S$(X),1,Y-1)+CH
    R$(INT(26*RND(1)+65))+MID$
    (S$(X),Y+1)
930 NEXT Y
940 NEXT X
950 REM DONE
960 PRINT
970 PRINT "I AM FINISHED. WHAT
     DO YOU WANT TO CALL THE W
    ORD SEARCH"
980 INPUT T$
990 SL=0
1000 PRINT
1010 PRINT "DO YOU WANT TO PRI
     NT THE SOLUTION (Y/N)"
1020 GOSUB 1180
1030 IF A$="N" THEN 1050
1040 SL=1
```

The complete 1541 enhancement system . . .

STARDOS

MAKES GREAT COFFEE!!

The 1541 is the slowest disk drive on planet earth. Even simple operations seem to take forever. Quickloaders and Fastloaders that software-patch the operating system are vulnerable to being knocked out of memory, rendering them totally useless. Even Flashier products that require permanent modifications to the 64 and 1541 can't compete with the blinding speed of STARDOS.

STARDOS accelerates every (yes, we said every) function of the 1541 disk drive. Other fast loaders only load PRG files faster. STARDOS also speeds up SEQ, REL, USR and DIRECT ACCESS files. Everything including FORMAT, VERIFY, SCRATCH, VALIDATE, INITIALIZE and COPY are much faster. In addition STARDOS adds a vast array of easy to use commands and utilities all at the touch of a key.

A sampling of STARDOS features:

- Accesses ALL types of files up to 1,000% faster!
- Saves up to 300% faster than normal (with extended verify)
- DOES NOT CHANGE THE SPEED AT WHICH THE DRIVE MOTOR SPINS
- Makes your 1541 MORE RELIABLE and LESS PRONE TO BREAK DOWN or OVERHEAT!
- STARDOS is fully expandable for multiple fast disk drives
- Easy (5 minute) plug in installation. User friendly manual
- 100% compatible with software and serial bus peripherals
- Adds years of life to your disks and drive in reduced wear
- Cures a number of bugs in the Commodore 64 and 1541 disk drive including :
  - The damaging 'Head Knock' that can mis-align your 1541
  - The @: Save with replace bug!!
  - The Editor lock-up bug
- Harness the full power of your disk drive with the built in DOS wedge
- Lock/unlock files and protect/unprotect disks from the keyboard
- Powerful sector editor allows direct viewing of diskette
- Upgrades computer and drive to the latest Commodore specifications
- Instant access to the built in mini-wordprocessor for short notes and memos
- Built in copier copies all file types (even relative) easily and effectively
- Built in disk duplicator copies an entire diskettte in less than 3 minutes
- Fully expanded machine language monitor, always on line
- Quality hardware. Full 120 day repair or replace warranty
- Satisfaction guaranteed or money back—no questions asked
- (Oh, by the way, we lied, STARDOS makes ***LOUSY COFFEE***)

Simply the best. **$74.95**

[Take advantage of our introduction special at **$64.95**]

Personalize your C64! for a $10 (non-refundable) charge we will include a 21 character power-on message with your favorite screen, border and text colors. *Example:*

	M	I	K	E	'	S		C	O	M	M	O	D	O	R	E	-	6	4	

Background Black Border Red Text Green

Write of phone for additional information.

*Trademark of Commodore Business Machines

. . . WRITE OR PHONE . . .

STARPOINT SOFTWARE

Star Route 10 Gazelle, CA 96034 [916] 435-2371

When ordering by mail:

- $64.95 + 3.00 shipping
- $64.95 + 4.00 COD orders
- Shipping out of USA $6.00
- Calif. residents add 6% sales tax
- VISA or Mastercard accepted

Please allow 4-6 weeks for delivery.

```
1050 GOSUB 2000
1060 GOSUB 1720
1070 F=0
1080 PRINT "DO YOU WANT ANOTHE
     R GRID (Y/N)"
1090 GOSUB 1180
1100 IF A$="Y" THEN 1120
1110 END
1120 PRINT
1130 PRINT "DO YOU WANT TO USE
      THE SAME WORDS (Y/N)"
1140 GOSUB 1180
1150 IF A$="N" THEN 1280
1160 F=1
1170 GOTO 1340
1180 INPUT A$
1190 IF A$<>"Y" AND A$<>"N" TH
     EN 1180
1200 RETURN
1210 REM INITIALIZATION
1220 GOSUB 1720
1230 LL=6
1240 GOSUB 1740
1250 PRINT "{8 SPACES}WORD SEA
     RCH"
1260 LL=4
1270 GOSUB 1740
1280 FOR I=1 TO W0
1290 W$(I)=""
1300 L(I)=0
1310 NEXT I
1320 PRINT "HOW MANY WORDS WOU
     LD YOU LIKE IN YOUR WORD
     {SPACE}SEARCH"
1330 INPUT W0
1340 PRINT
1350 PRINT "HOW MANY ROWS AND
     {SPACE}COLUMNS IN THE GRI
     D"
1360 INPUT R0,C0
1370 PRINT
1380 PRINT
1390 IF R0*C0>=10*W0 THEN 1440
1400 PRINT "I DON'T THINK I CO
     ULD DO THIS."
1410 FOR I=1 TO 1000
1420 NEXT I
1430 GOTO 1340
1440 PRINT "I THINK I CAN DO T
     HIS."
1450 IF C0<=MC THEN 1470
1460 PRINT "(BUT IT WON'T FIT
     {SPACE}ON THE PAPER.)"
1470 IF F=1 THEN 1660
1480 LL=3
1490 GOSUB 1740
1500 PRINT "ENTER THE ";STR$(W
     0);" WORDS. TO CORRECT A
     {SPACE}MISTAKE, ENTER X"
1510 PRINT
1520 FOR I=1 TO W0
1530 PRINT "WORD NUMBER ";I;":
     "
1540 INPUT X$
1550 IF LEN(X$)<=R0 AND LEN(X$
     )<=C0 AND X$<>"X" THEN 16
     10
1560 IF X$<>"X" THEN 1590
1570 I=I-(I>1)*(1=1)
1580 GOTO 1530
1590 PRINT "OOPS...THE WORD IS
      TOO LONG."
1600 GOTO 1530
1610 W$(I)=X$
1620 L(I)=LEN(X$)
1630 NEXT I
1640 GOSUB 1720
1650 GOSUB 270
1660 PRINT
1670 PRINT "OKAY, I WILL GO TO
      WORK (WISH ME LUCK...)."
1680 FOR I=1 TO R0
1690 S$(I)=LEFT$(G$,C0)
1700 NEXT I
1710 GOTO 490
1730 RETURN
1740 FOR I=1 TO LL
1750 PRINT
1760 NEXT I
1770 RETURN
1999 REM PRINTER ROUTINE
```

Program 2: Line Changes For Commodore 64, 128, Plus/4, 16, PET, and VIC-20

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
1720 PRINT CHR$(147)    :rem 69
2000 OPEN3,4:PRINT#3,T$:PRINT#
     3                  :rem 101
2010 PRINT#3,"{4 SPACES}";:FOR
     I=1TOC0:IFI/10<>INT(I/10)
     THENPRINT#3," ";:GOTO2030
                        :rem 101
2020 PRINT#3,MID$(STR$(I),2,1)
     ;                  :rem 207
2030 NEXTI:PRINT#3      :rem 106
2040 PRINT#3,"{4 SPACES}";:FOR
     I=1TOC0:PRINT#3,RIGHT$(ST
     R$(I),1);:NEXTI:PRINT#3
                        :rem 172
2050 FORX=1TOR0:IFX<10THENPRIN
     T#3," ";           :rem 20
2060 PRINT#3,STR$(X)" ";
                        :rem 28
2070 FORY=1TOC0:PRINT#3,MID$(S
     $(X),Y,1);         :rem 98
2080 NEXTY:PRINT#3:NEXTX:PRINT
     #3:PRINT#3:PRINT#3,"WORD
     {SPACE}LIST:"      :rem 201
2090 FORX=1TOW0:IFW$(X)="/"THE
     N2110              :rem 50
2100 PRINT#3,W$(X)      :rem 246
2110 NEXTX:FORI=1TO5:PRINT#3:N
     EXTI:IFSL=0THEN2180
                        :rem 185
2120 PRINT#3,"SOLUTION LIST:":
     PRINT#3,"WORD{21 SPACES}R
     OW{3 SPACES}COLUMN";
                        :rem 213
2130 PRINT#3,"{3 SPACES}DIR"
                        :rem 248
2140 FORX=1TOW0:IFW$(X)="/"THE
     N2170              :rem 52
2150 PRINT#3,W$(X);LEFT$(G$,25
     -LEN(W$(X)));RR(X);LEFT$(
     G$,8-LEN(STR$(RR(X))));
                        :rem 218
2160 PRINT#3,CC(X);LEFT$(G$,6-
     LEN(STR$(CC(X))));FF$(X)
                        :rem 61
2170 NEXTX               :rem 97
2180 CLOSE3:RETURN      :rem 142
```

Program 3: Additional Line Changes For Commodore 16

```
95 MC=60
100 DIM FF$(60),S$(60),W$(60),
    CC(60),RR(60),L(60),E$(2,2
    )
```

Program 4: Additional Line Changes For 8K VIC-20

```
95 MC=50                :rem 160
100 DIM FF$(50),S$(50),W$(50),
    CC(50),RR(50),L(50),E$(2,2
    )                   :rem 25
```

Program 5: Line Changes For Apple

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

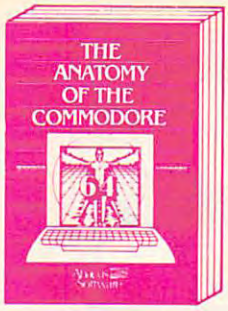
```
38 90 D$ = CHR$ (4):I$ = CHR$ (9
   )
4E 1720 HOME
E1 2000 PRINT D$;"PR#1": PRINT I
   $;"80N"
9E 2010 PRINT T$: PRINT
3F 2020 PRINT "   ";: FOR I = 1
   TO C0: IF I / 10 < > INT
    (I / 10) THEN PRINT " "
   ;: GOTO 2040
1A 2030 PRINT MID$ ( STR$ (I),1,
   1);
77 2040 NEXT I: PRINT
98 2050 PRINT "   ";: FOR I = 1
   TO C0: PRINT RIGHT$ ( ST
   R$ (I),1);: NEXT I: PRIN
   T
C0 2060 FOR X = 1 TO R0: IF X <
   10 THEN PRINT " ";
61 2070 PRINT STR$ (X)" ";
5A 2080 FOR Y = 1 TO C0: PRINT M
   ID$ (S$(X),Y,1);
21 2090 NEXT Y: PRINT : NEXT X:
   PRINT : PRINT : PRINT "W
   ORD LIST:"
30 2100 FOR X = 1 TO W0: IF W$(X
   ) = "/" THEN 2120
C2 2110 PRINT W$(X)
27 2120 NEXT X: FOR I = 1 TO 5:
   PRINT : NEXT I: IF SL =
   0 THEN 2160
B5 2130 PRINT "SOLUTION LIST:":
   PRINT "WORD
            ROW   COLUMN   D
   IR": FOR X = 1 TO W0: IF
    W$(X) = "/" THEN 2150
4D 2140 PRINT W$(X) LEFT$ (G$,26
    - LEN (W$(X)))RR(X) LEF
   T$ (G$,9 - LEN ( STR$ (R
   R(X))))CC(X) LEFT$ (G$,6
    - LEN ( STR$ (CC(X))))F
   F$(X)
91 2150 NEXT X
BF 2160 PRINT : PRINT D$;"PR#0":
   RETURN
```

Program 6: IBM PC/PCjr Line Changes

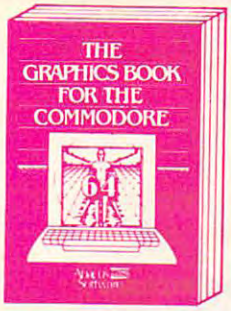
For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
IC 10 DEF SEG=0:POKE 1047,(PEEK(
   1047) OR 64)
JD 20 WIDTH 40:KEY OFF:DEF SEG=&
   H40:RANDOMIZE PEEK(&H6D)
ND 1720 CLS
NF 2000 ON ERROR GOTO 2170
EK 2010 OPEN "LPT1:" FOR OUTPUT
    AS #1:PRINT #1,T$:PRINT
   #1,
MH 2020 PRINT #1,"    ";:FOR I=1
    TO C0:IF I/10<>INT(I/10
   ) THEN PRINT #1," ";:GOT
   O 2040
NH 2030 PRINT #1,MID$(STR$(I),2,
   1);
KE 2040 NEXT I:PRINT #1,
AF 2050 PRINT #1,"    ";:FOR I=1
    TO C0:PRINT #1,RIGHT$(S
   TR$(I),1);:NEXT I:PRINT
   #1,
EH 2060 FOR X=1 TO R0:IF X<10 TH
   EN PRINT #1," ";
PH 2070 PRINT #1,STR$(X)" ";
```

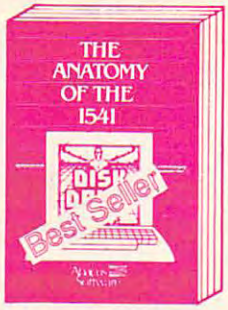
SUPER BOOKS



300 pages, $19.95



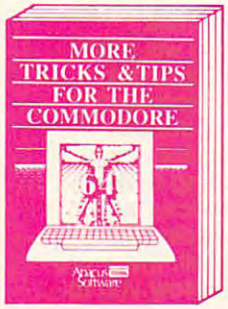
350 pages, $19.95



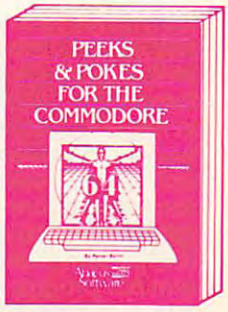
320 pages, $19.95

200 pages, $19.95

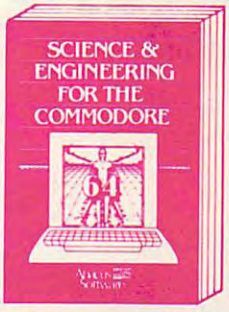
275 pages, $19.95



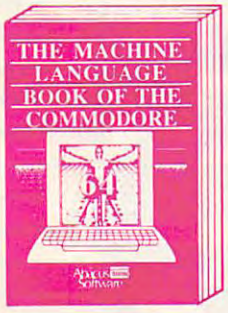
250 pages, $19.95



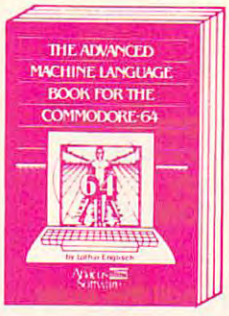
200 pages, $14.95



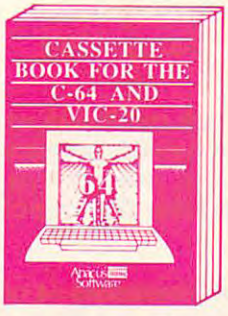
340 pages, $19.95



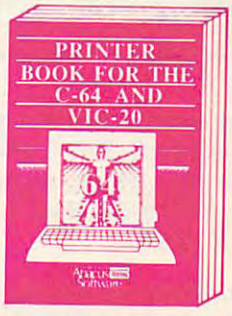
215 pages, $14.95



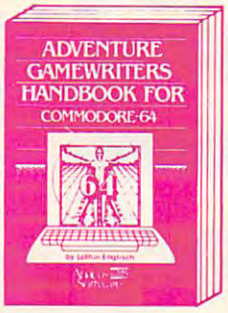
210 pages, $14.95



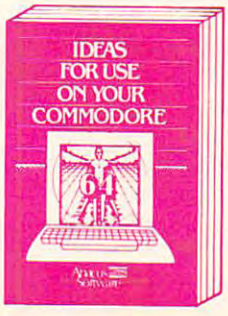
210 pages, $14.95



330 pages, $19.95

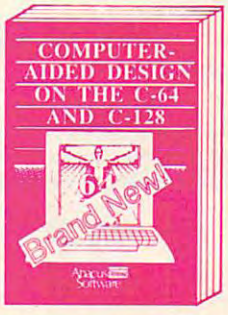


225 pages, $14.95



220 pages, $12.95

250 pages, $19.95



250 pages, $19.95

For fast service call **616/241-5510.** For postage and handling, include $4.00 per order. Foreign orders include $8.00 per item. Money orders and checks in U.S. dollars only. Mastercard, Visa and Amex accepted.

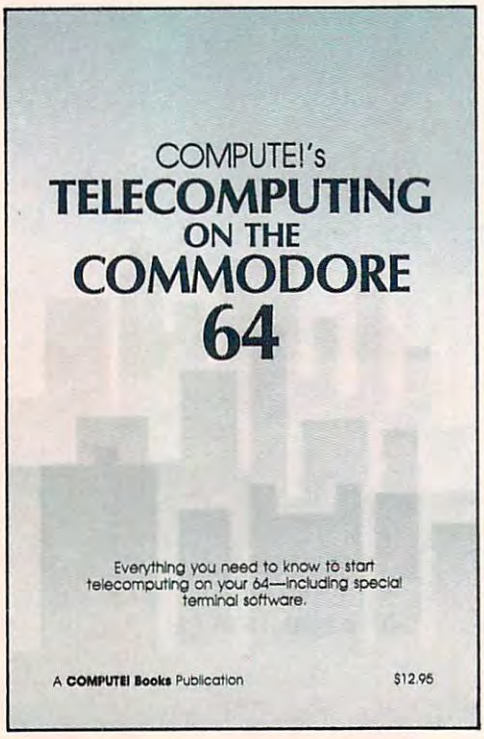
Dealer Inquiries Welcome
More than 1200 dealers nationwide

Abacus Software

P.O. Box 7211 Grand Rapids, MI 49510 616/241-5510

Other software also available!
Call now for free catalog and the name of your nearest dealer. Phone:616/241-5510.

Learn All About TELECOMPUTING



with

COMPUTE!'s Telecomputing on the Commodore 64

This comprehensive guide explains everything you need to know to start telecomputing on your Commodore 64.

Many home computer analysts believe that telecomputing will soon be the most popular use for home computers. *COMPUTE!'s Telecomputing on the Commodore 64* introduces readers to telecommunications, with sections on buying and using modems, accessing information services and bulletin boards, and uploading and downloading files. Best of all, there are terminal programs which allow you to communicate with the information services and bulletin boards.

Edited
$12.95
0-87455-009-2

For your convenience, there's also a disk available which includes all the programs in the book. This 5¼-inch floppy disk saves you hours of typing and prevents many typing errors. It's fast, inexpensive, and ready to load on your Commodore 64. You can order the *Disk* directly from COMPUTE! Books for only $12.95.

To order *COMPUTE!'s Telecomputing on the Commodore 64* or the *Disk*, mail the order form with your payment to COMPUTE! Books, P.O. Box 5058, Greensboro, NC 27403. For fastest service, call toll free 800-334-0868 (in NC 919-275-9809.)

YES! I WANT TO LEARN TO TELECOMPUTE WITH MY COMMODORE 64.

Please send me:

____ ***COMPUTE!'s Telecomputing on the Commodore 64,*** (009-2) $12.95 ea. _____
(Add $2.00 per book U.S. and surface mail, $5.00 airmail, for shipping charges.)

____ ***COMPUTE!'s Telecomputing on the Commodore 64 Disk,*** $12.95 each _____
(Add $2.00 per disk for shipping charges.)

ALL ORDERS MUST BE PREPAID IN U.S. FUNDS

Subtotal _____
NC residents add 4.5% tax _____
Shipping charges _____
Total paid _____

☐ Payment enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ Visa ☐ American Express

Account No. _____ Exp. Date _____
(Required)

Signature _____
Name _____
Address _____
City _____ State _____ Zip _____

Please allow 4–6 weeks for delivery

7590091

COMPUTE! Publications, Inc.
One of the ABC Publishing Companies

COMPUTE! Books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England.

```
IF 2080 FOR Y=1 TO C0:PRINT #1,MID$(S$(X),Y,1);
CC 2090 NEXT Y:PRINT #1,:NEXT X:PRINT #1,:PRINT #1
      ,:PRINT #1,"WORD LIST:"
DM 2100 FOR X=1 TO W0:IF W$(X)="/" THEN 2120
CI 2110 PRINT #1,W$(X)
HJ 2120 NEXT X:FOR I=1 TO 5:PRINT #1,:NEXT I:IF SL
      =0 THEN 2160
GJ 2130 PRINT #1,"SOLUTION LIST:":PRINT #1,"WORD
                        ROW    COLUMN    DIR":FOR
       X=1 TO W0:IF W$(X)="/" THEN 2150
LA 2140 PRINT #1,W$(X);LEFT$(G$,25-LEN(W$(X)));RR(
      X);LEFT$(G$,8-LEN(STR$(RR(X))));CC(X);LEFT
      $(G$,6-LEN(STR$(CC(X))));FF$(X)
AG 2150 NEXT X
DA 2160 CLOSE #1:ON ERROR GOTO 0:RETURN
IM 2170 CLOSE #1:PRINT "PRINTER ERROR #";ERR;"OCCU
      RRED.":PRINT "TRY AGAIN."
JL 2180 PRINT:PRINT "HIT A KEY TO CONTINUE"
CA 2190 A$=INKEY$:IF A$="" THEN 2190
HM 2200 RESUME 2010
```

Program 7: TI-99/4A Line Changes

```
80 RANDOMIZE
95 MC=50
100 DIM FF$(50),S$(50),W$(50),CC(50),RR(50),L(5
    0),E$(2,2)
180 G$=G$&" "
510 R1=INT(RND*R0)
520 C1=INT(RND*C0)
530 D1=INT(RND*8)+1
570 IF (R1+DX*L(X)<1)+(R1+DX*L(X)>R0)+(C1+DY*L(
    X)<1)THEN 590
640 IF (SEG$(W$(X),Z,1)<"A")+(SEG$(W$(X),Z,1)>"
    Z")THEN 680
670 IF (SEG$(S$(R1),C1,1)<>" ")*(SEG$(S$(R1),C1
    ,1)<>SEG$(W$(X),Z,1))THEN 590
700 IF (SEG$(W$(X),Z,1)<"A")+(SEG$(W$(X),Z,1)>"
    Z")THEN 770
710 S$(R1)=SEG$(S$(R1),1,C1-1)&SEG$(W$(X),Z,1)&
    SEG$(S$(R1),C1+1,LEN(S$(R1))-C1)
840 IF SEG$(X$,1,2)="ST" THEN 1670
850 IF SEG$(X$,1,2)="TR" THEN 500
860 IF SEG$(X$,1,2)<>"SK" THEN 830
910 IF SEG$(S$(X),Y,1)<>" " THEN 930
920 S$(X)=SEG$(S$(X),1,Y-1)&CHR$(INT(26*RND+65)
    )&SEG$(S$(X),Y+1,LEN(S$(X))-Y)
1190 IF (A$<>"Y")*(A$<>"N")THEN 1180
1550 IF (LEN(X$)<=R0)*(LEN(X$)<=C0)*(X$<>"X")TH
     EN 1610
1690 S$(I)=SEG$(G$,1,C0)
1720 CALL CLEAR
2000 OPEN #1:"RS232"
2010 PRINT #1:T$
2020 PRINT #1
2030 PRINT #1:"(3 SPACES)";
2040 FOR I=1 TO C0
2050 IF I/10=INT(I/10)THEN 2080
2060 PRINT #1:" ";
2070 GOTO 2090
2080 PRINT #1:SEG$(STR$(I),1,1);
2090 NEXT I
2100 PRINT #1
2110 PRINT #1:"(3 SPACES)";
2120 FOR I=1 TO C0
2130 PRINT #1:SEG$(STR$(I),LEN(STR$(I)),1);
2140 NEXT I
2150 PRINT #1
2160 FOR X=1 TO R0
2170 IF X>=10 THEN 2190
2180 PRINT #1:" ";
2190 PRINT #1:STR$(X);" ";
2200 FOR Y=1 TO C0
2210 PRINT #1:SEG$(S$(X),Y,1);
2220 NEXT Y
2230 PRINT #1
2240 NEXT X
2250 PRINT #1
2260 PRINT #1
2270 PRINT #1:"WORD LIST:"
2280 FOR X=1 TO W0
2290 IF W$(X)="/" THEN 2310
2300 PRINT #1:W$(X)
2310 NEXT X
2320 FOR I=1 TO 5
2330 PRINT #1
2340 NEXT I
2350 IF SL=0 THEN 2450
2360 PRINT #1:"SOLUTION LIST:"
2370 PRINT #1:"WORD(21 SPACES)ROW(3 SPACES)COLUM
     N";
2380 PRINT #1:"(3 SPACES)DIR"
2390 FOR X=1 TO W0
2400 IF W$(X)="/" THEN 2440
2410 PRINT #1:W$(X);SEG$(G$,1,25-LEN(W$(X)));RR
     (X);
2420 PRINT #1:SEG$(G$,1,7-LEN(STR$(RR(X))));CC(
     X);SEG$(G$,1,4-LEN(STR$(CC(X))));
2430 PRINT #1:FF$(X)
2440 NEXT X
2450 CLOSE #1
2460 RETURN
```

A SPECIAL APPLE-SPECIFIC PUBLICATION FROM THE PUBLISHERS OF *COMPUTE!*, *COMPUTE!'s Gazette*, AND BEST-SELLING COMPUTE! BOOKS

The second semiannua edition of *COMPUTE!'s Apple Applications Specia* goes on sale October 1, 198

Price: $3.95
(in Canada $4.75)

Editorial

COMPUTE!'s Apple Applications Special second issue features applications, purchasing decisions, tutorials, and in-depth feature articles for owners and users of Apple personal computers. There are exciting applications for business, school, and home. From software to hardware to the state of the industry, this special issue serves as a useful tool and a handy reference. The special issue includes:

Features

Apple at Ten, and What's Coming in the Next Decade: This in-depth look describes Apple's place in the industry and predicts what it will do in the future. Can the Macintosh Office concept succeed against IBM? How will Apple retain its position in the market when the newest round of computers—such as the Commodore Amiga and Atari ST—reaches homes and schools? This intriguing survey includes comments by computer industry analysts and software manufacturers.

Cruising MAUG: The Micronet Apple Users Group is probably the best connection any Apple owner can make. Available through CompuServe, MAUG lets Apple users communicate and exchange information and programs. This guide to MAUG describes just some of its features, and highlights programs from Macintosh desktop utilities to complete terminal software, all of which can be retrieved with a modem.

The Big Picture: Innovative hardware and software can transform the Apple II computer into a powerful graphics machine and enhance the Macintosh's already considerable abilities. Drawing programs, digitizers, and graphics tablets are featured and evaluated in this buyer's guide and tutorial.

Applications

Dr. Disk: Allows you to read from, edit, and write to any block on any disk. An excellent utility which lets you examine disk contents, manipulate catalogs, and even change machine language programs.

Enhanced Applesoft INPUT: A short machine language utility which turns Applesoft INPUT into a more flexible and powerful statement. Allows entry of any valid numeric expression, as well as commas, quotes, or colons as responses to the INPUT prompt.

The Office for Everyone: A major applications feature on using *Word, Chart, File, Multiplan,* and *MacTerminal* on the Macintosh. This tutorial shows how to turn the Macintosh into a powerful business computer.

Apple Electrotype: This simple BASIC program turns any Apple II-series computer into an electronic-style typewriter. Set margins and tabs, underline, and print out letters, memos, and notes.

Quality editorial in the style and tradition of *COMPUTE!*, the leading magazine of home, educational, and recreational computing.

There's also a disk available which includes all the programs from the magazine. The 5¼-inch floppy disk for the Apple II+, IIe, or IIc costs $16.95 and is available only from COMPUTE! Publications. Send in the attached card with your payment today to order the *Disk*.

Look for the October issue of *COMPUTE!'s Apple Applications Special* on sale where you buy other COMPUTE! publications, or at Apple computer retailers. You can also order directly from COMPUTE! Publications.

To order, send in the attached card with your payment or call toll-free 800-334-0868 (in NC 919-275-9809.)

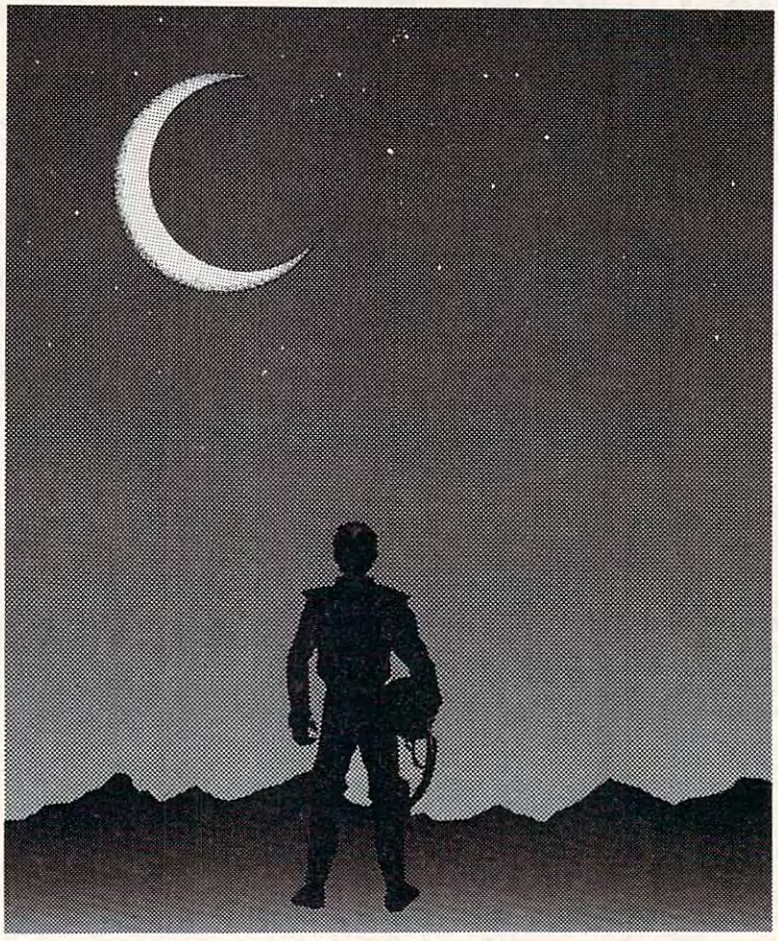
Program 8: Atari Version

Version By Patrick Parrish, Programming Supervisor

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
MG 100  NR=99:NW=100:REM NR I
        S MAX # OF ROWS,COLUM
        NS; NW IS MAX # OF WO
        RDS
MJ 110  DIM G$(NR),FF$(2*NW),
        S$(NR*NR),W$(NW*20),C
        C(NW),RR(NW),L(NW),E$
        (18),D(2,8),A$(5),X$(
        20),T$(30)
CD 120  READ E$:DATA NW NNE W
        {3 SPACES}ESW SSE
KD 130  G$=" ":G$(NR)=G$:G$(2
        )=G$:W$=" ":W$(20*NW)
        =W$:W$(2)=W$
LP 140  FOR I=1 TO 8:READ A,B
        :D(1,I)=A:D(2,I)=B:NE
        XT I:DATA -1,-1,-1,0,
        -1,1,0,-1,0,1,1,-1,1,
        0,1,1
LH 150  X$=" ":X$(20)=X$:X$(2
        )=X$:GOTO 580
DL 160  REM SHELL SORT
BJ 170  PRINT "SORTING...":X=
        1
ML 180  X=2*X:IF X<=W0 THEN 1
        80
MB 190  X=INT(X/2):IF X=0 THE
        N RETURN
BJ 200  FOR Y=1 TO W0-X:Z=Y
MM 210  A=Z+X:IF L(Z)>=L(A) T
        HEN 240
IP 220  X$=W$((Z-1)*20+1,Z*20
        ):W$((Z-1)*20+1,Z*20)
        =W$((A-1)*20+1,A*20):
        W$((A-1)*20+1,A*20)=X
        $
PB 230  B=L(Z):L(Z)=L(A):L(A)
        =B:Z=Z-X:IF Z>0 THEN
        210
DL 240  NEXT Y:GOTO 190
CE 250  REM HIDE WORDS
GP 260  FOR X=1 TO W0
HJ 270  FOR Y=1 TO 50:R1=INT(
        RND(1)*R0):C1=INT(RND
        (1)*C0):D1=INT(RND(1)
        *8)+1:O1=D1
CC 280  DX=D(1,D1):DY=D(2,D1)
        :IF R1+DX*L(X)>=1 AND
         R1+DX*L(X)<=R0 AND C
        1+DY*L(X)>=1 AND C1+D
        Y*L(X)<=C0 THEN 310
ID 290  D1=D1*(D1<8)+1:IF D1<
        >O1 THEN 280
DK 300  NEXT Y:GOTO 390
PI 310  FOR Z=1 TO L(X):IF W$
        ((X-1)*20+Z,(X-1)*20+
        Z)<"A" OR W$((X-1)*20
        +Z,(X-1)*20+Z)>"Z" TH
        EN 340
MG 320  R1=R1+DX:C1=C1+DY
MC 330  IF S$((R1-1)*C0+C1,(R
        1-1)*C0+C1)<>" " AND
        S$((R1-1)*C0+C1,(R1-1
        )*C0+C1)<>W$((X-1)*20
        +Z,(X-1)*20+Z) THEN 2
        90
GL 340  NEXT Z:FOR Z=L(X) TO
        1 STEP -1:IF W$((X-1)
        *20+Z,(X-1)*20+Z)<"A"
         OR W$((X-1)*20+Z,(X-
        1)*20+Z)>"Z" THEN 370
MN 350  S$((R1-1)*C0+C1,(R1-1
        )*C0+C1)=W$((X-1)*20+
        Z,(X-1)*20+Z)
NJ 360  RR(X)=R1:CC(X)=C1:FF$
        ((X-1)*2+1,X*2)=E$((D
        X+1)*6+(DY+1)*2+1,(DX
        +1)*6+(DY+1)*2+2):R1=
        R1-DX:C1=C1-DY
DD 370  NEXT Z
DO 380  NEXT X:GOTO 450
EK 390  PRINT "{CLEAR}Sorry,
        but I can't fit word
        number ";STR$(X);" ,
        ";W$((X-1)*20+1,X*20)
        ;" , into the grid."
LJ 400  PRINT "Should I SKip
        it, STart over, or TR
        y again":INPUT X$
GP 410  IF X$(1,2)="ST" THEN
        710
HA 420  IF X$(1,2)="TR" THEN
        270
KB 430  IF X$(1,2)<>"SK" THEN
         400
CM 440  W$((X-1)*20+1,(X-1)*2
        0+1)="/":GOTO 380
PO 450  FOR X=1 TO R0:FOR Y=1
         TO C0:IF S$((X-1)*C0
        +Y,(X-1)*C0+Y)<>" " T
        HEN 470
DH 460  S$((X-1)*C0+Y,(X-1)*C
        0+Y)=CHR$(INT(26*RND(
        1)+65))
AE 470  NEXT Y:NEXT X
KG 480  REM DONE
HG 490  PRINT :PRINT "I am fi
        nished. What do you w
        ant to call the word
        search":INPUT T$
EJ 500  SL=0:PRINT :PRINT "Do
         you want to print th
        e solution (Y/N)":GOS
        UB 550:IF A$="N" THEN
         520
KD 510  SL=1
CN 520  GOSUB 2000:F=0:PRINT
        "{CLEAR}Do you want a
        nother grid (Y/N)":GO
        SUB 550:IF A$="N" THE
        N END
IF 530  PRINT :PRINT "Do you
        want to use the same
        words (Y/N)":GOSUB 55
        0:IF A$="N" THEN 590
FH 540  F=1:GOTO 610
OB 550  INPUT A$:IF A$<>"Y" A
        ND A$<>"N" THEN 550
HL 560  RETURN
KI 570  REM INITIALIZATION
LD 580  PRINT CHR$(125):LL=6:
        GOSUB 720:PRINT "
        {12 RIGHT}WORD SEARCH
        ":LL=4:GOSUB 720
FE 590  FOR I=1 TO W0:W$((I-1
        )*20+1,I*20)=G$(1,20)
        :L(I)=0:NEXT I
MP 600  PRINT "How many words
         would you like in yo
        ur word search":INPUT
         W0
IB 610  PRINT :PRINT "How man
        y rows and columns in
         the grid":INPUT R0,C
        0:PRINT
MK 620  IF R0*C0<10*W0 THEN P
        RINT "I don't think I
         could do this.":FOR
        I=1 TO 300:NEXT I:GOT
        O 610
AO 630  PRINT "I think I can
        do this.":IF C0>NR TH
        EN PRINT "(But it won
        't fit on the paper.)
        "
KE 640  IF F=1 THEN 710
LJ 650  LL=3:GOSUB 720:PRINT
        "Enter the ";STR$(W0)
        ;" words. To correct
        a mistake, enter X":P
        RINT
GE 660  FOR I=1 TO W0
BM 670  PRINT "Word number ";
        I;":":INPUT X$:IF LEN
        (X$)<=R0 AND LEN(X$)<
        =C0 AND X$<>"X" THEN
        700
AD 680  IF X$<>"X" THEN PRINT
         "Oops...the word is
        too long.":GOTO 670
LE 690  I=I-(I>1):GOTO 670
IL 700  L(I)=LEN(X$):W$((I-1)
        *20+1,(I-1)*20+L(I))=
        X$:NEXT I:PRINT CHR$(
        125):GOSUB 170
KF 710  PRINT "{DOWN}Okay, I
        will go to work. Wish
         me luck!":FOR I=1 TO
         R0:S$((I-1)*C0+1,I*C
        0)=G$:NEXT I:GOTO 260
BF 720  FOR I=1 TO LL:PRINT :
        NEXT I:RETURN
AK 1999 REM PRINTER ROUTINE
CI 2000 TRAP 2190:OPEN #1,8,
        0,"P:":PRINT #1;T$:P
        RINT #1
HP 2010 PRINT #1;"
        {3 SPACES}";:FOR I=1
         TO C0:IF I/10<>INT(
        I/10) THEN PRINT #1;
        " ";:GOTO 2030
CA 2020 X$=STR$(I):PRINT #1;
        X$(1,1);
GI 2030 NEXT I:PRINT #1
LI 2040 PRINT #1;"
        {3 SPACES}";:FOR I=1
         TO C0:X$=STR$(I):PR
        INT #1;X$(LEN(X$),LE
        N(X$));:NEXT I:PRINT
         #1
CB 2050 FOR X=1 TO R0:IF X<1
        0 THEN PRINT #1;" ";
GE 2060 PRINT #1;STR$(X);" "
        ;
GC 2070 FOR Y=1 TO C0:PRINT
        #1;S$((X-1)*C0+Y,(X-
        1)*C0+Y);
NA 2080 NEXT Y:PRINT #1:NEXT
         X:PRINT #1:PRINT #1
        :PRINT #1;"WORD LIST
        :"
OE 2090 FOR X=1 TO W0:IF W$(
        (X-1)*20+1,(X-1)*20+
        1)="/" THEN 2110
KK 2100 PRINT #1;W$((X-1)*20
        +1,X*20)
LH 2110 NEXT X:FOR I=1 TO 5:
        PRINT #1:NEXT I:IF S
        L=0 THEN 2180
JD 2120 PRINT #1;"SOLUTION L
        IST:":PRINT #1;"WORD
        {21 SPACES}ROW
        {3 SPACES}COLUMN
        {3 SPACES}DIR"
OF 2130 FOR X=1 TO W0:IF W$(
        (X-1)*20+1,(X-1)*20+
        1)="/" THEN 2170
PL 2140 PRINT #1;W$((X-1)*20
        +1,X*20);G$(1,6);RR(
        X);
HH 2150 PRINT #1;G$(1,9-LEN(
        STR$(RR(X))));CC(X);
        G$(1,6-LEN(STR$(CC(X
        ))));
IF 2160 PRINT #1;FF$((X-1)*2
        +1,X*2)
GB 2170 NEXT X
BE 2180 CLOSE #1:TRAP 40000:
        RETURN
EB 2190 CLOSE #1:TRAP 40000:
        PRINT "Turn on your
        printer--press RETUR
        N":INPUT X$:GOTO 200
        0
```

THE LAST WARRIOR



David Engebretsen

This arcade-style action game was originally written for the IBM PC (with BASICA and color/graphics adapter) and PCjr (with Cartridge BASIC). We've added adaptations for the Commodore 64; Atari 400/800/ XL/XE series (with at least 16K RAM for tape or 24K RAM for disk); and Apple II series. A joystick is required for all versions except the Apple. The Commodore 64 and Atari programs are written completely in machine language.

"Attacked by countless alien ships . . ."

You're the last member of the scouting party sent from Earth. While flying a routine mission, you and your fellow scouts were suddenly attacked by countless alien ships. Your comrades put up a good fight but couldn't survive in the face of the aliens' nonstop shooting. Now the only things between you and utter destruction are your highly advanced force shields and lasers. The aliens may not be as well armed, but they make up for it in sheer numbers. As you blast yet another hostile ship, it is immediately replaced, and your energy supply dwindles....

"The Last Warrior," as you've guessed, is a space shoot-em-up game. The classic object is to destroy as many aliens as possible before they destroy you. Your performance is graded at the end of the game by the number of points you score and by rank: captain, major, colonel, general, or warrior. Scoring and a few other details vary from version to version, but all the programs have one thing in common—the highest ranks are attainable only by the very best players.

IBM Version

After typing the program and saving at least one copy on disk, plug in a joystick and type RUN. Your starfighter appears on the screen, and the program asks you to move the stick to the upper-left corner and press the fire button. Next you're asked to move the stick to the lower-right corner and press the button again. This calibrates the program with your joystick, since different sticks tend to yield different values. (You may also prefer to flip the switches on the bottom of the controller to free the stick from its self-centering mode.)

When the game begins, you find yourself looking out of the front cockpit window at a star field. Below the window is an instrument panel, and an aiming sight floats somewhere on the screen. By maneuvering the sight with the joystick, you can aim your lasers at the alien ships which suddenly appear in view. Press the joystick button to fire shots as the aliens make their passes. With any luck, you'll witness a brilliant explosion as the alien attacker is reduced to stardust. But more aliens soon appear to take his place (up to three at a time), and the battle continues.

Don't fire your lasers indiscriminately, because each shot burns up energy, as indicated by the lower horizontal bar on the instrument panel. This bar shortens toward the left side of the screen as your energy decreases. Alien hits on your force shields also sap energy. The upper horizontal bar on the instrument panel shows the relative number of points you've scored. When this bar goes off the scale toward the right, you advance one rank and the bar starts again at the left. Your rank is constantly displayed on the panel and starts at captain.

The game ends when your ship runs out of energy. Your final rank and score appear on the screen—a higher rank with few points is considered better than a lower rank with many points. Press the joystick button to start another game.

The IBM version of The Last Warrior is written entirely in BASIC and animates the aiming sight and alien ships with the PUT statement. To reduce flickering, one set of variables stores the existing positions of the images while another set holds the new positions. That way, when the program erases an existing image, it can draw the new one immediately without pausing to update the variables. As a result, flickering is hardly noticeable, especially when the program runs on the PC (which is faster than the PCjr).

64 Version

Written entirely in machine language, the 64 version of The Last Warrior must be typed with the "MLX" machine language entry utility found elsewhere in this issue. MLX makes it much easier to enter machine language programs without typos. Be sure you read and understand the instructions for using MLX before entering the data from Program 2.

When you run MLX, you'll be asked for the starting and ending addresses of the program to be entered. For The Last Warrior, the values are:

```
STARTING ADDRESS?   49152
ENDING ADDRESS?     51811
```

If you enter the data from Program 2 in more than one sitting, be sure to use these same values whenever you reload your partially completed work.

After you've finished entering the data and saved at least one copy of the game on disk or tape, load it by typing LOAD"*filename*",8,1 for disk or LOAD"*filename*",1,1 for tape (replace *filename* with whatever name you used for your final version). Next type SYS 49152 and press RETURN. Then plug a joystick into port 2 and push the joystick up to start.

The screen shows the front view from the cockpit with alien ships appearing in the distance against the star field. As the aliens get closer, their ships seem to grow larger. Up to five of them can attack you at once. Move the joystick to aim the floating crosshair and press the button to fire your lasers. Each hit scores 100 points.

The instrument panel at the bottom of the screen shows the level of your ship's shield energy, the number of points you've scored, and a special targeting scope. When the game begins, the energy indicator is set at 5,000 units. Each laser shot you fire depletes the shield energy by 20 units. Alien hits cost 100 units of shield energy. When the energy indicator drops to zero, your shields collapse, leaving you completely vulnerable. The next alien hit will destroy your ship and end the game. At this point, you might as well shoot like crazy, since you're out of shield energy anyway.

To help you hit distant ships, the targeting scope on the instrument panel alerts you when your aiming sight has locked onto an alien. If you press the fire button at this instant, you're guaranteed a direct hit.

When the game ends, the program displays your final score and

rank, then waits for you to push the joystick up to start another game. During a game, you can freeze the action by pressing any key, and continue playing by pressing another key.

The 64 version of The Last Warrior uses the multicolor high-resolution graphics screen and all eight sprites for the aiming crosshair, explosion effects, targeting scope image, and maximum of five alien vessels.

Atari Version

Like the 64 version, the Atari adaptation of The Last Warrior is written entirely in machine language and must be typed with the MLX entry utility found elsewhere in this issue. MLX greatly reduces the chances of typos when entering long machine language programs. Be sure you read the instructions and understand how to use Atari MLX before entering data from Program 3.

When you run the MLX program, you'll be asked for starting, ending, and run/init addresses. For The Last Warrior, the proper values are:

```
STARTING ADDRESS?    8192
ENDING ADDRESS?      10249
RUN/INIT ADDRESS?    8192
```

If you enter the data from Program 3 in more than one sitting, be sure to use these same values whenever you reload your partially completed work. You'll then be asked whether you wish to create a boot tape, a boot disk, or a disk binary file. For The Last Warrior, you can choose any of these three. However, you should avoid the binary file option if you are not familiar with the procedure for loading and executing such files.

After you finish entering the data from Program 3, and you've saved at least one copy of The Last Warrior on disk or tape, start the program by loading the boot disk or boot tape or running the binary file created with MLX. For a boot disk, simply insert the disk in the drive and switch on the computer after removing the BASIC cartridge (on a 600XL, 800XL, or XE-series computer, hold down the OPTION button while turning on the machine). To run a boot tape, switch on the computer while holding down the START button (again, remove the BASIC cartridge with a 400, 800, or 1200XL, or simultaneously hold down START and OPTION with a 600XL, 800XL, or XE). Then press the PLAY button on the cassette recorder and hit RETURN. If you used MLX to save the program as a binary disk file, load it with the binary load option in DOS and run at hex address 2000 (decimal 8192).

Plug a joystick into port 1 and press the fire button to start. The screen shows the front view from your ship's cockpit window. Alien vessels first appear as distant dots against the star field, then grow larger as they approach. Their weapons are limited, so they can start shooting at you only at point-blank range. But you can shoot them at any point during their attack. For every alien ship you destroy, you score 100 points; for each hit they make on your energy shield, you lose 100 points of shield energy. You begin the game with 5,000 units of energy, and every shot you fire uses 20 units. (All of this information is indicated on the screen's instrument panel.) You can pause and then continue a game in progress by pressing any key.

All the animation in the Atari version of The Last Warrior is driven by a vertical blank interrupt routine—objects are moved during the split-second interval when the TV's electron beam returns from the lower-right corner of the screen to the upper-left corner to scan another frame. Player/missile graphics are used for the crosshair and alien ships, so no more than three aliens can appear at once. Alien ships actually consist of six separate images which are flipped in succession to create the illusion of an approaching object. The program employs a custom display list to put GRAPHICS 7 at the top of the screen and GRAPHICS 1 at the bottom. The ship's cockpit window is not plotted with the Atari's built-in line-drawing routines, but rather with custom-designed routines which are faster and do not destroy the screen background. Otherwise, laser shots would gradually erase the lines representing the cockpit window.

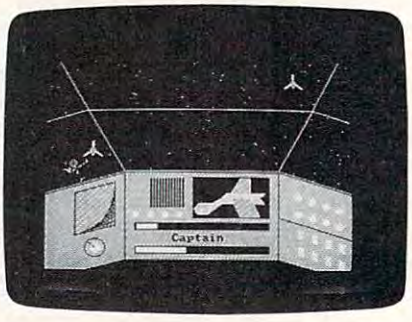
Apple Version

Like the IBM program, the Apple adaptation of The Last Warrior is written in BASIC. However, it does use the HROUT machine language character-plotting routine from "Apple SuperFont" (COMPUTE!, April 1985). All of the alien ships are custom characters created with SuperFont and plotted onto the hi-res graphics screen. The aiming crosshair is drawn with shape tables.

The keyboard controls are programmed in the efficient upside-down T arrangement: I for up, K for down, J for left, and L for right. This is more convenient than the usual I-J-K-M diamond, because you can rest your first three fingers on J-K-L and quickly move your middle finger up and down between I and K.

To fire a laser shot, press the space bar. Press P to pause a game, and press it again to continue.

An instrument panel at the bottom of the cockpit window displays all the important information: points scored (100 for each alien ship you destroy), units of shield energy remaining (the game begins with 5,000), and your current rank. Enemy hits reduce shield energy by 100 units, and your own laser shots cost 20 units each.



An alien ship explodes near the cockpit window while another zooms in for attack in the IBM version of "The Last Warrior."

Program 1: The Last Warrior, IBM Version

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
HB 20 SCREEN 1:COLOR 0,0:CLS:KEY
      OFF:RANDOMIZE TIMER:PLAY"
      mb":STRIG ON
CL 30 DIM SIGHT%(20),SHIP%(50),I
      NFI%(404),HA%(50),HB%(60),
      HC%(105),INVER%(100)
FE 40 REM ** get the images
JJ 50 CIRCLE(5,5),3,,,,1:LINE(3,
      3)-(4,4):LINE(7,3)-(6,4):L
      INE(7,7)-(6,6):LINE(3,7)-(
      4,6):GET(2,2)-(8,8),SIGHT%
      :CLS
NJ 60 CIRCLE(10,10),10,2:PAINT(1
      0,10),2,2:GET(0,0)-(20,20)
```

With NRI training at home, you can . . .

Move up to a high paying career servicing computers

And you can start by actually building NRI's 16-bit IBM-compatible computer.

You can create your own bright, high paying future as an NRI trained computer service technician. The biggest growth in jobs between now and 1995, according to Department of Labor predictions, will occur in computer service and repair, where demand for trained technicians will double. There is still plenty of room for you to get in on the action—if you get the proper training now.

Total computer systems training, only from NRI

To learn how to work on computers, you have to get inside one. And only NRI takes you inside a computer, with total systems training that gives you hands-on experience with computers, peripherals, and software. As part of your training, you'll build a Sanyo MBC-550-2, which experts have hailed as the "most intriguing" of all the new IBM-compatibles.

Even if you've never had any previous training in electronics, you can succeed with NRI training. You'll start with the basics, rapidly building on the fundamentals of electronics until you master advanced concepts like digital logic, microprocessor design and computer memory. You'll probe into electronic circuits, using the exclusive NRI Discovery Lab® and professional Digital Multimeter, that you keep.

Learn to service today's computers

You'll assemble Sanyo's intelligent keyboard, install the power supply and disk drive, and attach the high resolution monitor—all the while performing hands-on experiments and demonstrations that reinforce your skills.

As you complete your Sanyo, you grasp the "secrets" that qualify you for a new career. You'll learn to program in BASIC and machine language. You'll use utility programs to check out the Sanyo 8088 microprocessor (the same chip used in the IBM PC). And you also get over $1,000 worth of software, including WordStar and CalcStar.

Learn the basics at home

Most importantly, you'll understand the principles common to all computers. Only a person who fully understands all the fundamentals can hope to be able to tackle all computers. NRI makes sure that you'll gain the knowledge and skills to maintain, troubleshoot and service computers.

With NRI training, you'll learn at home on your own time. That means your preparation for a new career or part-time job doesn't have to interfere with your current job. You'll learn at your own pace, in the comfort and convenience of your own home. No classroom pressures, no rigid night school schedules. You're always backed up by the NRI staff and your instructor, who will answer questions, give you guidance and be available for special help if you need it.

Send for free NRI catalog

Let others worry about computers taking their jobs. With NRI training, you'll soon have computers making good paying jobs for you. Send the coupon today for NRI's 100-page catalog, with all the facts about computer training. If the coupon is missing, write to NRI Schools, 3939 Wisconsin Ave., Washington, D.C. 20016.

NRI

McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue, Washington, DC 20016
We'll give you tomorrow.

For Career courses approved under GI bill, ☐ check for details.

☐ **CHECK ONE FREE CATALOG ONLY**

- ☐ **Computer Electronics with Microcomputers**
- ☐ Data Communications
- ☐ Robotics & Industrial Controls
- ☐ Color TV, Audio, and Video System Servicing
- ☐ Electronic Design Technology
- ☐ Digital Electronics
- ☐ Communications Electronics
- ☐ Industrial Electronics
- ☐ Basic Electronics
- ☐ Telephone Servicing
- ☐ Small Engine Servicing
- ☐ Appliance Servicing
- ☐ Automotive Servicing
- ☐ Air Conditioning, Heating, Refrigeration, & Solar Technology
- ☐ Building Construction
- ☐ Locksmithing & Electronic Security

Name (Please Print) Age

Street

City/State/Zip Accredited by the National Home Study Council 198-095

```
      ,INFI%:CLS
JA 70 LINE(0,0)-(60,8),3,BF:GET(
      0,0)-(60,8),INVER%:CLS
ML 80 FOR LOOP=0 TO 50:READ SHIP
      %(LOOP):NEXT
ID 90 FOR LOOP=0 TO 50:READ HA%(
      LOOP):NEXT
II 100 FOR LOOP=0 TO 60:READ HB%
      (LOOP):NEXT
MK 110 FOR LOOP=0 TO 105:READ HC
      %(LOOP):NEXT
LI 120 REM ** set up the screen
OL 130 GOSUB 880
BL 140 SN=1:SX(1)=160:SY(1)=50:S
      XA(1)=SX(1):SYA(1)=SY(1):
      DLA=1:RANK =0:ENE=139:SCO
      =0
FB 150 GOSUB 1370
JP 160 SN=1:SX(1)=160:SY(1)=50:S
      XA(1)=SX(1):SYA(1)=SY(1):
      DLA=1
IH 170 PUT(127,167),INVER%,PRESE
      T:LOCATE 22,17:PRINT"Capt
      ain":PUT(127,167),INVER%
IC 180 XA=0:YA=0:PUT(XA,YA),SIGH
      T%:PUT(SX(1),SY(1)),SHIP%
MM 190 REM ** main program loop
KA 200 GOSUB 290
KP 210 GOSUB 560
PP 220 IF STRIG(0)=-1 THEN GOSUB
       380:V=STRIG(0)
HE 230 IF RND(1)<.2 THEN PSET(32
      0*RND(1),110*RND(1)),3*RN
      D(1)+1
KA 240 IF EC>0 THEN GOSUB 1110
CN 250 DLA=DLA+.01:DL=INT(DLA)
BD 260 GOTO 200
MJ 270 END
CJ 280 REM ** JOYSTICK
BF 290 X=STICK(0):Y=STICK(1):X=X
      -JSX1:Y=Y-JSY1:X=X*TFX:Y=
      Y*TFY
FC 300 IF X<0 THEN X=0
HF 310 IF X>313 THEN X=313
HP 320 IF Y<0 THEN Y=0
DF 330 IF Y>103 THEN Y=103
DC 340 IF X=0 AND Y=0 THEN X=XA:
      Y=YA
KE 350 PUT(XA,YA),SIGHT%:PUT(X,Y
      ),SIGHT%:XA=X:YA=Y
NJ 360 RETURN
FJ 370 REM ** fire !!
FI 380 PUT(X,Y),SIGHT%
IH 390 FOR P=1 TO SN:PUT(SX(P),S
      Y(P)),SHIP%:NEXT
FH 400 LINE(0,110)-(X+3,Y+3),2:L
      INE(319,110)-(X+3,Y+3),2
MJ 410 LINE(0,110)-(X+3,Y+3),0:L
      INE(319,110)-(X+3,Y+3),0
FJ 420 LINE(0,130)-(80,110):LINE
      -(240,110):LINE-(319,130)
GA 430 LINE(0,60)-(41,50):LINE-(
      280,50):LINE-(319,60)
KA 440 LINE(80,110)-(10,0):LINE(
      240,110)-(310,0)
FO 450 IF SX(LOOP)>290 THEN SX(L
      OOP)=290
IC 460 FOR P=1 TO SN:PUT(SX(P),S
      Y(P)),SHIP%:NEXT
EH 470 PUT(X,Y),SIGHT%
JC 480 PLAY"l64 t255 bagfedc <ba
      gfedc>"
NP 490 SNA=SN
QO 500 FOR LOOP=1 TO SNA
PM 510 IF ABS((X+3)-(SX(LOOP)+10
      ))<5 AND ABS((Y+3)-(SY(LO
      OP)+9))<5 THEN EC=EC+1:EX
      (EC)=SX(LOOP):EY(EC)=SY(L
      OOP):DC(EC)=0:SN=SN-1:PUT
      (SX(LOOP),SY(LOOP)),SHIP%
      :FOR L=LOOP TO 3:SX(L)=SX
      (L+1):SY(L)=SY(L+1):SYA(L
      )=SY(L):SXA(L)=SX(L):NEXT
       L:GOSUB 1220
NF 520 NEXT
KK 530 ENE=ENE-1:IF ENE<=0 THEN
      GOSUB 1500 ELSE LINE(91+E
      NE,180)-(91+ENE,184),0
MH 540 RETURN
IP 550 REM ** enemy ships
IA 560 IF RND(1)<.9 THEN GOTO 60
      0
BP 570 IF SN<3 THEN SN=SN+1:SX(S
      N)=INT(290*RND(1)):SY(SN)
      =INT(100*RND(1)):PUT(SX(S
      N),SY(SN)),SHIP%:SXA(SN)=
      SX(SN):SYA(SN)=SY(SN):GOT
      O 600
KH 580 IF SN=0 THEN RETURN
BM 590 IF RND(1)>.5 THEN PUT(SX(
      SN),SY(SN)),SHIP%:SN=SN-1
      :IF SN<0 THEN SN=0
LK 600 FOR LOOP=1 TO SN
KG 610 GOSUB 290
JP 620 IF RND(1)>.95 THEN MX(LOO
      P)=INT(10*RND(1)-5):MY(LO
      OP)=INT(10*RND(1)-5)
MC 630 SX(LOOP)=SX(LOOP)+MX(LOOP
      ):SY(LOOP)=SY(LOOP)+MY(LO
      OP)
IK 640 IF ABS((X+3)-(SX(LOOP)+10
      ))<3 AND ABS((Y+3)-(SY(LO
      OP)+9))<3 THEN MX(LOOP)=-
      MX(LOOP):IF RND(1)<.5 THE
      N MY(LOOP)=-MY(LOOP)
OL 650 IF SX(LOOP)<2 OR SX(LOOP)
      >250 THEN MX(LOOP)=-MX(LO
      OP):SX(LOOP)=SX(LOOP)+MX(
      LOOP)
DG 660 IF SY(LOOP)<2 OR SY(LOOP)
      >85 THEN MY(LOOP)=-MY(LOO
      P):SY(LOOP)=SY(LOOP)+MY(L
      OOP)
CC 670 IF SX(LOOP)<0 THEN SX(LOO
      P)=0
FG 680 IF SX(LOOP)>290 THEN SX(L
      OOP)=290
FI 690 IF SY(LOOP)<0 THEN SY(LOO
      P)=0
PI 700 PUT(SXA(LOOP),SYA(LOOP)),
      SHIP%:PUT(SX(LOOP),SY(LOO
      P)),SHIP%:SXA(LOOP)=SX(LO
      OP):SYA(LOOP)=SY(LOOP)
NF 710 NEXT
CA 720 IF RND(1)<(DL/20)+SN/10-.
      1 AND SN>0 THEN GOSUB 750
MH 730 RETURN
JD 740 REM ** enemy fire
BE 750 SNB=INT(SN*RND(1)+1)
BK 760 HX=INT(300*RND(1)):HY=INT
      (85*RND(1)):PUT(X,Y),SIGH
      T%
IH 770 FOR P=1 TO SN:PUT(SX(P),S
      Y(P)),SHIP%:NEXT
MF 780 PUT(HX,HY),INFI%:LINE(HX+
      10,HY+2)-(SX(SNB)+10,SY(S
      NB)+12),2:LINE-(HX+10,HY+
      18),2
GE 790 COLOR 4:PUT(HX,HY),INFI%:
      LINE(HX+10,HY+2)-(SX(SNB)
      +10,SY(SNB)+12),0:LINE-(H
      X+10,HY+18),0
OD 800 LINE(0,130)-(80,110):LINE
      -(240,110):LINE-(319,130)
      :COLOR 0
GA 810 LINE(0,60)-(41,50):LINE-(
      280,50):LINE-(319,60)
KA 820 LINE(80,110)-(10,0):LINE(
      240,110)-(310,0)
OM 830 FOR TIM=180 TO 20 STEP-4:
      SOUND 255-TIM,.1:NEXT
KP 840 PUT(X,Y),SIGHT%:FOR P=1 T
      O SN:PUT(SX(P),SY(P)),SHI
      P%:NEXT
HB 850 ENE=ENE-4:IF ENE<=0 THEN
      GOSUB 1500 ELSE LINE(91+E
      NE,180)-(229,184),0,BF
NO 860 RETURN
NP 870 REM ** THE SHIP
FL 880 FOR LOOP=1 TO 150:PSET(32
      0*RND(1),130*RND(1)),3*RN
      D(1)+1:NEXT
GL 890 LINE(0,130)-(80,110):LINE
      -(240,110):LINE-(319,130)
GP 900 LINE(0,60)-(41,50):LINE-(
      280,50):LINE-(319,60)
KP 910 LINE(80,110)-(10,0):LINE(
      240,110)-(310,0)
NA 920 LINE(40,199)-(80,190):LIN
      E-(240,190):LINE-(280,199
      )
MC 930 LINE(150,116)-(230,153),0
      ,BF:LINE(149,115)-(231,15
      4),,B
OE 940 PAINT(160,180),3,3
NL 950 LINE(0,131)-(80,111),0:LI
      NE-(240,111),0:LINE-(319,
      131),0:LINE(80,111)-(80,1
      99),0:LINE(240,111)-(240,
      199),0
EN 960 LINE(90,179)-(230,185),0,
      BF:LINE(91,180)-(229,184)
      ,1,BF
DC 970 LINE(90,158)-(230,164),0,
      BF
OI 980 LINE(151,145)-(156,140),1
      :LINE-(170,140),1:LINE-(1
      80,135),1:LINE-(185,131),
      1:LINE-(225,131),1:LINE-(
      220,135),1:LINE-(225,140)
      ,1:LINE-(180,140),1
NN 990 LINE-(165,150),1:LINE-(15
      5,150),1:LINE-(151,145),1
      :LINE-(163,145),1:LINE-(1
      68,140),1
GE 1000 LINE (190,131)-(200,117)
      ,1:LINE-(210,117),1:LINE
      -(210,131),1:LINE(190,13
      5)-(210,135),1:LINE-(220
      ,152),1:LINE-(200,152),1
      :LINE-(190,135),1:LINE(1
      94,140)-(212,140),0
FE 1010 PAINT(155,143),3,1:PAINT
      (170,145),CHR$(&H77)+CHR
      $(&HDD),1:PAINT(210,145)
      ,CHR$(&H11)+CHR$(&H44),1
      :PAINT(205,120),CHR$(&H6
      6)+CHR$(&H99),1
KM 1020 FOR LOOP=90 TO 140 STEP
      15:CIRCLE(LOOP,150),3,1:
      PAINT(LOOP,150),1,1:NEXT
II 1030 LINE(105,143)-(140,117),
      0,BF:FOR LOOP=105 TO 140
       STEP 3:LINE(LOOP,143)-(
      LOOP,117),3:NEXT
DK 1040 LO=160:FOR LOOP=70 TO 30
       STEP -4:LO=LO+.8:LINE(L
      OOP,LO)-(70,120+(70-LOOP
      )),0:NEXT:LINE(30,LO)-(3
      0,130),0:LINE-(70,120),0
      :PAINT(50,140),CHR$(&H66
      )+CHR$(&H99),0
JG 1050 CIRCLE(50,180),5,1:PAINT
      (50,180),1,1:LINE(50,180
      )-(43,175),0:CIRCLE(50,1
      80),10,0
EN 1060 LO=130:FOR LOOPA=1 TO 2:
      FOR LOOP=260 TO 310 STEP
       15:LO=LO+4:CIRCLE(LOOP,
      LO),4,1:PAINT(LOOP,LO),1
      ,1:NEXT LOOP:LO=145:NEXT
       LOOPA
KP 1070 LINE(240,153)-(319,173),
      0
HM 1080 LO=160:FOR LOOPA=1 TO 2:
      FOR LOOP=260 TO 310 STEP
       15:LO=LO+4:LINE(LOOP,LO
      )-(LOOP+6,LO+1),1:LINE-(
      LOOP+6,LO+8),1:LINE-(LOO
      P,LO+7),1:LINE-(LOOP,LO)
      ,1:PAINT(LOOP+2,LO+2),1,
      1:NEXT LOOP:LO=175:NEXT
      LOOPA
JG 1090 RETURN
MP 1100 REM ** explosion
```

OWN A MOVIE FOREVER FOR THE COST OF SEEING IT ONCE.

JUST $4.95.

WITH MEMBERSHIP

Now you can own any movie here for just $4.95 with membership in the CBS Video Club! That's less than the price of a single movie ticket, less than the price of most rentals, and it's yours for keeps!

Choose from the best, too. *The Empire Strikes Back, Romancing the Stone, Splash* and more. There's no membership fee, and you don't have to buy a lot of movies.

Just two more within the next year. The movies you order will be mailed and billed at regular Club prices which currently range from $39.95 to $79.95 per cassette—plus shipping and handling. (Extra-long films and specials may cost a bit more.)

BONUS PLAN—SAVE 50%

After buying two movies at regular Club prices in the next year, you can cancel. Or stay with us and save even more under our Bonus Plan. With each movie you buy, the plan currently allows you to help yourself to another movie of equal value or less at 50% off. And, right now, save up to $50 more—see the Advance Bonus box above.

About every four weeks (up to 13 times a year) we send you our CBS Video Club Magazine, reviewing our Director's Selection plus many alternate movies.

CHOOSE FROM HUNDREDS OF HIT MOVIES

If you want the Director's Selection, don't do a thing. It will arrive automatically. If you prefer an alternate title, or none at all, just return the card always provided by the date specified.

You'll always have two full weeks to decide. And a toll-free number to call if you have any questions or service requests. (If you ever receive a tape that you had less than two weeks to consider, send it back at our expense.)

Join today and we'll send your movie for just $4.95, along with details of how the Club works. If you're not satisfied, return everything within 10 days for a full, prompt refund—no further obligation.

For faster service, use your credit card and our toll-free number to order. Just call **1-800-457-0866** (in Indiana 1-800-742-1200). Or mail coupon.

Advance Bonus:

SAVE UP TO $50 MORE!

. . . by ordering a second movie right now. Any movie listed in this ad—yours for just $29.95 plus shipping and handling on videocassette. See coupon below.

60 TOP HITS TO CHOOSE FROM

TITLE	SELECTION NUMBER	TITLE	SELECTION NUMBER	TITLE	SELECTION NUMBER
THE EMPIRE STRIKES BACK	0910092	JANE FONDA'S WORKOUT CHALLENGE	5260042	ROBIN HOOD (Walt Disney)	5299092
ROMANCING THE STONE	0894092	CASABLANCA	0507082	EXCALIBUR	6021022
SPLASH	5304022	TOOTSIE	1509042	TIGHTROPE	6051052
THE BIG CHILL	1527022	DUMBO	5251052	BACHELOR PARTY	0926012
WARGAMES	0828002	THE AFRICAN QUEEN	0511022	COUNTRY	5341072
RISKY BUSINESS	6033082	ON GOLDEN POND	0523082	BODY DOUBLE	1713062
THE NATURAL	1649052	THE LONGEST DAY	0577032	REVENGE OF THE NERDS	0925022
STAR WARS	0564162	DIRTY HARRY	6017082	GREYSTOKE—THE LEGEND OF TARZAN, LORD OF THE APES	6045042
YENTL	0895082	STRIPES	1513082	EDUCATING RITA	1593012
COTTON CLUB	3100032	FUNNY GIRL	1511002	THE ROAD WARRIOR	6028052
CADDYSHACK	6023022	CHRISTINE	1580062	SUPERMAN III	6040092
MAKING MICHAEL JACKSON'S THRILLER	7103012	OCTOPUSSY	0856052	TWILIGHT ZONE—The Movie	6034072
KING KONG (The Original)	5502022	PORKY'S	0775112	ANNIE	1516052
POLICE ACADEMY	6049002	CLOSE ENCOUNTERS OF THE THIRD KIND—Special Edition	1510012	THE MUPPETS TAKE MANHATTAN	0923042
ARSENIC & OLD LACE	0735102	THE RIGHT STUFF	6043062	ALIEN	0002322
THE COMANCHEROS	0762242	NATIONAL LAMPOON'S VACATION	6039022	THE MAGNIFICENT SEVEN	0534212
KARATE KID	1710092	HIGH ROAD TO CHINA	6022012	THUNDERBALL	0709042
PRIVATE BENJAMIN	6018072	ARTHUR	6024092	BUTCH CASSIDY & THE SUNDANCE KID	0517302
SHE WORE A YELLOW RIBBON	5504002	ROOSTER COGBURN	1018082	PURPLE RAIN	6048012
RED RIVER	7507032	NEVER SAY NEVER AGAIN	6042072	THE MALTESE FALCON	0508072

CBS VIDEO CLUB

1400 North Fruitridge Avenue, Terre Haute, IN 47811

CBS VIDEO CLUB
Dept. 62E, P.O. Box 1111, Terre Haute, IN 47811

Yes, please enroll me in the CBS Video Club under the terms outlined in this advertisement. As a member, I need buy just two more movies at regular Club prices within the next year.

Check one: ☐ **BETA** ☐ **VHS**

Send me movie #_____ **for $4.95**

Please check how paying:

☐ My check is enclosed. Z54/Z56

☐ Charge my introductory movie(s) and future Club purchases to: Z55/Z57

☐ MasterCard ☐ Diners Club
☐ American Express ☐ VISA

Account #_____

Expiration Date_____

Signature_____

☐ **Also send me as my Advance Bonus:**

movie #_____ for $29.95 on videocassette plus $3.00 shipping and handling which I'm adding to my above payment.

Name_____

Address_____

City_____ State_____

Zip_____ Phone (_____)_____

Note: CBS Video Club reserves the right to reject any application or cancel any membership. Offer limited to continental U.S. (excluding Alaska). **Applicable sales tax added to all orders.**

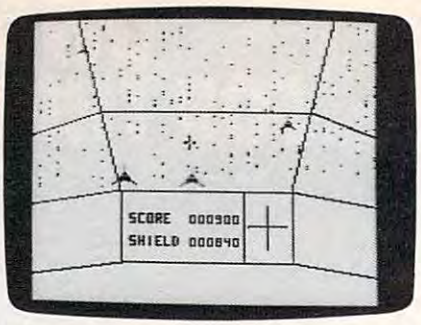
```
MJ 1110 ECA=EC
PB 1120 FOR CO=1 TO EC
QN 1130 IF EX(CO)=0 AND EY(CO)=0
        THEN GOTO 1190
AF 1140 IF DC(CO)=0 THEN PUT(EX(
        CO),EY(CO)),HA%
BN 1150 IF DC(CO)=1 THEN PUT(EX(
        CO),EY(CO)),HA%:PUT(EX(C
        O),EY(CO)),HB%
JN 1160 IF DC(CO)=2 THEN PUT(EX(
        CO),EY(CO)),HB%:PUT(EX(C
        O),EY(CO)),HC%
CB 1170 DC(CO)=DC(CO)+1
DD 1180 IF DC(CO)=4 THEN PUT(EX(
        CO),EY(CO)),HC%:EC=EC-1:
        FOR LO=CO TO EC+1:DC(LO)
        =DC(LO+1):EX(LO)=EX(LO+1
        ):EY(LO)=EY(LO+1):NEXT L
        O:DC(EC+1)=0:EX(EC+1)=0:
        EY(EC+1) =0
BJ 1190 NEXT
IP 1200 RETURN
AE 1210 REM ** scoring for a hit
LI 1220 ENE=ENE+8:IF ENE>139 THE
        N ENE=139
KE 1230 LINE(90+ENE,180)-(83+ENE
        ,184),1,BF:SCO=SCO+3
EC 1240 IF SCO>=136 THEN GOSUB 1
        280
CD 1250 LINE(89+SCO,159)-(91+SCO
        ,163),1,BF
JB 1260 RETURN
CE 1270 REM ** promotion
MA 1280 LINE(90,158)-(230,164),0
        ,BF:SCO=3:PUT(127,167),I
        NVER%,PRESET
QP 1290 RANK=RANK+1
AC 1300 IF RANK=1 THEN LOCATE 22
        ,19:PRINT"Major"
JC 1310 IF RANK=2 THEN LOCATE 22
        ,17:PRINT"Colonel"
FL 1320 IF RANK=3 THEN LOCATE 22
        ,17:PRINT"General"
NF 1330 IF RANK=>4 THEN LOCATE 2
        2,17:PRINT"Warrior"
CM 1340 PUT(127,167),INVER%
JA 1350 RETURN
BK 1360 REM ** title page
BD 1370 A=STRIG(0)
FA 1380 LOCATE 5,13:PRINT "The L
        ast Warrior"
IE 1390 LOCATE 8,12:PRINT"Move t
        he joystick":LOCATE 9,9:
        PRINT"to the upper-left
        corner ":LOCATE 10,12:PR
        INT"and press button"
CH 1400 IF STRIG(0)=-1 THEN JSX1
        =STICK(0):JSY1=STICK(1):
        A=STRIG(0) ELSE GOTO 140
        0
FJ 1410 FOR WAI=1 TO 800:NEXT WA
        I
NA 1420 LOCATE 9,9:PRINT"to the
        lower-right corner"
IH 1430 IF STRIG(1)=-1 THEN JSX2
        =STICK(0):JSY2=STICK(1)
        ELSE GOTO 1430
HB 1440 IF JSX2<=JSX1 OR JSY2<JS
        Y1 THEN GOTO 1390
GJ 1450 LOCATE 8,12:PRINT SPC(18
        ):LOCATE 9,9:PRINT SPC(2
        5):LOCATE 10,12:PRINT SP
        C(18):LOCATE 5,13:PRINT
        SPC(16):DL=1
GC 1460 TFX=ABS(313/(JSX1-JSX2))
        :TFY=ABS(103/(JSY1-JSY2)
        )
BF 1470 A=STRIG(0)
KL 1480 RETURN
PO 1490 REM ** end
EI 1500 LINE(91,180)-(229,184),0
        ,BF
OF 1510 LOCATE 5,16:PRINT"Game O
        ver"
DA 1520 IF RANK=0 THEN LOCATE 8,
        14:PRINT"Rank: Captain"
LJ 1530 IF RANK=1 THEN LOCATE 8,
        15:PRINT"Rank: Major"
IJ 1540 IF RANK=2 THEN LOCATE 8,
        14:PRINT"Rank: Colonel"
PM 1550 IF RANK=3 THEN LOCATE 8,
        14:PRINT"Rank: General"
PF 1560 IF RANK=>4 THEN LOCATE 8
        ,14:PRINT"Rank: Warrior"
HP 1570 LOCATE 9,16:PRINT"Points
        :";INT(SCO/1.36)
PI 1580 FOR L=1 TO 25
PF 1590 SOUND 250+L*3,.01:SOUND
        215-L*7,.5:SOUND 200,.1
QM 1600 T=INT(50*RND(1)+20):FOR
        LO=1 TO T:NEXT LO
CC 1610 T=INT(5*RND(1)+4):COLOR
        T
QO 1620 NEXT
FM 1630 COLOR 0
NO 1640 IF STRIG(1)=0 THEN 1640
DC 1650 FOR LOOP=1 TO SN:PUT(SX(
        LOOP),SY(LOOP)),SHIP%:NE
        XT:PUT(X,Y),SIGHT%
HJ 1660 LINE(91,180)-(229,184),1
        ,BF
LD 1670 LINE(90,158)-(230,164),0
        ,BF
EE 1680 RETURN 140
JI 1690 END
NI 1700 DATA 42,15,0,20,0,0,20,0
        ,0,20,0,0,20,0,0,20,0,0,
        20,0,0,20,0,0,65,0,256,1
        6404,0,256,16404,0,5376,
        21569,0,21504,5441,0,163
        89,276,80,21,20,84,80,0,
        5,0,0,0,0
KC 1710 DATA 42,15,0,4096,0,0,20
        480,0,0,16384,0,0,16384,
        0,0,16385,0,0,16389,0,0,
        4240,0,0,8261,0,5376,-28
        582,0,21509,20649,0,1640
        5,5282,0,80,1414,0,0,272
        ,80,0,256,80,0,0,20,0,0,
        0,0
KB 1720 DATA 42,19,0,0,16385,0,0
        ,5,0,0,17,0,0,136,0,256,
        16,0,256,64,0,-23294,0,0
        ,8454,40,6400,-23984,128
        ,21765,-22174,64,16465,2
        2232,0,0,1578,64,0,1696,
        16,0,0,80,0,0,32,0,0,37,
        0,0,7,0,0,5,0,0,1,0,0
AB 1730 DATA 60,26,0,0,0,0,0,0,0,0
        ,0,0,0,-32640,0,0,0,9218
        ,0,0,0,8448,8192,8192,0,
        -23552,-32768,0,0,16386,
        0,0,0,9,0,544,0,24,0,34,
        -32502,96,0,2560,-22903,
        128,0
QJ 1740 DATA -28150,-30552,2,0,5
        716,-23932,8,0,25089,-21
        872,0,0,22786,-26112,0,0
        ,4736,6306,0,0,512,4608,
        0,0,512,2560,10368,8192,
        512,-32256,2048,0,0,2560
        ,0,0,0,512,128,0,128,0,1
        48,8,512,0,160,32,-32768
        ,0,24,0,0,0,32,0,0,2048,
        0,0
```

Program 2: The Last Warrior, 64 Version

Version by Kevin Mykytyn, Editorial Programmer

Please refer to the "MLX" article before entering this listing.

```
49152 :162,000,181,000,157,099,087
49158 :202,202,208,248,076,137,055
49164 :201,169,147,032,210,255,002
```



Enemy ships are approaching your scout vessel in the Commodore 64 version of "The Last Warrior."

```
49170 :169,000,141,170,002,141,129
49176 :171,002,141,168,002,141,137
49182 :169,002,141,172,002,133,137
49188 :191,160,023,169,000,153,220
49194 :000,212,136,016,248,169,055
49200 :047,141,024,212,169,242,115
49206 :141,023,212,169,240,141,212
49212 :013,212,169,128,141,018,229
49218 :212,169,255,141,015,212,046
49224 :169,026,141,005,212,169,026
49230 :003,141,001,212,032,082,037
49236 :194,032,065,193,173,098,071
49242 :202,208,009,032,098,194,065
49248 :032,161,194,032,010,196,209
49254 :032,206,196,169,001,141,079
49260 :098,202,032,122,194,169,157
49266 :000,133,039,032,237,196,239
49272 :032,065,193,169,000,174,241
49278 :170,002,172,171,002,032,163
49284 :192,200,169,080,166,187,102
49290 :164,188,032,192,200,032,178
49296 :162,195,032,168,195,032,160
49302 :190,192,032,131,199,032,158
49308 :080,200,165,197,201,064,039
49314 :240,215,238,172,002,165,170
49320 :197,201,064,208,250,165,229
49326 :197,201,064,240,250,165,011
49332 :197,201,064,208,250,206,026
49338 :172,002,240,189,173,000,194
49344 :220,074,176,010,174,070,148
49350 :003,224,046,240,003,206,152
49356 :070,003,074,176,010,174,199
49362 :070,003,224,155,240,003,137
49368 :238,070,003,074,176,031,040
49374 :174,080,003,208,007,174,100
49380 :060,003,224,019,240,019,025
49386 :072,173,060,003,056,233,063
49392 :001,141,060,003,173,080,186
49398 :003,233,000,141,080,003,194
49404 :104,074,176,020,174,080,112
49410 :003,240,007,174,060,003,233
49416 :224,070,240,008,238,060,080
49422 :003,208,003,238,080,003,037
49428 :074,176,004,162,001,134,059
49434 :034,096,169,000,133,012,214
49440 :162,006,160,012,189,080,129
49446 :003,074,038,012,189,060,158
49452 :003,153,000,208,189,070,155
49458 :003,153,001,208,136,136,175
49464 :202,016,233,165,012,141,057
49470 :016,208,096,160,041,185,000
49476 :045,198,153,000,008,185,145
49482 :086,198,153,137,008,185,073
49488 :128,198,153,198,008,185,182
49494 :169,198,153,000,009,136,239
49500 :016,229,169,016,141,089,240
49506 :008,169,056,141,092,008,060
49512 :160,050,185,030,200,153,114
49518 :064,009,136,016,247,169,239
49524 :032,141,248,007,169,127,072
49530 :141,021,208,169,100,141,134
49536 :060,003,141,070,003,169,062
49542 :000,141,080,003,133,034,013
49548 :162,007,189,200,193,157,024
49554 :039,208,169,000,157,130,081
49560 :003,202,016,242,169,000,016
49566 :141,027,208,133,013,162,074
```

```
49572 :005,032,230,197,202,208,014
49578 :250,169,037,141,254,007,004
49584 :169,136,133,187,169,019,221
49590 :133,188,169,185,141,015,245
49596 :208,169,228,141,014,208,132
49602 :169,036,141,255,007,096,130
49608 :010,003,004,005,006,007,235
49614 :001,072,138,072,152,072,201
49620 :169,000,133,006,133,009,150
49626 :152,072,041,007,133,004,115
49632 :104,074,074,074,133,002,173
49638 :138,072,041,252,010,038,013
49644 :009,133,003,104,041,003,017
49650 :133,008,169,003,056,229,072
49656 :008,168,165,016,192,000,029
49662 :240,005,010,010,136,208,095
49668 :251,133,008,165,002,162,213
49674 :006,010,038,006,202,208,224
49680 :250,133,005,165,006,024,087
49686 :101,002,133,006,165,005,178
49692 :101,003,133,005,165,006,185
49698 :101,009,133,006,165,005,197
49704 :101,004,133,005,144,003,174
49710 :230,006,024,105,000,133,032
49716 :005,165,006,105,032,133,242
49722 :006,160,000,177,005,166,060
49728 :039,240,005,005,008,076,181
49734 :074,194,069,008,145,005,053
49740 :104,168,104,170,104,096,054
49746 :169,059,141,017,208,169,077
49752 :216,141,022,208,169,029,105
49758 :141,024,208,096,169,000,220
49764 :133,005,169,008,133,006,042
49770 :162,056,160,000,152,145,013
49776 :005,136,208,251,230,006,180
49782 :202,208,246,096,169,232,247
49788 :133,005,169,000,133,002,054
49794 :169,003,133,006,169,216,058
49800 :133,003,162,004,160,000,086
49806 :169,199,145,005,169,014,075
49812 :145,002,136,208,245,230,090
49818 :006,230,003,202,208,238,017
49824 :096,162,018,160,000,169,253
49830 :000,133,002,169,032,133,123
49836 :003,173,027,212,201,020,040
49842 :176,007,173,027,212,041,046
49848 :003,208,002,169,000,145,199
49854 :002,136,208,235,230,003,236
49860 :202,208,230,096,133,014,055
49866 :072,138,072,152,072,134,074
49872 :010,134,018,132,011,132,133
49878 :020,169,000,133,017,133,174
49884 :019,165,021,133,015,165,226
49890 :014,197,010,176,016,169,040
49896 :001,141,090,003,165,010,130
49902 :056,229,014,141,110,003,023
49908 :076,004,195,169,000,141,061
49914 :090,003,165,014,056,229,039
49920 :010,141,110,003,165,015,188
49926 :197,011,176,042,169,001,090
49932 :141,100,003,165,011,056,232
49938 :229,015,141,120,003,076,090
49944 :052,195,169,000,141,100,169
49950 :003,165,015,056,229,011,253
49956 :141,120,003,173,110,003,074
49962 :205,120,003,176,003,173,210
49968 :120,003,133,037,165,018,012
49974 :170,165,020,164,039,208,052
49980 :012,197,038,208,004,228,235
49986 :193,240,008,133,038,134,044
49992 :193,168,032,207,193,173,014
49998 :090,003,208,017,165,017,066
50004 :024,109,110,003,133,017,224
50010 :165,018,105,000,133,018,017
50016 :076,113,195,165,017,056,206
50022 :237,110,003,133,017,165,255
50028 :018,233,000,133,018,173,171
50034 :100,003,208,017,165,019,114
50040 :024,109,120,003,133,019,016
50046 :165,020,105,000,133,020,057
50052 :076,149,195,165,019,056,024
50058 :237,120,003,133,019,165,047
50064 :020,233,000,133,020,238,020
50070 :022,212,198,037,208,152,211
50076 :104,168,104,170,104,096,134
50082 :160,000,136,208,253,096,247
50088 :165,034,240,093,169,002,103
50094 :133,016,169,006,141,008,135
50100 :212,169,033,141,011,212,190
50106 :173,060,003,056,233,012,211
50112 :133,002,173,080,003,233,048
50118 :000,133,003,070,003,102,253
50124 :002,162,000,160,121,173,054
50130 :070,003,056,233,040,133,233
50136 :021,165,002,032,200,194,062
50142 :162,159,032,200,194,162,107
50148 :000,032,200,194,162,159,207
50154 :032,200,194,169,000,133,194
50160 :034,141,008,212,165,187,219
50166 :056,233,020,133,187,165,016
50172 :188,233,000,133,188,016,242
50178 :006,169,000,133,187,133,118
50184 :188,096,169,000,141,033,123
50190 :208,169,001,133,016,133,162
50196 :039,162,040,160,121,169,199
50202 :120,133,021,169,120,032,109
50208 :200,194,162,040,160,121,141
50214 :169,120,133,021,169,120,002
50220 :032,200,194,162,040,160,064
50226 :120,169,000,133,021,169,150
50232 :020,032,200,194,162,120,016
50238 :160,120,169,000,133,021,153
50244 :169,140,032,200,194,162,197
50250 :030,160,065,169,064,133,183
50256 :021,169,130,032,200,194,058
50262 :162,000,160,080,169,065,210
50268 :133,021,169,030,032,200,165
50274 :194,162,159,160,080,169,254
50280 :064,133,021,169,130,032,141
50286 :200,194,162,000,160,130,188
50292 :169,120,133,021,169,040,000
50298 :032,200,194,162,159,169,014
50304 :120,032,200,194,162,120,188
50310 :160,170,169,120,133,021,139
50316 :032,200,194,162,159,160,023
50322 :180,169,170,133,021,169,220
50328 :120,032,200,194,162,040,132
50334 :160,171,032,200,194,169,060
50340 :120,133,021,169,040,032,167
50346 :200,194,132,021,138,162,249
50352 :000,160,180,032,200,194,174
50358 :162,097,160,170,169,120,036
50364 :133,021,138,032,200,194,138
50370 :162,108,160,160,169,130,059
50376 :133,021,138,032,200,194,150
50382 :162,099,160,145,169,144,061
50388 :133,021,169,118,032,200,117
50394 :194,160,047,185,035,199,014
50400 :153,152,053,185,083,199,025
50406 :153,024,056,136,016,241,088
50412 :096,120,169,250,141,020,008
50418 :003,169,196,141,021,003,007
50424 :088,096,173,172,002,240,251
50430 :003,076,049,234,162,005,015
50436 :189,090,003,208,042,189,213
50442 :140,003,024,125,110,003,159
50448 :157,140,003,189,060,003,056
50454 :105,000,157,060,003,189,024
50460 :080,003,008,105,000,157,125
50466 :080,003,040,144,052,189,030
50472 :060,003,201,060,144,045,041
50478 :032,230,197,144,040,189,110
50484 :140,003,056,253,110,003,105
50490 :157,140,003,189,060,003,098
50496 :233,000,157,060,003,189,194
50502 :080,003,008,233,000,157,039
50508 :080,003,040,176,010,189,062
50514 :060,003,201,030,176,003,043
50520 :032,230,197,189,100,003,071
50526 :208,031,189,150,003,024,187
50532 :125,120,003,157,150,003,146
50538 :189,070,003,105,000,157,118
50544 :070,003,201,157,176,004,211
50550 :201,020,176,031,032,230,040
50556 :197,144,026,189,150,003,065
50562 :056,253,120,003,157,150,101
50568 :003,189,070,003,233,000,122
50574 :157,070,003,201,020,144,225
50580 :229,201,157,176,225,222,078
50586 :130,003,208,060,169,040,252
50592 :157,130,003,189,248,007,126
50598 :201,036,208,045,165,191,244
50604 :208,044,173,027,212,201,013
50610 :090,176,037,169,001,133,016
50616 :191,189,060,003,056,233,148
50622 :012,133,253,189,080,003,092
50628 :233,000,133,254,070,254,116
50634 :102,253,189,070,003,056,107
50640 :233,040,133,254,076,218,138
50646 :197,254,248,007,202,240,082
50652 :003,076,004,197,032,028,048
50658 :193,076,049,234,173,027,210
50664 :212,157,060,003,173,027,096
50670 :212,074,157,070,003,160,146
50676 :000,173,027,212,016,002,162
50682 :160,001,152,157,090,003,045
50688 :160,000,173,027,212,016,076
50694 :002,160,001,152,157,100,066
50700 :003,173,027,212,074,024,013
50706 :105,050,157,120,003,173,114
50712 :027,212,074,024,105,050,004
50718 :157,110,003,169,100,157,214
50724 :130,003,169,033,157,248,008
50730 :007,024,096,000,000,000,169
50736 :000,000,000,000,000,000,048
50742 :000,000,000,000,024,000,078
50748 :000,024,000,000,024,000,108
50754 :000,024,000,003,195,192,224
50760 :003,195,192,000,024,000,230
50766 :000,024,000,000,024,000,126
50772 :000,024,000,000,000,000,108
50778 :000,000,000,008,000,000,098
50784 :008,000,000,008,000,000,112
50790 :020,000,000,062,000,000,184
50796 :213,128,001,000,064,000,002
50802 :000,000,000,000,000,000,114
50808 :000,000,000,000,000,000,120
50814 :000,000,000,000,000,000,126
50820 :008,000,000,008,000,000,148
50826 :008,000,000,008,000,000,154
50832 :028,000,000,034,000,000,206
50838 :127,000,000,127,000,001,149
50844 :162,192,006,000,048,000,052
50850 :000,000,000,000,000,000,162
50856 :000,000,008,000,000,008,184
50862 :000,000,008,000,000,008,190
50868 :000,000,008,000,000,028,216
50874 :000,000,062,000,000,065,057
50880 :000,000,255,128,001,255,063
50886 :192,001,255,192,006,193,013
50892 :176,024,000,012,096,000,000
50898 :003,000,084,068,068,068,245
50904 :068,084,000,000,016,016,144
50910 :016,016,016,016,000,000,030
50916 :084,004,084,064,064,084,100
50922 :000,000,084,004,084,004,154
50928 :004,084,000,000,068,068,208
50934 :084,004,004,004,000,000,086
50940 :084,064,084,004,004,084,064
50946 :000,000,084,064,084,068,046
50952 :068,084,000,000,084,004,248
50958 :004,004,004,004,000,000,030
50964 :084,068,084,068,068,084,220
50970 :000,000,084,068,084,004,010
50976 :004,084,000,168,128,128,032
50982 :168,008,008,168,000,168,046
50988 :128,128,128,128,128,168,084
50994 :000,168,136,136,136,136,250
51000 :136,168,000,168,136,136,032
51006 :168,160,136,136,000,168,062
51012 :128,128,160,128,128,168,140
51018 :000,000,000,000,000,000,074
51024 :000,000,000,252,192,192,204
51030 :252,012,012,252,000,204,050
51036 :204,204,252,204,204,204,084
51042 :000,048,048,048,048,048,082
51048 :048,048,000,252,192,192,068
51054 :240,192,192,252,000,192,154
51060 :192,192,192,192,192,252,048
51066 :000,240,204,204,204,204,154
51072 :204,240,000,198,013,208,223
51078 :008,169,000,141,066,003,009
51084 :141,086,003,162,005,169,194
51090 :224,056,125,248,007,133,171
51096 :002,073,255,133,003,173,023
51102 :060,003,056,253,060,003,081
51108 :197,003,176,004,197,002,231
51114 :176,017,173,070,003,056,153
51120 :253,070,003,197,003,176,110
51126 :015,197,002,176,002,144,206
51132 :009,202,208,221,169,127,100
51138 :141,021,208,096,169,255,060
51144 :141,021,208,165,034,240,241
51150 :246,189,060,003,141,066,143
51156 :003,189,070,003,141,076,182
51162 :003,169,020,133,013,169,213
51168 :128,141,004,212,169,129,239
51174 :141,004,212,189,080,003,091
```

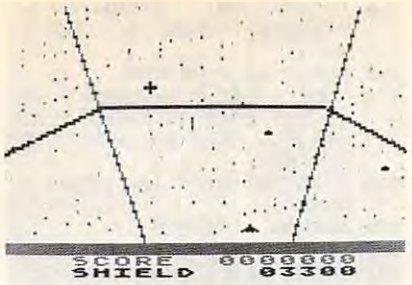
```
51180 :141,086,003,173,170,002,043
51186 :024,105,100,141,170,002,016
51192 :173,171,002,105,000,141,072
51198 :171,002,173,169,002,201,204
51204 :004,240,018,238,168,002,162
51210 :173,168,002,201,019,208,013
51216 :008,169,000,141,168,002,248
51222 :238,169,002,032,230,197,122
51228 :144,167,000,146,000,002,231
51234 :004,000,009,033,032,000,112
51240 :142,000,018,056,072,005,077
51246 :065,000,065,010,212,022,164
51252 :121,192,002,042,068,084,049
51258 :148,168,001,162,066,020,111
51264 :208,144,006,100,004,001,015
51270 :136,192,002,036,008,000,188
51276 :128,128,000,040,165,191,216
51282 :240,107,166,253,224,159,207
51288 :176,092,164,254,192,040,238
51294 :144,086,192,150,176,082,156
51300 :169,003,133,016,152,056,117
51306 :233,030,133,021,165,253,173
51312 :109,027,212,201,140,176,209
51318 :247,201,020,144,243,169,118
51324 :129,141,011,212,141,008,254
51330 :212,032,200,194,072,169,241
51336 :010,141,033,208,104,032,152
51342 :200,194,032,162,195,169,070
51348 :000,141,033,208,165,187,114
51354 :005,188,208,003,076,063,185
51360 :201,165,187,056,233,100,078
51366 :133,187,165,188,233,000,048
51372 :133,188,016,006,169,000,172
51378 :133,187,133,188,169,000,220
51384 :133,191,169,016,141,011,077
51390 :212,096,133,002,138,072,075
51396 :169,000,133,003,162,003,154
51402 :006,002,038,003,202,208,149
51408 :249,165,002,024,105,208,193
51414 :141,046,201,165,003,105,107
51420 :053,141,047,201,104,133,131
51426 :247,132,248,169,000,133,131
51432 :249,133,250,133,251,162,130
51438 :015,006,247,038,248,120,144
51444 :248,165,249,101,249,133,109
51450 :249,165,250,101,250,133,118
51456 :250,165,251,101,251,133,127
51462 :251,216,088,202,016,227,238
51468 :162,002,181,249,072,074,240
51474 :074,074,074,032,034,201,251
51480 :104,041,015,032,034,201,195
51486 :202,016,237,096,134,002,205
51492 :010,010,010,168,162,007,147
51498 :185,211,198,141,255,255,007
51504 :238,046,201,208,003,238,214
51510 :047,201,200,202,016,238,190
51516 :166,002,096,162,255,154,127
51522 :169,128,141,004,212,141,093
51528 :011,212,169,129,141,004,226
51534 :212,162,000,142,066,003,151
51540 :142,076,003,189,099,202,027
51546 :149,000,202,208,248,162,035
51552 :150,173,027,212,041,007,194
51558 :133,002,173,017,208,041,164
51564 :248,005,002,141,017,208,217
51570 :141,022,208,141,033,208,099
51576 :165,162,197,162,240,252,018
51582 :142,001,212,202,208,221,088
51588 :169,016,141,004,212,032,194
51594 :129,255,032,138,255,169,092
51600 :027,141,017,208,169,200,138
51606 :141,022,208,169,000,141,063
51612 :033,208,173,098,202,240,086
51618 :054,162,008,160,014,024,072
51624 :032,240,255,169,042,160,042
51630 :202,032,030,171,173,171,185
51636 :002,174,170,002,032,205,253
51642 :189,162,013,160,013,024,235
51648 :032,240,255,169,092,160,116
51654 :202,032,030,171,173,169,207
51660 :002,010,170,189,249,201,001
51666 :168,189,248,201,032,030,054
51672 :171,162,015,173,098,202,013
51678 :208,002,162,011,160,011,008
51684 :024,032,240,255,169,074,254
51690 :160,202,032,030,171,173,234
51696 :000,220,074,176,250,076,012
51702 :013,192,002,202,010,202,099
51708 :018,202,026,202,034,202,168
51714 :067,065,080,084,065,073,180
51720 :078,000,032,032,077,065,036
51726 :074,079,082,000,067,079,139
51732 :076,079,078,069,076,000,142
51738 :071,069,078,069,082,065,204
51744 :076,000,087,065,082,082,168
51750 :073,079,082,000,129,071,216
51756 :065,077,069,032,079,086,196
51762 :069,082,017,017,017,157,153
51768 :157,157,157,157,157,157,230
51774 :157,157,157,158,032,083,038
51780 :067,079,082,069,032,000,141
51786 :154,080,082,069,083,083,113
51792 :032,085,080,032,084,079,216
51798 :032,080,076,065,089,000,172
51804 :082,065,078,075,032,000,168
51810 :000,013,013,013,013,013,163
```

Program 3: The Last Warrior, Atari Version

Version by Kevin Mykytyn, Editorial Programmer

Please refer to the "MLX" article before entering this listing.

```
8192:173,148,002,024,105,040,236
8198:133,067,173,149,002,105,123
8204:000,133,068,032,017,033,039
8210:032,083,036,032,139,033,117
8216:076,002,039,032,083,033,033
8222:032,144,035,032,210,032,003
8228:162,000,169,100,141,000,096
8234:006,141,005,006,169,000,113
8240:141,016,006,206,022,006,189
8246:208,012,173,023,006,141,105
8252:022,006,032,216,033,032,145
8258:223,037,032,227,035,032,140
8264:123,037,169,000,133,077,099
8270:173,252,002,048,224,169,178
8276:255,141,252,002,238,250,198
8282:006,173,252,002,048,251,054
8288:169,255,141,252,002,206,097
8294:250,006,240,201,138,072,241
8300:024,105,044,141,134,032,076
8306:141,191,032,189,200,006,105
8312:141,133,032,169,000,133,216
8318:204,141,190,032,160,016,101
8324:153,255,255,136,208,250,109
8330:189,016,006,133,203,168,085
8336:185,201,032,157,008,208,167
8342:160,004,006,203,038,204,253
8348:136,208,249,165,203,024,117
8354:105,000,133,203,165,204,204
8360:105,040,133,204,189,000,071
8366:006,157,000,208,189,005,227
8372:006,157,200,006,170,160,111
8378:000,177,203,157,255,255,209
8384:232,200,192,016,208,245,005
8390:104,170,096,000,000,000,056
8396:000,000,000,000,000,000,204
8402:169,062,141,047,002,169,032
8408:004,141,111,002,162,003,127
8414:189,013,033,157,192,002,040
8420:202,016,247,169,003,141,238
8426:029,208,169,040,141,007,060
8432:212,133,204,169,000,133,067
8438:203,162,008,160,000,145,156
8444:203,136,208,251,202,208,180
8450:248,185,044,039,153,000,159
8456:040,136,208,247,096,127,094
8462:127,127,127,169,064,133,249
8468:160,162,096,169,012,157,008
8474:066,003,032,086,228,162,091
8480:096,169,003,157,066,003,014
8486:169,099,157,068,003,169,191
8492:033,157,069,003,169,007,226
8498:157,075,003,169,028,157,127
8504:074,003,032,086,228,173,140
8510:048,002,133,014,173,049,225
8516:002,133,015,160,088,169,123
8522:006,145,014,200,192,091,210
8528:208,249,096,160,037,185,247
8534:101,033,145,067,136,016,072
8540:248,169,007,141,198,002,089
8546:096,083,058,000,000,000,079
8552:179,163,175,178,165,000,196
8558:000,144,144,144,144,144,062
8564:144,144,000,000,000,000,148
8570:000,000,051,040,041,037,035
8576:044,036,000,000,000,016,224
8582:021,016,016,016,000,169,116
8588:001,141,029,006,141,043,245
8594:006,162,053,160,080,169,008
8600:000,141,047,006,169,023,026
8606:032,145,034,162,110,160,033
8612:080,169,000,141,047,006,095
8618:169,140,032,145,034,162,084
8624:159,160,050,169,035,141,122
8630:047,006,169,125,032,145,194
8636:034,162,125,160,035,169,105
8642:034,141,047,006,169,038,117
8648:032,145,034,162,000,160,221
8654:050,032,145,034,169,000,124
8660:141,043,006,096,173,120,023
8666:002,074,176,010,174,005,147
8672:006,224,035,240,003,206,170
8678:005,006,074,176,010,174,163
8684:005,006,224,175,240,003,121
8690:238,005,006,074,176,010,239
8696:174,000,006,224,053,240,177
8702:003,206,000,006,074,176,207
8708:010,174,000,006,224,195,101
8714:240,003,238,000,006,173,158
8720:132,002,208,005,162,001,014
8726:142,021,006,096,138,072,241
8732:173,029,006,162,006,010,158
8738:202,208,252,141,048,006,123
8744:162,005,032,128,034,133,022
8750:186,134,187,162,003,032,238
8756:128,034,024,101,186,133,146
8762:186,138,101,187,133,187,222
8768:104,074,144,006,078,048,006
8774:006,078,048,006,074,144,170
8780:012,078,048,006,078,048,090
8786:006,078,048,006,078,048,090
8792:006,024,101,186,144,002,039
8798:230,187,024,101,088,133,089
8804:186,165,187,101,089,133,193
8810:187,160,000,177,186,174,222
8816:043,006,208,006,077,048,244
8822:006,076,125,034,013,048,164
8828:006,145,186,096,169,000,214
8834:141,026,006,152,010,046,255
8840:026,006,202,208,249,174,233
8846:026,006,096,141,038,006,199
8852:072,138,072,152,072,142,028
8858:032,006,142,035,006,140,003
8864:030,006,140,037,006,169,036
8870:000,141,034,006,141,036,012
8876:006,141,046,006,173,047,079
8882:006,141,039,006,173,038,069
8888:006,205,032,006,176,018,115
8894:169,001,141,128,006,173,040
8900:032,006,056,237,038,006,059
8906:141,168,006,076,223,034,082
8912:169,000,141,128,006,173,057
8918:038,006,056,237,032,006,077
8924:141,168,006,173,030,006,232
8930:056,237,039,006,141,176,113
8936:006,173,035,006,170,173,027
8942:037,006,172,043,006,208,198
8948:016,205,044,006,208,005,216
8954:236,045,006,240,010,141,160
8960:044,006,142,045,006,168,155
8966:032,026,034,173,128,006,149
8972:208,021,173,034,006,024,222
8978:109,168,006,141,034,006,226
8984:173,035,006,105,000,141,228
8990:035,006,076,053,035,173,152
8996:034,006,056,237,168,006,031
9002:141,034,006,173,035,006,181
9008:233,000,141,035,006,173,124
9014:036,006,056,237,176,006,059
9020:141,036,006,173,037,006,203
9026:233,000,141,037,006,206,177
9032:046,006,208,157,104,168,249
9038:104,170,104,096,162,003,205
9044:032,106,032,202,016,250,210
9050:096,216,173,250,006,208,015
9056:009,032,082,035,032,115,145
9062:036,032,109,035,076,098,232
9068:228,238,246,006,173,246,221
9074:006,141,004,210,201,045,209
```



Notice how distant aliens appear smaller and nearby ships loom larger in the Atari version of "The Last Warrior."

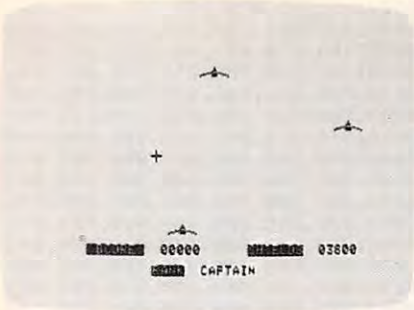
```
9080:208,005,169,020,141,246,141
9086:006,173,247,006,240,011,041
9092:206,247,006,074,074,074,045
9098:041,239,141,007,210,096,104
9104:160,091,162,035,169,007,000
9110:032,092,228,169,200,141,244
9116:023,006,169,000,141,021,004
9122:006,141,040,006,141,249,233
9128:006,141,250,006,162,003,224
9134:032,049,037,169,000,157,106
9140:208,006,141,030,208,202,207
9146:208,242,169,000,133,119,033
9152:169,040,133,120,162,008,056
9158:160,000,152,145,119,136,142
9164:208,251,230,120,202,208,143
9170:246,169,003,141,015,210,226
9176:169,004,141,008,210,169,149
9182:255,141,252,002,096,173,117
9188:021,006,240,044,169,025,221
9194:141,000,210,141,246,006,210
9200:169,100,141,001,210,206,043
9206:021,006,169,002,032,026,246
9212:036,169,002,032,055,036,070
9218:169,002,032,026,036,169,180
9224:002,032,055,036,169,002,048
9230:141,248,006,032,065,036,030
9236:169,000,141,001,210,096,125
9242:141,029,006,162,000,160,012
9248:079,173,000,006,056,233,067
9254:045,072,173,005,006,056,139
9260:233,028,074,141,047,006,061
9266:104,032,145,034,096,141,090
9272:029,006,162,159,160,079,139
9278:076,033,036,173,249,006,123
9284:208,012,160,035,162,003,136
9290:032,086,038,206,248,006,178
9296:208,239,096,165,088,133,241
9302:203,165,089,133,204,162,018
9308:012,160,000,173,010,210,145
9314:201,020,176,004,041,003,031
9320:145,203,200,208,242,230,052
9326:204,202,208,237,096,162,195
9332:003,189,208,006,240,003,253
9338:076,042,037,189,128,006,088
9344:208,027,189,149,006,024,219
9350:125,168,006,157,149,006,233
9356:189,000,006,105,000,157,085
9362:000,006,201,205,144,030,220
9368:032,049,037,144,025,189,116
9374:149,006,056,253,168,006,028
9380:157,149,006,189,000,006,159
9386:233,000,157,000,006,201,255
9392:015,176,003,032,049,037,232
9398:189,144,006,208,027,189,177
9404:154,006,024,125,176,006,167
9410:157,154,006,189,005,006,199
9416:105,000,157,005,006,201,162
9422:186,144,030,032,049,037,172
9428:144,025,189,154,006,056,018
9434:253,176,006,157,154,006,202
9440:189,005,006,233,000,157,046
9446:005,006,201,020,176,003,129
9452:032,049,037,222,160,006,230
9458:208,054,169,020,157,160,242
9464:006,189,016,006,201,006,160
9470:144,039,173,040,006,208,096
9476:037,173,010,210,201,080,203
9482:176,030,169,001,141,040,055
9488:006,189,000,006,056,233,250
9494:045,141,240,006,189,005,136
9500:006,056,233,028,074,141,054
9506:241,006,076,042,037,254,178
9512:016,006,202,240,003,076,071
9518:117,036,096,173,010,210,176
9524:157,000,006,173,010,210,096
9530:201,180,176,249,157,005,002
9536:006,160,000,173,010,210,111
9542:016,002,160,001,152,157,046
9548:128,006,160,000,173,010,041
9554:210,016,002,160,001,152,111
9560:157,144,006,173,010,210,020
9566:074,024,105,050,157,176,168
9572:006,173,010,210,074,024,085
9578:105,050,157,168,006,169,249
9584:100,157,160,006,169,001,193
9590:157,016,006,024,096,173,078
9596:040,006,240,083,174,240,139
9602:006,240,073,224,158,176,239
9608:069,172,241,006,192,021,069
9614:144,062,192,078,176,058,084
9620:152,056,233,020,141,047,029
9626:006,173,240,006,109,010,186
9632:210,201,185,176,246,201,099
9638:055,144,242,169,003,141,152
9644:029,006,032,145,034,072,234
9650:169,055,141,200,002,104,081
9656:032,145,034,032,212,037,164
9662:169,000,141,200,002,032,222
9668:074,038,173,249,006,240,208
9674:003,076,126,038,169,000,102
9680:141,040,006,096,162,007,148
9686:160,000,136,208,253,202,149
9692:208,250,096,162,003,189,104
9698:208,006,240,025,222,208,111
9704:006,208,030,032,049,037,082
9710:238,250,006,032,106,032,134
9716:169,000,141,250,006,141,183
9722:030,208,076,009,038,189,032
9728:012,208,041,001,240,003,249
9734:076,013,038,202,208,213,244
9740:096,169,000,141,030,208,144
9746:173,021,006,240,242,032,220
9752:048,038,169,007,157,016,203
9758:006,169,003,157,208,006,067
9764:169,120,141,247,006,169,120
9770:150,141,006,210,208,217,206
9776:160,014,138,072,162,004,086
9782:056,177,067,105,000,201,148
9788:154,144,002,169,144,145,050
9794:067,136,202,016,240,104,063
9800:170,096,160,034,162,003,185
9806:169,010,141,248,006,076,216
9812:065,036,024,177,067,233,174
9818:000,201,015,240,005,056,095
9824:145,067,176,005,169,025,171
9830:145,067,024,136,202,016,180
9836:234,160,035,177,067,201,214
9842:016,208,008,136,192,031,193
9848:208,245,238,249,006,096,138
9854:162,003,169,002,157,016,123
9860:006,202,208,250,169,255,198
9866:141,247,006,169,150,141,224
9872:006,210,173,010,210,141,126
9878:200,002,041,007,170,189,247
9884:001,040,160,000,145,014,004
9890:173,247,006,208,235,169,176
9896:112,145,014,169,000,141,237
9902:200,002,160,022,185,150,125
9908:039,145,067,200,192,038,093
9914:208,246,032,030,039,160,133
9920:024,185,164,039,145,067,048
9926:200,192,029,208,246,160,209
9932:011,177,067,201,144,240,020
9938:004,169,008,208,012,200,043
9944:192,013,208,241,177,067,090
9950:056,233,144,074,010,170,141
9956:189,209,039,141,245,038,065
9962:189,210,039,141,246,038,073
9968:162,000,160,029,189,255,011
9974:255,240,006,145,067,200,135
9980:232,208,245,032,030,039,014
9986:160,022,185,171,039,145,212
9992:067,200,192,038,208,246,191
9998:173,132,002,208,251,173,185
10004:132,002,240,251,162,255,038
10010:154,076,027,032,169,255,227
10016:141,248,006,032,212,037,196
10022:206,248,006,208,248,096,026
10028:000,016,016,016,124,124,084
10034:016,016,016,000,000,000,098
10040:000,000,000,000,000,000,056
10046:000,000,000,000,000,000,062
10052:008,000,000,000,000,000,076
10058:000,000,000,000,000,000,074
10064:000,000,000,000,012,000,092
10070:000,000,000,000,000,000,086
10076:000,000,000,000,000,000,092
10082:000,008,028,000,000,000,134
10088:000,000,000,000,000,000,104
10094:000,000,000,000,000,008,118
10100:028,028,000,000,000,000,172
10106:000,000,000,000,000,000,122
10112:000,000,008,008,028,034,206
10118:000,000,000,000,000,000,134
10124:000,000,000,000,000,008,148
10130:008,028,062,085,000,000,073
10136:000,000,000,000,000,000,152
10142:000,040,068,048,134,065,001
10148:148,066,148,066,036,020,136
10154:000,000,000,000,000,039,209
10160:033,045,037,000,047,054,136
10166:037,050,000,000,000,000,013
10172:050,033,046,043,000,048,152
10178:050,037,051,051,000,038,165
10184:041,050,037,034,053,052,211
10190:052,047,046,219,039,227,068
10196:039,233,039,241,039,249,028
10202:039,035,033,048,052,033,202
10208:041,046,000,045,033,042,175
10214:047,050,000,035,047,044,197
10220:047,046,037,044,000,039,193
10226:037,046,037,050,033,044,233
10232:000,055,033,050,050,041,221
10238:047,050,000,000,016,032,143
10244:048,064,080,096,112,000,148
```

Program 4: The Last Warrior, Apple Version

Version by Tim Victor, Editorial Programmer

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
CF 100 D$ = CHR$ (4): DIM P$(8),
       PX(3),PY(3),PZ(3),VX(3),V
       Y(3),QX(3),QY(3),QZ(3),R$
       (4)
BF 110 GOSUB 1000
3E 120 SH = 5000:SC = 0
03 130 P$(0) = "/012":P$(1) = "(
       )$+":P$(2) = "!" + CHR$ (
       34) + "##":P$(3) = "34":P
       $(4) = "%&":P$(5) = ",-"
77 140 P$(6) = "5":P$(7) = ".":P
       $(8) = "'"
58 150 GOSUB 970
C4 160 FOR I = 0 TO 3:PZ(I) = 10
       00:QZ(I) = 1000: NEXT
7D 170 CO = .95:SH = 5000:SC = 0
       : GOSUB 910: GOSUB 930
C2 180 XP = 52:YP = 59: XDRAW 1
       AT XP,YP
8F 190 RF = 0: FOR M = 0 TO 3
71 200 IF SH = 0 THEN 330
9C 210 I = FRE (0): GOSUB 400: O
       N I GOSUB 430,440,450,460
       ,470,560
5F 220 IF PZ(M) = 1000 THEN GOSU
       B 570: GOTO 310
FC 230 IF PZ(M) < 0 THEN RF = 1:
        GOTO 300
48 240 GOSUB 610
F3 250 IF PZ(M) > 15 THEN 300
7A 260 IF RND (1) < CO * .8 THEN
       300
```



The Apple version of "The Last Warrior" animates the alien ships using custom characters designed with the previously published "Apple SuperFont" utility.

```
01 270 XDRAW 1 AT XP,YP:XT = INT
       (PX(M)) * 7 - 7:YT = INT
       (PY(M)) * 8 - 4
0E 280 HCOLOR= 5: GOSUB 370: HCO
       LOR= 0: GOSUB 370: GOSUB
       650: XDRAW 1 AT XP,YP
EB 290 SH = SH - 100: GOSUB 930
F3 300 CO = CO * .9999: NEXT
92 310 IF RF < > 0 THEN XDRAW 1
       AT XP,YP: GOSUB 650: XDRA
       W 1 AT XP,YP
99 320 GOTO 190
64 330 XDRAW 1 AT XP,YP: VTAB 21
       : HTAB 2: PRINT "ANOTHER
       GAME? (Y OR N)"
46 340 GET A$: IF A$ = "N" OR A$
       = "n" THEN TEXT : END
5A 350 IF A$ = "Y" OR A$ = "y" T
       HEN 150
9D 360 GOTO 340
9A 370 HPLOT XT,YT TO 0,0: HPLOT
       XT,YT TO 0,159
85 380 HPLOT XT,YT TO 279,0: HPL
       OT XT,YT TO 279,159
25 390 RETURN
5E 400 I = 0:A = PEEK (49152)
84 410 IF A > 127 THEN POKE 4916
       8,0:A$ = CHR$ (A - 128):
       FOR I = 1 TO 6: IF A$ < >
       MID$ ("JILK P",I,1) THEN
       NEXT
18 420 RETURN
6E 430 GOSUB 890:XP = XP - (XP >
       6) * 7: GOTO 900
4B 440 GOSUB 890:YP = YP - (YP >
       7) * 8: GOTO 900
74 450 GOSUB 890:XP = XP + (XP <
       273) * 7: GOTO 900
D4 460 GOSUB 890:YP = YP + (YP <
       152) * 8: GOTO 900
CA 470 HCOLOR= 7: XDRAW 1 AT XP,
       YP: HPLOT 0,159 TO XP,YP:
       HPLOT 279,159 TO XP,YP
E1 480 HCOLOR= 0: HPLOT 0,159 TO
       XP,YP: HPLOT 279,159 TO
       XP,YP
57 490 XC = INT (XP / 7) + 1:YC
       = INT (YP / 8) + 1: FOR J
       = 0 TO 3: IF PZ(J) = 100
       0 THEN 540
18 500 IF YC < > INT (PY(J)) THE
       N 540
E8 510 IF XC < INT (PX(J)) - (PZ
       (J) < = 15) - (PZ(J) < =
       30) THEN 540
61 520 IF XC > INT (PX(J)) + (PZ
       (J) < = 15) THEN 540
4D 530 PZ(J) = - PZ(J): GOSUB 65
       0:SC = SC + 100: GOSUB 91
       0: GOTO 550
05 540 NEXT
91 550 XDRAW 1 AT XP,YP:SH = SH
       - 20: GOSUB 930: RETURN
67 560 GET A$: RETURN
4A 570 IF RND (1) < CO THEN 600
BB 580 PX(M) = RND (1) * 35 + 3:
       PY(M) = RND (1) * 20 + 1:
       PZ(M) = 45
F7 590 R = RND (1) - .5:VX(M) =
       ( ABS (R) - .25):VY(M) =
       SQR (.0625 - VX(M) * VX(M
       )) * SGN (R):RF = 1
16 600 RETURN
85 610 PX(M) = PX(M) + VX(M) * (
       PX(M) > 4 AND PX(M) < 37)
       : IF INT (QX(M)) < > INT
       (PX(M)) THEN RF = 1
5D 620 PY(M) = PY(M) + VY(M) * (
       PY(M) > 2 AND PY(M) < 20)
       : IF INT (QY(M)) < > INT
       (PY(M)) THEN RF = 1
77 630 PZ(M) = PZ(M) - 2 * (PZ(M
       ) > 2): IF PZ(M) = 30 OR
       PZ(M) = 15 THEN RF = 1
1E 640 RETURN
86 650 FOR I = 0 TO 3: IF QZ(I)
       = 1000 THEN 730
4A 660 NF = QZ(I):QZ(I) = ABS (Q
       Z(I))
0D 670 IF QZ(I) < = 15 THEN GOSU
       B 800: GOTO 700
0F 680 IF QZ(I) < = 30 THEN GOSU
       B 810: GOTO 700
5A 690 GOSUB 820
E8 700 IF NF > = 0 THEN 730
08 710 IF I < 3 THEN GOSUB 870:I
       = I - 1
67 720 QZ(3) = 1000
3D 730 NEXT : FOR I = 3 TO 0 STE
       P - 1: IF PZ(I) = 1000 TH
       EN 780
B0 740 QX(I) = PX(I):QY(I) = PY(
       I):QZ(I) = PZ(I)
EC 750 IF ABS (PZ(I)) < = 15 THE
       N GOSUB 830: GOTO 780
FE 760 IF ABS (PZ(I)) < = 30 THE
       N GOSUB 840: GOTO 780
5D 770 GOSUB 850
0F 780 NEXT
29 790 RETURN
19 800 HTAB QX(I) - 2: VTAB QY(I
       ): PRINT "     ": RETURN
FA 810 HTAB QX(I) - 1: VTAB QY(I
       ): PRINT "  ": RETURN
A6 820 HTAB QX(I): VTAB QY(I): P
       RINT " ": RETURN
42 830 GOSUB 860: HTAB PX(I) - 2
       : VTAB PY(I): PRINT P$(PH
       ): RETURN
98 840 GOSUB 860: HTAB PX(I) - 1
       : VTAB PY(I): PRINT P$(PH
       + 3): RETURN
21 850 GOSUB 860: HTAB PX(I): VT
       AB PY(I): PRINT P$(PH + 6
       ): RETURN
FD 860 PH = (PZ(I) > = 0) * INT
       (PX(I) - 2 * INT (PX(I) /
       2) + 1): RETURN
3B 870 FOR K = I TO 2:PX(K) = PX
       (K + 1):PY(K) = PY(K + 1)
       :PZ(K) = PZ(K + 1)
55 880 VX(K) = VX(K + 1):VY(K) =
       VY(K + 1):QX(K) = QX(K +
       1):QY(K) = QY(K + 1):QZ(
       K) = QZ(K + 1): NEXT :PZ(
       3) = 1000: RETURN
1B 890 OX = XP:OY = YP: RETURN
76 900 XDRAW 1 AT OX,OY: XDRAW 1
       AT XP,YP: RETURN
1A 910 N$ = STR$ (SC): VTAB 22:
       HTAB 11: GOSUB 950:R = IN
       T (SC / 2000): IF R > 4 T
       HEN R = 4
AC 920 VTAB 24: HTAB 16: CALL -
       868: PRINT R$(R);: RETURN
7E 930 IF SH < 0 THEN SH = 0
55 940 N$ = STR$ (SH): VTAB 22:
       HTAB 31: GOTO 950
3E 950 IF LEN (N$) < 5 THEN PRIN
       T LEFT$ ("0000",5 - LEN (
       N$));
48 960 PRINT N$;: RETURN
07 970 HOME : HGR : INVERSE : VT
       AB 22: HTAB 2: PRINT " SC
       ORE ";: HTAB 22: PRINT "S
       HIELDS";
50 980 VTAB 24: HTAB 10: PRINT "
       RANK";
F1 990 NORMAL : RETURN
A3 1000 POKE 232,100: POKE 233,3
20 1010 POKE 868,1: POKE 870,4:
       POKE 871,0
48 1020 FOR I = 0 TO 4: READ A:
       POKE 872 + I,A: NEXT
AF 1030 HCOLOR= 7: ROT= 0: SCALE
       = 4
42 1040 FOR I = 0 TO 4: READ R$(
       I): NEXT
3C 1050 FOR I = 768 TO I + 87: R
       EAD A: POKE I,A: NEXT
54 1060 FOR I = 138 * 256 TO I +
       175: READ A: POKE I,A:
       NEXT
A6 1070 IF PEEK (191 * 256) = 76
       THEN PRINT D$;"PR#A$300
       ": GOTO 1090
35 1080 POKE 54,0: POKE 55,3: CA
       LL 1002
14 1090 POKE 6,0: POKE 7,138: RE
       TURN
22 1100 DATA 176,12,31,5,0
6B 1110 DATA CAPTAIN,MAJOR,COLON
       EL,GENERAL,WARRIOR
D6 1120 DATA 216,120,133,69,134,
       70
2E 1130 DATA 132,71,166,7,10,10
44 1140 DATA 176,4,16,62,48,4
BB 1150 DATA 16,1,232,232,10,134
60 1160 DATA 27,24,101,6,133,26
A3 1170 DATA 144,2,230,27,165,40
95 1180 DATA 133,8,165,41,41,3
B1 1190 DATA 5,230,133,9,162,8
18 1200 DATA 160,0,177,26,36,50
63 1210 DATA 48,2,73,127,164,36
47 1220 DATA 145,8,230,26,208,2
9F 1230 DATA 230,27,165,9,24,105
0F 1240 DATA 4,133,9,202,208,226
B7 1250 DATA 165,69,166,70,164,7
       1
72 1260 DATA 88,76,240,253
6A 1270 DATA 0,0,0,0,0,0,0,0
E8 1280 DATA 0,0,0,0,0,0,40,42,2
5D 1290 DATA 64,64,96,16,21,117,
       112,0
19 1300 DATA 0,0,1,2,42,43,3,0
B7 1310 DATA 0,0,0,0,0,5,21,16
B4 1320 DATA 0,0,0,0,64,104,66,0
A9 1330 DATA 0,0,0,1,3,23,67,0
40 1340 DATA 0,0,0,0,8,42,0,0
74 1350 DATA 0,0,0,0,64,84,21,1
2A 1360 DATA 32,32,112,8,10,122,
       120,0
C3 1370 DATA 0,0,0,1,21,85,65,0
64 1380 DATA 0,0,0,0,0,2,10,8
12 1390 DATA 0,0,0,64,96,116,97,
       0
E7 1400 DATA 0,0,0,0,1,11,33,0
22 1410 DATA 0,0,0,0,4,21,0,0
45 1420 DATA 24,48,24,64,118,3,5
       6,0
9D 1430 DATA 56,99,48,55,88,111,
       102.0
A4 1440 DATA 6,12,63,27,113,31,1
       12,0
75 1450 DATA 3,6,48,108,12,51,0,
       0
BB 1460 DATA 0,56,99,48,55,88,0,
       0
7F 1470 DATA 0,6,12,63,27,113,0,
       0
5A 1480 DATA 0,0,76,118,54,0,0,0
```

Rescue On Fractalus! And Ballblazer

Tom R. Halfhill, Editor

Requirements: Atari 400/800, XL, or XE computer with at least 48K RAM, a disk drive, and a joystick (two joysticks are recommended for Ballblazer*). Versions for the Commodore 64 and Apple II-series computers were due to be released early this summer (except for the 64 version of* Ballblazer, *which is still under development).*

Delayed for a frustrating year by the turmoil of the home computer wars, *Rescue on Fractalus!* and *Ballblazer* have finally hit the market for Atari computers and are pending for the Commodore 64 and Apple as well. It's about time, too, because these action games have been anxiously awaited since their unveiling in mid-1984. Designed by Lucasfilm—the production company which brought us the *Star Wars* trilogy—both games were supposed to be marketed in cooperation with Atari. Unfortunately, Atari fell on hard times and the Lucasfilm games fell into limbo.

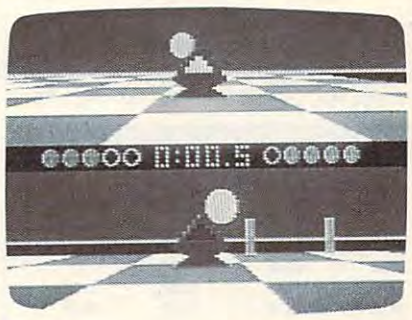
For a while, enthusiasts wondered if the games would ever see the glow of home video screens. Tantalizing preproduction copies of *Ballblazer* were known to be circulating in the pirate underground. Finally, Epyx, Inc. clinched a deal with Lucasfilm to market the programs. Now everyone can decide: Were they worth the wait?

A Mission Of Mercy

Rescue on Fractalus! integrates the best features of Brøderbund's *Choplifter*, Atari's *Star Raiders*, and Microprose's *Solo Flight*. Similar to *Choplifter*, your mission is to locate and rescue fellow pilots stranded in enemy territory—while fighting off hostile aircraft and ground targets. As in *Star Raiders*, you fly a spaceship from a first-person perspective—the video screen is a windshield onto the world beyond. And like *Solo Flight*, success depends on your ability to skillfully maneuver over an ever-changing landscape—while keeping an eye on your flight instruments at the bottom of the screen.

The scenario is that a number of space pilots have been shot down by alien Jaggies on the planet Fractalus. (The planetary landscape is generated by fractal mathematics—get it?) You're an old-fashioned air pilot who has been called back into the Ethercorps to rescue the downed space pilots. Launched

Rescue on Fractalus!: *As you look out onto the jagged mountains of Fractalus, a downed space pilot runs for the safety of your airlock.*



Ballblazer: *With only a half-second left to play and the score 4-3, player two (bottom window) tries to shove the Plasmorb past player one (top window) and into the goal.*

from an orbiting mother ship, you have to save a certain quota of pilots during each mission to advance to the next level. The task involves locating the pilots one by one, landing within walking distance, waiting for the pilot to enter your airlock, and then taking off again to resume the search. When your quota is filled, you return the pilots to the mother ship. Meanwhile, you have to duel with Jaggi gun emplacements dug into the mountainsides and fight off kamikaze attacks by Jaggi saucers.

Your craft, a modified Valkyrie-class fighter, is equipped with defense shields, Antimatter Bubble Torpedos, a targeting scope, a long-range scanner that picks up the presence of nearby space pilots, and a detector that warns when a Jaggi gun has locked onto your ship. Flight instruments include an artificial horizon, an energy-level meter, two altimeters, a compass, a speed indicator, a device that shows the clearance between your wingtips and the canyon walls, and digital readouts that tell how many Jaggies you've destroyed, how many pilots you have to rescue, and your distance from the pilot on the long-range scanner. All these dials and gauges are especially important on the highest levels, because you have to fly at night on instruments only.

A team of eight people created this game, and the attention to detail shows. In fact, the flight simulation could be a game in itself. You can climb, dive, and bank by steering the sensitive joystick, and keyboard controls let you speed up, slow down, land, switch your shields on and off, and open the airlock doors. Sound effects are rich: the whine of your engines, the explosions of torpedos and Jaggi gunshots, the anxious knock of pilots pounding on your airlock door to be rescued, and the hiss of the door as it opens and closes. Even the documentation is entertaining and professionally done.

Rescue on Fractalus!, like *Star Raiders*, calls for strategic thinking and contains some surprises and secrets for you to discover before you can move to the highest levels. It's definitely not a fast-paced twitch game. Indeed, at times it moves rather slowly as you search for the stranded pilots. But overall, it's an exceptional effort.

Split-Screen Soccer

Lucasfilm's other release, *Ballblazer*, is equally impressive. The split-screen, high-speed graphics of this frenetic game must be seen to be believed. Like *Rescue on Fractalus!*, it's a first-person perspective game that shows you the view from the driver's seat. But *Ball-*

blazer goes a step further and actually splits the screen into *two* views—one for each player. Two people can compete using two joysticks, or one person can play the computer.

Essentially, *Ballblazer* is space-age soccer played on a checkered field that measures 21 squares wide by 55 squares long (each square represents 5 × 5 meters). The Grid, as it's known, has a pair of goalposts at each end and is surrounded by force fields to keep players from straying out of bounds. As in soccer, the object is to score more goals than your opponent.

Unlike old-fashioned soccer, however, this game isn't played by teams of flesh-and-blood athletes trying to kick around a rubber ball. Instead, there are only two players, and each one drives a fast-moving hovercraft called a Rotofoil. The "ball" is a Plasmorb, a glowing object that floats two meters above the playing field. When you push the joystick forward to cruise over the Grid, your Rotofoil automatically points itself toward the Plasmorb. When you make contact, a force field grabs the Plasmorb and locks it in front of your Rotofoil. Then the Rotofoil reorients itself toward your goal, and away you go.

If you shove the Plasmorb between the goalposts, you get one point. By pressing the joystick button, you can also shoot the Plasmorb forward, recoiling your Rotofoil backward. By shooting the Plasmorb through the goal at close or intermediate range, you can score one or two points. You can even get three points by scoring a goal with an over-the-horizon shot (since the Grid is slightly curved, the goalposts are invisible at long range).

Meanwhile, of course, your computer or human opponent pursues in another Rotofoil, trying to block your shots and steal the Plasmorb. Whoever scores the most goals before the clock expires—usually three minutes—is the winner.

Like most sports, *Ballblazer* appears simple but actually contains many hidden strategies and possibilities. Championship play requires good defensive as well as offensive tactics. You can develop these skills by playing practice games against the computer (with adjustable difficulty levels) and by studying the amusing manual. *Ballblazer* looks like a three-point goal for Lucasfilm and Epyx.

Rescue on Fractalus!
Ballblazer
Epyx, Inc.
1043 Kiel Court
Sunnyvale, CA 94089
$40 each

Below The Root

Nick Piazza, Jr.

Requirements: Commodore 64 with a disk drive; Apple II-series computer with at least 48K RAM and a disk drive; IBM PC with at least 64K RAM, a disk drive, and color/graphics adapter; or an Enhanced Model IBM PCjr. A joystick is required for the 64 version and recommended for the Apple and IBM versions.

It didn't take long for Hollywood to realize that great books could often be made into great movies. The software industry appears to have made the same discovery, and Windham Classics has developed a superb adaptation of Zilpha Keatley Snyder's *Green Sky Trilogy*. (In fact, Snyder collaborated with programmer Dale Disharoon to create *Below the Root*.)

The *Green Sky Trilogy* is set in a fantasy world of trees and tunnels known as Green Sky, and it's up to a character on a quest to save this world from pending destruction. *Below the Root* casts the player as the quester in an

enchanting blend of an action and adventure game. It has been designed for players aged ten to adult, but my seven-year-old daughter was able to enjoy the game while playing with a grownup. It's even more enjoyable when several people join together to guide the quest. Indeed, one of the game's strong points is that it encourages cooperation rather than isolated play or deadly competition.

Colorful Graphics

One of the first things that impresses you about *Below the Root* is the quality of the screen graphics—the color and detail rival that of any arcade game. There are more than 100 different screens, each a delight to the eye.

Unlike text adventures, *Below the Root* doesn't require you to enter your commands by typing short sentences such as "Look North" or "Take Object." Instead, you select functions from various menus of choices (with the joystick, if you're using one). This makes the game more suitable for younger children. For example, the main menu lets you start a new game, save a current game on disk, continue a previously saved game, or view a sample game simply by indicating your choice. The last option, by the way, is particularly recommended for first-time players—it's wise to take a few minutes to orient yourself before plunging headlong into this unknown world.

After reading the well-written instructions and viewing the sample game, you're ready to start. First, the program asks which of five questers you wish to adopt. Each comes with varying degrees of stamina and "spirit skill." Questers also represent the two races which occupy Green Sky: the tree-loving Kindar and their cousins, the Erdling. Each race has its own attributes and limitations. All the questers, however, can grow in strength and spirit as they progress through the game.

What really sets this game apart is that questers can be either male or female. My daughter thought it was unfair that she was limited to choosing between three male characters and only two female characters, but still, at a time when computers are becoming increasingly important, it's gratifying to find a game that goes out of its way to encourage young girls as well as boys.

The level of each quester's spirit skill is an important factor in mastering the environment of Green Sky and successfully completing the quest. Spirit skills include the ability to read the emotions and thoughts of others (*pensing*), to heal yourself if injured, to influence tree growth (*grunspreke*), or to move yourself or other objects with your mind (*kiniport*). Each requires higher levels of spirit skill, and it's up to the player to determine how to raise this level. Those new to Green Sky should select questers with more spirit skill, while those who have played before may want to try questers with less spirit skill for a more challenging game.

Once you've selected your quester, the game begins in the quester's home. At this point, you have 50 days (in game time) to complete your quest and save Green Sky. Initial supplies are available in the quester's home, and players decide their course of action by making selections from the options menu. Many of these options are familiar to those who have played text adventures. You can examine, take, buy, eat, offer, drop, or sell various objects. You can also list an inventory of what you're carrying and call upon your spirit skills.

Quester, Heal Thyself

Questers are free to move throughout Green Sky in various ways: They can walk, run, jump, glide, climb, crawl, or enter and exit buildings. Since much of the action occurs in the treetops of Green Sky, you must be careful not to fall—unless you have a *shuba* for gliding, your quester will suffer a bump on the head. But watching the comical way in which questers rub their heads after a fall may help soothe the pain.

When you first encounter other characters in the game, an important spirit skill to use is pensing. This allows you to determine if they're friendly before speaking to them. This is vital, because some inhabitants are hostile. From time to time, it's also important to check your status, get adequate rest, eat when you're hungry, and heal yourself of any injuries. If your situation becomes too desperate, you may have to *renew* yourself. This option returns you home, but costs you a day from your quest.

The renew option, incidentally, spotlights another attractive feature of *Below the Root*: Questers are never killed or destroyed during their quest. While the world may be lost, violence rarely befalls the quester. This may be an important consideration for young players who would become upset if a character they created was destroyed during a game, or for parents who are disturbed by violence in computer games.

Below the Root
Windham Classics/Spinnaker Software
One Kendall Square
Cambridge, MA 02139
$26.95

Companion

Roger B. Crampton

Requirements: TI-99/4A with 32K RAM expansion card or box, Extended BASIC, a disk drive, and a printer.

Until I saw *Companion*, I considered replacing my TI-99/4A with a much more expensive computer for my serious word processing needs. I had tried several other word processors and found them either too slow, too cumbersome, or lacking essential features. But *Companion*, an inexpensive program written entirely in machine language, solves all of those problems.

Companion's editing features are superb—you have instantaneous full-screen editing capability. And the editing comes naturally, because all normal features of the TI keyboard retain their functions. For example, pressing Function 2 (Insert) works the same way with *Companion* as it does when you're entering a program in console or Extended BASIC. There are no surprises or tricky key sequences with *Companion*. Everything is logical and works in much the same manner as screen editing in BASIC. A delightful exception is the up- and down-arrow keys—they really move the cursor up and down, the way you wish they did in BASIC.

Of course, *Companion* has all of the usual word processing features. You can center headings, set tabs, automatically indent new paragraphs, search for text strings, and move or copy blocks of text. And you don't have to memorize a complex series of keystrokes to do simple things. For instance, pressing CTRL-P automatically generates a linefeed, a carriage return, and indents five spaces for the next paragraph.

The manual is well-written, succinct, and most important, understandable. At 142 pages, it may seem intimidating at first, but there is a good reason for its length. *Companion* has so many features that it takes that many pages to describe them.

Companion works flexibly with different kinds of printers. It lets you send control characters so you can switch to compressed or expanded fonts, or any other fonts allowed by your printer. A little judicious study of your printer manual, along with the *Companion* manual, should enable you to produce a brief list of control characters to adjust nearly any printer parameter.

Companion
Intelpro
5825 Baillargeon Street
Brossard, Quebec
Canada J42 1T1
$79.95

Jr-Draw For PCjr

Norm Cohen

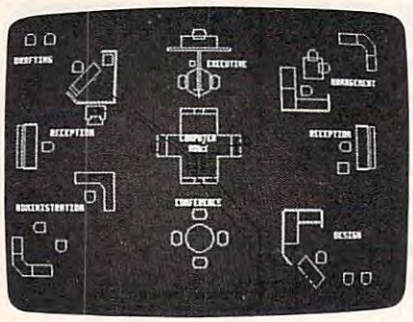
Requirements: Enhanced Model IBM PCjr. Light pen optional.

Jr-Draw is an interactive program which allows a PCjr user to create, save, modify, and print various types of graphics.

Using the keyboard or optional light pen, you can combine a virtually unlimited number of predefined and user-defined symbols, freehand objects, and text labels into a drawing. You can direct output to a graphics printer, and an optional driver is available for the HP 7470A and 7475A plotters. *Jr-Draw* seems most suited for technical drawings, layouts, or business-type graphics.

Assembling Symbols Into Drawings

You create drawings by typing two-keystroke combinations to select and modify primitive symbols, from which more complex shapes are assembled. For example, typing ALT-S followed by



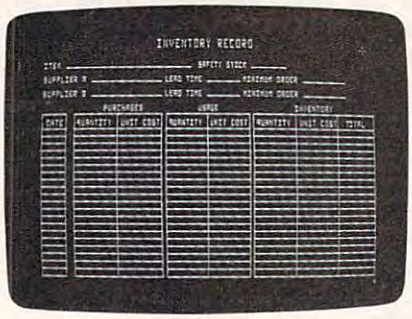
An office layout designed on a PCjr with Jr-Draw. *This sample screen is included with the software.*

10 places a circle (symbol number 10) in the drawing area of the screen. Once it's there, you can use the cursor control keys and function keys to move and change the size of the object. You can rotate objects in increments of 90 degrees—except for circles and ellipses. Another option is selective erasure.

Once created, adjacent objects can be selected together as if they were a single object, and all these manipulations can be performed on the group as a whole.

There are two ways to draw lines. The most flexible method is the freehand mode. You enter this mode by typing ALT-X, which converts the screen into something like an Etch-a-Sketch brand toy. As you move a crosshair around the screen with the cursor keys, a line is left in its wake.

I found myself using freehand mode almost exclusively. The second method requires you to press FN-4 at the beginning and end of each line segment to be plotted. Presumably this mode was intended for lines consisting of a single segment, but it's just as simple to use freehand mode for these as well.



This inventory record chart is one of the predefined templates included on the Jr-Draw *disks.*

By combining these lines with the primitive symbols, pictures are built piece by piece. You can save the pictures on disk at any point.

Transferring To Paper

Ultimately, though, the object is to get these graphics onto paper. *Jr-Draw* offers eight different formats in which the drawing can be produced on any of a dozen graphics printers. Variations include the orientation of the drawing on the page and whether the drawing is printed in condensed, emphasized, or full-width typestyles.

Since a drawing can consist of up to 99 pages or screenfuls of information, you can also specify a range of pages to be printed at one time.

If you want a higher resolution copy, you can buy an optional driver for the plotters mentioned above. Using a plotter should minimize the jagged appearance of diagonal lines which characterizes graphics printed in screen resolution.

Jr-Draw comes with several symbol templates. They contain flow-charting symbols, electrical schematic symbols, large and small block text, and a few symbols designated "interior" for floor plans.

But the key to *Jr-Draw*'s flexibility lies in the ability to define custom symbol templates for specific applications. For instance, a template of architectural symbols might be useful for creating an elevation drawing. Or a band director might find a template of musical instruments helpful for charting seating arrangements.

Custom templates are created in much the same way as drawings—they're composed of previously defined symbols and freehand lines. Once the new combination is "compressed" and placed into the template, it can be used in defining yet another new symbol. Like drawings, these templates may be stored on disk.

A Little Confusion

Jr-Draw is a complex piece of software; it's not something which can be used intuitively. Fortunately, an extensive interactive tutorial spares you from having to read the entire 174-page reference manual before you start. The tutorial covers the program's basic operations.

Unfortunately, not everything in the tutorial works correctly. Furthermore, the manual states that the tutorial is on disk 2 (of the three disks provided with the package), when it's actually on disk 3. But overall, the tutorial is a useful feature and can be covered completely in a little over two hours.

Once beyond the tutorial, you'll find that unless you use *Jr-Draw* regularly and frequently, the quick reference card will be a necessity. It is expecting a lot of a user, for example, to remember that small block text should be spaced six units apart while large text is spaced 32 units apart. If any program ever begged for a keyboard overlay, *Jr-Draw* is it. On the plus side, *Jr-Draw* wisely displays the meanings of the ten function keys along the bottom of the screen.

Jr-Draw never crashed during testing, but there were several instances—although minor and correctable—when results did not match what the manual indicates should happen. For example, changing the aspect of an ellipse so that it was flattened horizontally resulted in it springing to a vertical orientation. And the TAB and ENTER keys did not work as described when adding text to a drawing.

Inadvertent keystrokes can also cause problems. Typing the BACKSPACE key caused the template to disappear, for example. It took several moments scanning through the manual to learn that the way to restore it was to type CTRL-H.

Sometimes the corrective action itself is a source of aggravation. If you try to fill with color an object that is not completely enclosed, it "springs a leak" and the entire screen is filled. The only remedy is to delete the object, redraw the screen, and recreate the object.

Would A Mac Be Better?

User feedback is, in general, good. Typically, the object or objects selected for manipulation blink on and off to distinguish them from other objects in the drawing. As these objects become numerous or complex, however, the blinking slows down. Eventually, you reach the point where there is a significant lag between a keystroke and a screen update. In most instances, though, this is not a serious problem.

There were moments, brief but real, when I wondered if a Macintosh with *MacPaint* would be better for the job. The Macintosh mouse and pull-down menus make it very easy to manipulate. Presumably, *Jr-Draw* would be much easier to use with the optional light pen instead of the keyboard, but I lacked a light pen for testing.

Only one other annoyance was encountered: *Jr-Draw* requires you to frequently interchange the program and data disks when moving from one menu to another. *Jr-Draw* is a good candidate for conversion to cartridge, which would eliminate this drawback.

The disks are not copy-protected, but neither the manual nor the tutorial emphasizes the importance of backing up the disks before proceeding (this information is in Appendix B of the manual—read it *first*). The manual recommends everyday use of the original disk and setting aside the copies for backups, just the opposite of what most experts advise. Make sure your backups really work before following this practice.

Practical Applications

It is reasonable to use a computer to create drawings only when the computer offers some advantages over conventional methods. It may be that drawings can be created more quickly on a computer, or that once created, they are more easily modified. Or perhaps the quality of the drawings is improved, or the drawings can be produced more cost-effectively.

The answers to these issues depend partially on the specific software, but to a larger degree on the environment in which the software will be operated.

A site with no flat-art capability yet a need for casual graphics such as organizational charts may find *Jr-Draw* a useful tool. A one-page chart can be created in less than half an hour, and changes or updates are easily made.

But it should be understood that *Jr-Draw* produces graphics suitable for use in reports to other members of your department, perhaps, but not necessarily for sale to clients or for presentation to a board of directors.

There are many graphics programs on the market for the PC and PCjr. One of the worthy competitors to *Jr-Draw* is IBM's own *ColorPaint* program. PCjr owners should consider several different systems before selecting one to meet their needs.

Jr-Draw
Micrografx
1701 N. Greenville Avenue
Suite 703
Richardson, TX 75081
$195

HOTWARE: Software Best Sellers

This Month	Last Month	Title	Publisher	Remarks	Systems: Apple	Atari	Commodore	IBM	Macintosh
Entertainment									
1.	1.	*Flight Simulator II*	SubLogic	Aircraft simulation	●	●	●		
2.		*Karateka*	Brøderbund	Action karate game	●		●		
3.	4.	*F-15 Strike Eagle*	MicroProse	Air combat simulation	●	●	●	●	
4.	2.	*The Hitchhiker's Guide To The Galaxy*	Infocom	Comic adventure strategy game	●	●	●	●	●
5.	3.	*Flight Simulator*	Microsoft	Aircraft simulation				●	
Education									
1.	3.	*Math Blaster!*	Davidson	Introductory math program, ages 6-12	●	●	●	●	
2.	2.	*Typing Tutor III*	Simon & Schuster	Typing instruction program	●		●	●	●
3.	1.	*New Improved MasterType*	Scarborough	Typing instruction program	●	●	●	●	●
4.	4.	*Early Games*	Springboard	Educational games, ages 2-6	●	●	●	●	
5.		*Bank Street Music Writer*	Mindscape	Music composition program	●	●	●		
Home Management									
1.	1.	*Print Shop*	Brøderbund	Do-it-yourself print shop	●	●	●		
2.	2.	*Print Shop Graphics Library*	Brøderbund	100 additional graphics	●	●	●		
3.	4.	*HomePak*	Batteries Included	Word processing, telecommunications, & data management	●	●	●	●	
4.	5.	*Andrew Tobias's Managing Your Money*	Meca	Home financial program				●	
5.		*The Newsroom*	Springboard	Do-it-yourself newspaper	●		●	●	

Commodore 64 Memory Manager

Robert Lee

If you find yourself using several BASIC programs repeatedly, here's a way you can load them all into your computer at once, and run them independently. "Memory Manager" keeps track of up to eight programs in your Commodore 64 and lets you switch between them with the special function keys.

The Commodore 64 has 38K of Random Access Memory (RAM) available for BASIC programs. However, unless you're using a very large program, most of that memory is sitting empty, wasted.

"Memory Manager" is a utility which takes advantage of the leftover memory by using it to store other BASIC programs. It also uses 8K of additional RAM which is hidden beneath the Read Only Memory (ROM). Normally, this ROM prevents you from using the additional RAM, but Memory Manager collects every available byte of RAM (49.5K total) and partitions it into eight sections. You can load, list, run, and save up to eight BASIC programs in your computer with Memory Manager.

To use Memory Manager, type in and run the accompanying program. It asks you for the maximum amount of memory (in kilobytes) to be reserved for BASIC. The default response printed on the screen for you is 9K; simply press RETURN, or enter another value if you like. You can't change this value later without restarting the computer, so your response defines the maximum size of the BASIC program you can run. If you aren't sure how long your programs are, you can make a close estimate if you have a disk drive. Load a disk directory and note the number of blocks the program consumes on the disk. Since each block equals 256 bytes, four blocks equal one kilobyte. Simply divide the number of blocks by four to estimate the length. (For instance, a program that is 25 blocks long on the directory takes about 6.25K of RAM.) However, keep in mind that some programs require additional RAM when they run.

After you enter your answer, the cursor reappears and Memory Manager is ready to run. Activate it by typing SYS 53128 and pressing RETURN.

Eight Partitions

Depending on the amount of memory space available, up to eight programs can be handled by Memory Manager. The partitions are accessed by pressing one of the four special function keys. Press f1 to access partition 1, f2 for partition 2, and so on. When you flip to a different partition, Memory Manager displays the partition number on the screen.

For example, try typing or loading a program into the computer. This is partition 1. Type LIST to confirm that it's in memory. Now press one of the function keys—say, f5. When you type LIST again, nothing's there. To fill partition 5, just type or load another program. You can switch from partition to partition as often as you like. (If you press f5 when you're already in partition 5, nothing happens.)

Memory Manager uses only the space required to store a program, so none is wasted. If there is not enough room to store a certain program, Memory Manager delivers an error message.

If you wish to deactivate Memory Manager for some reason, type SYS 53144 and press RETURN. Pressing the RUN/ STOP–RESTORE combination also disables Memory Manager. You can turn it on again by entering SYS 53128. All the programs in memory will remain intact—although they may be damaged if you perform other tasks while Memory Manager is deactivated.

Remember that Memory Manager works only with BASIC programs; machine language programs are almost sure to cause memory conflicts. (The machine language portion of Memory Manager is stored above address 52736, $CE00 hex. It frees up RAM from $0800 to $CDFF minus the memory space assigned to BASIC.) Even with BASIC,

Commodore 64 Accessories

Sale ☆ ☆ **SUPER** ☆ ☆ SAVE

Summer Sizzler Sale

Sale Positively Ends 9-15-85

Prices will go back to regular sale prices

	List	Reg. Sale Price	Summer Sizzler Sale
C128 Commodore Computer Expandable to 512K, runs C-64, CPM, and 7.0 Programs. (Add $10 Shipping)	$349.00	$299.09	$289.95*
15½" Commodore 150-170 CPS Printer Near Letter Quality, Multiple Pin Tractor / Friction Feed. Best Printer Value in U.S.A. (Add $17.50 Shipping)	$895.00	$299.00	$199.95*
Commodore-64 IEEE Interface Allows you to run Pet Peripherals on the C-64, including the One Megabyte Disk Drive and 15½" Printer	$109.95	$69.00	$65.95
Juki Printer/Typewriter Letter Quality, daisy wheel, use as typewriter and/or printer (auto correction) (Add $10 Shipping)	$349.00	$249.00	$229.95*
SCM 80 CPS Printer Tractor/Friction 10" Famous Name Printer does Graphics w/Interface. (Add $10 Shipping)	$299.00	$159.00	$149.95*
Cardco G Plus Interface Converts Commodore to centronics for use with most printers, plus does Commodore graphics on graphic printers	$109.00	$59.00	$49.95
Alphacom 40 Column Printer Thermal technology - does graphics. (Add $7.50 Shipping) *Alphacom C-64 or Atari Interface $8.95.*	$99.00	$24.95	$22.95
190K Slim-Line Disk Drive Cooler, 20% faster, quieter than 1541 drive (Add $10 Shipping)	$249.00	$169.00	$139.95*
One Megabyte Disk Drive (1000K) Double sided drive hooks up to C-64 with IEEE interface, perfect as a second drive. (Add $10 Shipping)	$889.00	$199.00	$179.95*
13" Premium Quality Color Monitor Better than 1702, with separated video inputs. (Add $14.50 Shipping)	$399.00	$199.00	$169.95*
Voice Synthesizer Allows you to talk through your computer. Optional software lets you play talking adventure games (Zork, etc.)	$89.00	$49.00	$39.95
80 Column Board - 4 Slot Expander Allows you to program in 80 columns, plus gives you a 4 slot switch selectable expander	$149.00	$79.00	$59.95
Auto Dial Modem with Super Smart Software Upload, Download, Print, Catalog, This Package has it all!!! Best Modem Value in the U.S.A.!!	$129.95	$59.00	$39.95
Musicalc I Software The Best Musical Software for the C-64, allows you to change all parameters.	$59.95	$29.95	$14.95
Oil Barrons Software Better than Monopoly, comes with game board, disks and instruction manual. Strike Oil or Live in the Poor Farm	$49.95	$19.95	$9.95

*** plus Software Coupon Discounts**

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!

VISA — MASTER CARD — C.O.D.

No C.O.D. to Canada, APO-FPO.

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

keeping the programs from interfering with each other in every instance is practically impossible. BASIC programs with machine language subroutines, custom character sets, or POKEs into memory locations beyond the top of BASIC memory can mess up the programs stored in other partitions.

Variables set to certain values by a program in one partition will retain those values when you switch to another partition (although they'll be reset when you type RUN). For these reasons, we don't recommend using Memory Manager for critical applications such as software development. Instead, it's more suitable for keeping frequently used programs in memory rather than constantly accessing the cassette or disk drive, or for loading up a series of programs for a young person who cannot handle tapes or disks.

Commodore 64 Memory Manager

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
10 PRINT"{CLR}{6 DOWN}
   {11 RIGHT}MEMORY MANAGER"
                          :rem 62
20 PRINT"{3 DOWN}{11 RIGHT}
   {3 SPACES}FOR THE ":rem 109
30 PRINT"{3 DOWN}{12 RIGHT}COM
   MODORE 64{2 SPACES}"
                          :rem 210
100 FORX=52736TO53215 :rem 183
110 READA:CK=CK+A:POKEX,A
                          :rem 28
120 NEXT                  :rem 210
130 IF CK<>68936 THEN PRINT"
    {RVS}{2 DOWN} ERROR IN DAT
    A STATEMENTS":STOP :rem 50
140 INPUT"{5 DOWN}HOW MANY K F
    OR PROGRAM (6 TO 24)
    {2 RIGHT}9{3 LEFT}";M
                          :rem 141
145 IFM<6ORM>25THENPRINT"{CLR}
    NUMBER SHOULD BE FROM 6 TO
     24":GOTO140          :rem 168
150 POKE55,0:POKE56,M*4+8
                          :rem 153
160 FORX=53224TO53231:POKEX,M*
    4+8:POKEX+16,M*4+8:NEXT
                          :rem 181
170 FORX=0TO6:POKEX+53217,X*3+
    1:POKEX+53233,X*3+4:NEXT
                          :rem 237
180 POKE53214,X*3+1:POKE53215,
    M*4+8                 :rem 167
190 FORX=(M*4+8)*256+1TO(M*4+8
    )*256+24:POKEX,0:NEXT
                          :rem 136
200 PRINT"{CLR}{5 DOWN}
    {7 RIGHT}SYS 53128 TO ACTI
    VATE"                 :rem 12
210 PRINT"{3 DOWN}{7 RIGHT}SYS
     53144 TO DEACTIVATE"
                          :rem 223
220 PRINT"{3 DOWN}{9 RIGHT}PRO
    GRAM #1 IN USE"   :rem 141
230 PRINT"{4 DOWN}SYS 53128
    {3 UP}"               :rem 95
52736 DATA 169,255,141,180,207
      ,162                :rem 154
52742 DATA 19,189,181,207,32,2
      10                  :rem 49
52748 DATA 255,202,16,247,88,7
      6                   :rem 17
52754 DATA 49,234,162,255,165,
      157                 :rem 113
52760 DATA 240,247,165,203,201
      ,64                 :rem 91
52766 DATA 208,5,141,180,207,2
      40                  :rem 45
52772 DATA 236,172,180,207,192
      ,64                 :rem 106
52778 DATA 208,229,201,3,208,2
                          :rem 204
52784 DATA 162,6,201,4,208,2
                          :rem 98
52790 DATA 162,0,201,5,208,2
                          :rem 90
52796 DATA 162,2,201,6,208,2
                          :rem 99
52802 DATA 162,4,224,255,240,2
      01                  :rem 33
52808 DATA 173,141,2,240,1,232
                          :rem 190
52814 DATA 236,221,207,240,190
      ,120                :rem 134
52820 DATA 160,8,132,88,160,0
                          :rem 147
52826 DATA 132,87,173,222,207,
      133                 :rem 99
52832 DATA 89,173,223,207,133,
      90                  :rem 54
52838 DATA 134,91,162,3,165,90
                          :rem 211
52844 DATA 201,206,240,144,177
      ,87                 :rem 101
52850 DATA 145,89,230,87,208,2
                          :rem 213
52856 DATA 230,88,230,89,208,2
                          :rem 215
52862 DATA 230,90,201,0,208,22
      8                   :rem 244
52868 DATA 202,208,227,165,1,4
      1                   :rem 254
52874 DATA 254,133,1,166,91,18
      9                   :rem 12
52880 DATA 240,207,56,253,224,
      207                 :rem 98
52886 DATA 133,87,189,248,207,
      253                 :rem 124
52892 DATA 232,207,133,88,172,
      221                 :rem 102
52898 DATA 207,173,222,207,153
      ,224                :rem 154
52904 DATA 207,173,223,207,153
      ,232                :rem 142
52910 DATA 207,165,89,153,240,
      207                 :rem 101
52916 DATA 165,90,153,248,207,
      160                 :rem 105
52922 DATA 7,185,232,207,221,2
      48                  :rem 50
52928 DATA 207,144,44,208,8,18
      5                   :rem 10
52934 DATA 224,207,221,240,207
      ,144                :rem 139
52940 DATA 34,185,224,207,56,2
      29                  :rem 56
52946 DATA 87,153,224,207,185,
      232                 :rem 110
52952 DATA 207,229,88,153,232,
      207                 :rem 108
52958 DATA 185,240,207,56,229,
      87                  :rem 71
52964 DATA 153,240,207,185,248
      ,207                :rem 157
52970 DATA 229,88,153,248,207,
      136                 :rem 116
52976 DATA 16,201,189,224,207,
      133                 :rem 103
52982 DATA 94,189,232,207,133,
      95                  :rem 68
52988 DATA 169,0,133,87,169,8
                          :rem 180
52994 DATA 133,88,189,240,207,
      133                 :rem 114
53000 DATA 92,189,248,207,133,
      93                  :rem 53
53006 DATA 160,0,177,94,145,87
                          :rem 208
53012 DATA 230,87,208,2,230,88
                          :rem 198
53018 DATA 230,94,208,2,230,95
                          :rem 200
53024 DATA 165,95,197,93,208,2
      34                  :rem 62
53030 DATA 165,94,197,92,208,2
      28                  :rem 60
53036 DATA 189,224,207,133,87,
      189                 :rem 114
53042 DATA 232,207,133,88,177,
      94                  :rem 55
53048 DATA 145,87,230,87,208,2
                          :rem 211
53054 DATA 230,88,230,94,208,2
                          :rem 202
53060 DATA 230,95,165,95,197,9
      0                   :rem 11
53066 DATA 208,234,165,94,197,
      89                  :rem 72
53072 DATA 208,228,172,221,207
      ,185                :rem 147
53078 DATA 240,207,141,222,207
      ,185                :rem 143
53084 DATA 248,207,141,223,207
      ,142                :rem 142
53090 DATA 221,207,165,1,9,1
                          :rem 92
53096 DATA 133,1,173,221,207,2
      4                   :rem 244
53102 DATA 105,49,141,209,207,
      162                 :rem 87
53108 DATA 19,189,201,207,32,2
      10                  :rem 39
53114 DATA 255,202,16,247,169,
      255                 :rem 100
53120 DATA 141,180,207,88,76,4
      9                   :rem 3
53126 DATA 234,0,120,169,20,14
      1                   :rem 235
53132 DATA 20,3,169,206,141,21
                          :rem 186
53138 DATA 3,88,96,0,0,0
                          :rem 156
53144 DATA 120,169,49,141,20,3
                          :rem 194
53150 DATA 169,234,141,21,3,88
                          :rem 201
53156 DATA 96,0,0,0,0,255
                          :rem 197
53162 DATA 0,255,0,255,0,255
                          :rem 91
53168 DATA 0,255,0,255,64,141
                          :rem 149
53174 DATA 89,82,79,77,69,77
                          :rem 144
53180 DATA 32,72,71,85,79,78
                          :rem 121
53186 DATA 69,32,84,79,78,141
                          :rem 178
53192 DATA 147,141,69,83,85,32
                          :rem 216
53198 DATA 78,73,32,49,35,32
                          :rem 120
53204 DATA 77,65,82,71,79,82
                          :rem 121
53210 DATA 80,141,147,0,21,204
                          :rem 180
```

88K *Lowest Price In The USA!* 152K

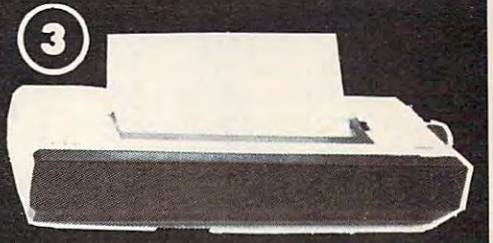
ATARI®

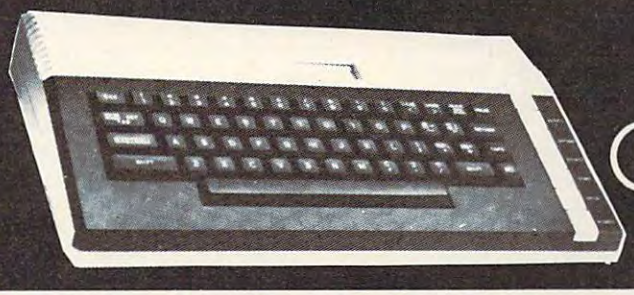
Computer System Sale

• Students • Word Processing • Home • Business

88K System $449
152K System $499*

EDUCATE WITH ATARI





① *Rated "Best Buy" by Consumers Digest Buyers Guide, January 1985*

LOOK AT ALL YOU GET FOR ONLY **$449** SYSTEM PRICE

	LIST PRICE	INDIVIDUAL SALE PRICE
Atari 800XL 88K Computer	$179.00	$109.00
Atari 1050 127K Disk Drive	299.00	199.00
Atari 1027 Letter Quality 20 CPS Printer	299.00	199.00
Atari Writer Word Processer	59.95	39.95
Atari BASIC Tutorial Manual	16.95	12.95
TOTALS	**$852.90**	**$559.90**

All connecting cables & T.V. interface included. Monitors sold separetly.

SAVE $100
All 5 ONLY
$449.00
SYSTEM SALE PRICE

152K SYSTEM $499.00

Other Accessories	List	Sale
12" Hi Resolution Green or Amber Screen Monitor	$199.00	**69.95**
13" Hi Resolution Color Monitor	$399.00	**169.95**
* ATARI 130XE 152K Computer	$249.00	**159.00**

Add $9.95 for Connection Cables (Monitors Only)
(Ltd. Qty)
Add $10 for UPS

15 DAY FREE TRIAL. We give you 15 days to try out this ATARI COMPUTER SYSTEM!! If it doesn't meet your expectations, just send it back to us prepaid and we will refund your purchase price!! **90 DAY IMMEDIATE REPLACEMENT WARRANTY**. If any of the ATARI COMPUTER SYSTEM equipment or programs fail due to faulty workmanship or material within 90 days of purchase we will replace it IMMEDIATELY with no service charge!!

Best Prices • Over 1000 Programs and 500 Accessories Available • Best Service • One Day Express Mail • Programming Knowledge • Technical Support

Add $25.00 for shipping and handling!!
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail! We accept Visa and MasterCard. We ship C.O.D. to continental U.S. addresses only. Add $10 more if C.O.D.

COMPUTER DIRECT

We Love Our Customers
22292 N. Pepper Rd., Barrington, Ill. 60010
312/382-5050 to order

COMPUTE!'s Guide To Typing In Programs

Before typing in any program, you should familiarize yourself with your computer. Learn how to use the keyboard to type in and correct BASIC programs. Read your manuals to understand how to save and load BASIC programs to and from your disk drive or cassette unit. Computers are precise—take special care to type the program *exactly* as listed, including any necessary punctuation and symbols, except for special characters as noted below. To help you with this task, we have implemented a special listing convention as well as a program to help check your typing—the "Automatic Proofreader." Please read the following notes before typing in any programs from COMPUTE!. They can save you a lot of time and trouble.

Commodore, Apple, and Atari programs can contain some hard-to-read (and hard-to-type) special characters, so we have developed a listing system that indicates the function of these control characters. (There are no special control characters in our IBM or TI-99/4A listings.) You will find Commodore and Atari special characters within curly braces; *do not type the braces.* For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. For Commodore, Apple, and Atari, a symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CTRL 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. Commodore computers also have a special control key labeled with the Commodore logo. Graphics characters entered with the Commodore logo key are enclosed in a special bracket that looks like this: [<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 S. One exception is {SHIFT-SPACE}. When you see this, hold down SHIFT and press the space bar. If a number precedes a symbol, such as {5 RIGHT}, {6 S}, or [<8 Q>], you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (printed in white on black) should be entered after pressing the inverse video key.

Since spacing is sometimes important, any more than two spaces will be listed. For example, {6 SPACES} means to press the space bar six times. Our listings never leave a space at the end of a line, instead moving it to the next printed line as {SPACE}. For your convenience, we have prepared this quick-reference chart for the Commodore and Atari special characters:

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:
{CLR}	SHIFT CLR/HOME	
{HOME}	CLR/HOME	
{UP}	SHIFT ↑ CRSR ↓	
{DOWN}	↑ CRSR ↓	
{LEFT}	SHIFT ← CRSR →	
{RIGHT}	← CRSR →	
{RVS}	CTRL 9	
{OFF}	CTRL 0	
{BLK}	CTRL 1	
{WHT}	CTRL 2	
{RED}	CTRL 3	
{CYN}	CTRL 4	
{PUR}	CTRL 5	
{GRN}	CTRL 6	
{BLU}	CTRL 7	
{YEL}	CTRL 8	

When You Read:	Press:	See:
[<1>]	COMMODORE 1	
[<2>]	COMMODORE 2	
[<3>]	COMMODORE 3	
[<4>]	COMMODORE 4	
[<5>]	COMMODORE 5	
[<6>]	COMMODORE 6	
[<7>]	COMMODORE 7	
[<8>]	COMMODORE 8	
{ F1 }	f1	
{ F2 }	SHIFT f1	
{ F3 }	f3	
{ F4 }	SHIFT f3	
{ F5 }	f5	
{ F6 }	SHIFT f5	
{ F7 }	f7	
{ F8 }	SHIFT f7	
←	←	

The Automatic Proofreader

We have developed a series of simple, yet effective programs that can help check your typing. Type in the appropriate Proofreader program listed below, then save it for future use. On the VIC, 64, or Atari, run the Proofreader to activate it, then enter NEW to erase the BASIC loader (the Proofreader remains active, hidden in memory, as a machine language program). Pressing RUN/STOP-RESTORE or SYSTEM RESET deactivates the Proofreader. You can use SYS 886 to reactivate the VIC/64 Proofreader, or PRINT USR(1536) to reenable the Atari Proofreader. On the Apple, the Proofreader automatically erases the BASIC portion of itself after you activate it by typing RUN, 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. 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.

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a decimal number (on the Commodore), a hexadecimal number (on the Apple), or a pair of letters (on the Atari or IBM) appears. The number or pair of letters is called a *checksum*. Try making a change in the line, and notice how the checksum changes.

All you need to do is compare the value provided by the Proofreader with the checksum printed in the program listing in the magazine. In Commodore listings, the checksum is a number from 0 to 255. It is set off from the rest of the line with *rem*. This prevents a syntax error if the checksum is typed in, but the REM statements and checksums need *not* be typed in. It is just there for your information.

In Atari, Apple, and IBM listings, the checksum is given to the left of each line number. Just type in the program one line at a time (without the printed checksum) and compare the checksum generated by the Proofreader to the checksum in the listing. If they match, go on to the next line. If not, check your typing: You've made a mistake. On the Commodore, 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 Commodore and Atari Proofreaders do 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. Because of the checksum method used, do not type abbreviations, such as ? for PRINT. The IBM Proofreader is the pickiest of all; it *will* detect errors in spacing and transposition. Be sure to leave Caps Lock on, except when typing lowercase characters.

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 type NEW, the Proofreader prompts you to press Y to be 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 in BASIC as usual (this replaces the Proofreader in memory). You can now run the program, but you may want to resave it to disk. The version of your program that you resave from BASIC will take up less space on disk and will load faster, but it can no longer be edited with the Proofreader. If you want to convert a program to Proofreader format, save it to disk with SAVE *"filename"*,A.

Special Proofreader Notes For Commodore Cassette Users

The Proofreader resides in a section of memory called the cassette buffer, which is used during tape LOADs and SAVEs. Therefore, be sure to press RUN/STOP-RESTORE to get the Proofreader out of the way before saving or loading a program. If you want to use the Proofreader with tape, run the Proofreader, then enter these two lines *exactly* as shown, pressing RETURN after each one:

```
A$="PROOFREADER.T":B$="{10
 SPACES}":FOR X=1 TO 4:A$=A$
 +B$:NEXT
FOR X=886 TO 1018:A$=A$+CHR$
(PEEK(X)):NEXT:OPEN 1,1,1,A$:
 CLOSE1
```

Then insert a blank tape and press RECORD and PLAY to save a special version of the Proofreader. Anytime you need to reload the Proofreader after it has been erased—for example, after you reload a paritally completed program—just rewind the tape, type OPEN1:CLOSE1, then press PLAY. You'll see the message FOUND PROOFREADER.T, but not the familiar LOADING message. Don't worry; the Proofreader is in memory. When READY comes back, enter SYS 886.

Program 1: VIC/64 Proofreader

By Charles Brannon, Program Editor

```
10 PRINT"{CLR}PLEASE WAIT...":
   FORI=886TO1018:READA:CK=CK+
   A:POKEI,A:NEXT
20 IF CK<>17539 THEN PRINT"
   {DOWN}YOU MADE AN ERROR":PR
   INT"IN DATA STATEMENTS.":EN
   D
30 SYS886:PRINT"{CLR}{2 DOWN}P
   ROOFREADER ACTIVATED.":NEW
40 DATA 173,036,003,201,150,20
   8,001,096,141,151,003,173
50 DATA 037,003,141,152,003,16
   9,150,141,036,003,169,003
60 DATA 141,037,003,169,000,13
   3,254,096,032,087,241,133
70 DATA 251,134,252,132,253,00
   8,201,013,240,017,201,032
80 DATA 240,005,024,101,254,13
   3,254,165,251,166,252,164
90 DATA 253,040,096,169,013,03
   2,210,255,165,214,141,251
100 DATA 003,206,251,003,169,0
    00,133,216,169,019,032,210
110 DATA 255,169,018,032,210,2
    55,169,58,032,210,255,166
120 DATA 254,169,000,133,254,1
    72,151,003,192,087,208,006
130 DATA 032,205,189,076,235,0
    03,032,205,221,169,032,032
140 DATA 210,255,032,210,255,1
    73,251,003,133,214,076,173
150 DATA 003
```

Program 2: Atari Proofreader

By Charles Brannon, Program Editor

```
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 3: IBM Proofreader

By Charles Brannon, Program Editor

```
10 'Automatic Proofreader Ver
   sion 2.00 (Lines 270,510,5
   15,517,620,630 changed fro
   m V1.0)
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)
190 LNUM=VAL(L$):TEXT$=MID$(L
    $,LEN(STR$(LNUM))+1)
200 IF ASC(L$)>57 THEN 260 'n
    o line number, therefore
    command
210 IF TEXT$="" 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$:LNUM(P)=VAL(L$):
    L$(P)=MID$(L$,LEN(STR$(VA
    L(L$)))+1):P=P+1:WEND
480 MAX=P:CLOSE #1:GOTO 130
490 IF COMMAND$="NEW" THEN IN
    PUT "Erase program - Are
    you sure";L$:IF LEFT$(L$,
    1)="y" OR LEFT$(L$,1)="Y"
     THEN MAX=0:GOTO 130:ELSE
     130
500 IF COMMAND$="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: POKE I
   ,A: NEXT
20 IF C < > 7258 THEN PRINT "ER
   ROR IN PROOFREADER DATA STAT
   EMENTS": END
30 IF PEEK (190 * 256) < > 76 T
   HEN POKE 56,0: POKE 57,3: CA
   LL 1002: GOTO 50
40 PRINT CHR$ (4);"IN#A$300"
50 POKE 34,0: HOME : POKE 34,1:
    VTAB 2: PRINT "PROOFREADER
   INSTALLED"
60 NEW
100 DATA 216,32,27,253,201,141
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 Atari

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost fail-safe entry of machine language programs published in COMPUTE!. *You need to know nothing about machine language to use MLX—it was designed for everyone.*

MLX is a new way to enter long machine language (ML) programs with a minimum of fuss. MLX lets you enter the numbers from a special list that looks similar to BASIC DATA statements. It checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers. It won't let you enter numbers greater than 255 (forbidden in ML). It won't let you enter the wrong numbers on the wrong line. In addition, MLX creates a ready-to-use tape or disk file.

Using MLX

Type in and save the appropriate version of MLX (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. Both versions of MLX asks you for two numbers: the starting address and the ending address. In addition, the Atari version asks for a run/init address. These numbers are given in the article accompanying the ML program presented in MLX format. The Atari version also gives you three options for saving the file: as a boot tape, as disk binary file, or as boot disk. The article with the ML program should suggest which format to use.

When you run MLX, you'll see a prompt corresponding to the starting address. The prompt is the current line you are entering from the listing. It increases by six each time you enter a line. That's because each line has seven numbers—six actual data numbers plus a *checksum number*. The checksum verifies that you typed the previous six numbers correctly. If you enter any of the six numbers wrong, or enter the checksum wrong, the computer rings a buzzer and prompts you to reenter the line. If you enter it correctly, a bell tone sounds and you continue to the next line.

MLX accepts only numbers as input. If you make a typing error, press the Commodore INST/DEL key or the Atari DEL/BACK SPACE; the entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as listed, the computer automatically prints the comma and goes on to accept the next number. If you enter less than three digits, you can press either the space bar or RETURN key to advance to the next number. The checksum automatically appears in inverse video for emphasis.

To simplify your typing, the Commodore 64 version of MLX redefines part of the keyboard as a numeric keypad (lines 581–584):

```
    U  I  O                7  8  9
H   J  K  L   become   0    4  5  6
    M  ,  .                1  2  3
```

MLX Commands

When you finish typing an ML listing (assuming you type it all in one session), you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you've made a typo when entering the MLX program itself.

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later. Each command is accessed by pressing one letter, plus the SHIFT key for 64 MLX or the CTRL key for the Atari version. MLX recognizes these commands:

Commodore	Atari	Command
SHIFT-S	CTRL-S	Save
SHIFT-L	CTRL-L	Load
SHIFT-N	CTRL-N	New Address
SHIFT-D	CTRL-D	Display

When you enter a command, MLX jumps out of the line you've been typing, so we recommend you do it at a new prompt. Use the Save command to save what you've been working on. It will save on tape or disk, as if you've finished, but the tape or disk won't work, of course, until you finish the typing. Remember to make a note of what address you stop at. The next time you run MLX, answer all the prompts as you did before—regardless of where you stopped typing—then insert the disk or tape. When you get to the entry prompt, press SHIFT-L (64) or CTRL-L (Atari) to reload the partly completed file into memory. Then use the New Address command to resume typing.

To use the New Address command, press SHIFT-N (64) or CTRL-N (Atari) and enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the MLX-format listing, or else the checksum won't work. The Display command lets you display a section of your typing. After you press SHIFT-D or CTRL-D, enter two addresses within the line number range of the listing. You can break out of the listing display and return to the prompt by pressing any key.

Atari MLX: Machine Language Entry

```
DA 100 GRAPHICS 0:DL=PEEK(560)+256*PEEK(561)+4:POKE DL-1,71:POKE DL+2,6
NJ 110 POSITION 8,0:? "MLX":POSITION 23,0:? "[rat] safe entry":POKE 710,0:?
JK 120 ? "Starting Address";:INPUT BEG:? "  Ending Address";:INPUT FIN:? "Run/Init Address";:INPUT STARTADR
DD 130 DIM A(6),BUFFER$(FIN-BEG+127),T$(20),F$(20),CIO$(7),SECTOR$(128),DSKINV$(6)
JJ 140 OPEN #1,4,0,"K:":? :? ,"[Tape] or [Disk]:";
BM 150 BUFFER$=CHR$(0):BUFFER$(FIN-BEG+30)=BUFFER$:BUFFER$(2)=BUFFER$:SECTOR$=BUFFER$
GC 160 ADDR=BEG:CIO$="hhh":CIO$(4)=CHR$(170):CIO$(5)="LV":CIO$(7)=CHR$(228)
EJ 170 GET #1,MEDIA:IF MEDIA<>84 AND MEDIA<>68 THEN 170
PO 180 ? CHR$(MEDIA):? :IF MEDIA<>ASC("T") THEN BUFFER$="":GOTO 250
PL 190 BEG=BEG-24:BUFFER$=CHR$(0):BUFFER$(2)=CHR$(INT((FIN-BEG+127)/128))
KF 200 H=INT(BEG/256):L=BEG-H*256:BUFFER$(3)=CHR$(L):BUFFER$(4)=CHR$(H)
EC 210 PINIT=BEG+8:H=INT(PINIT/256):L=PINIT-H*256:BUFFER$(5)=CHR$(L):BUFFER$(6)=CHR$(H)
PB 220 FOR I=7 TO 24:READ A:BUFFER$(I)=CHR$(A):NEXT I:DATA 24,96,169,60,141,2,211,169,0,133,10,169,0,133,11,76,0,0
DP 230 H=INT(STARTADR/256):L=STARTADR-H*256:BUFFER$(15)=CHR$(L):BUFFER$(19)=CHR$(H)
KL 240 BUFFER$(23)=CHR$(L):BUFFER$(24)=CHR$(H)
HI 250 IF MEDIA<>ASC("D") THEN 360
OO 260 ? :? "Boot [Disk] or Binary [File:";
LI 270 GET #1,DTYPE:IF DTYPE<>68 AND DTYPE<>70 THEN 270
GM 280 ? CHR$(DTYPE):IF DTYPE=70 THEN 360
PJ 290 BEG=BEG-30:BUFFER$=CHR$(0):BUFFER$(2)=CHR$(INT((FIN-BEG+127)/128))
KG 300 H=INT(BEG/256):L=BEG-H*256:BUFFER$(3)=CHR$(L):BUFFER$(4)=CHR$(H)
HH 310 PINIT=STARTADR:H=INT(PINIT/256):L=PINIT-H*256:BUFFER$(5)=CHR$(L):BUFFER$(6)=CHR$(H)
AO 320 RESTORE 330:FOR I=7 TO 30:READ A:BUFFER$(I)=CHR$(A):NEXT I
GA 330 DATA 169,0,141,231,2,133,14,169,0,141,232,2,133,15,169,0,133,10,169,0,133,11,24,96
OB 340 H=INT(BEG/256):L=BEG-H*256:BUFFER$(8)=CHR$(L):BUFFER$(15)=CHR$(H)
DO 350 H=INT(STARTADR/256):L=STARTADR-H*256:BUFFER$(22)=CHR$(L):BUFFER
```

```
      $(26)=CHR$(H)
JP 360 GRAPHICS 0:POKE 712,1
      0:POKE 710,10:POKE 70
      9,2
JK 370 ? ADDR;":";:FOR J=1 T
      O 6
NF 380 GOSUB 570:IF N=-1 THE
      N J=J-1:GOTO 380
BF 390 IF N=-19 THEN 720
OI 400 IF N=-12 THEN LET REA
      D=1:GOTO 720
AI 410 TRAP 410:IF N=-14 THE
      N ? :? "New Address";
      :INPUT ADDR:? :GOTO 3
      70
JD 420 TRAP 32767:IF N<>-4 T
      HEN 480
AJ 430 TRAP 430:? :? "Displa
      y:From";:INPUT F:? ,"
      To";:INPUT T:TRAP 327
      67
ML 440 IF F<BEG OR F>FIN OR
      T<BEG OR T>FIN OR T<F
       THEN ? CHR$(253);"At
       least ";BEG;", Not M
      ore Than ";FIN:GOTO 4
      30
MH 450 FOR I=F TO T STEP 6:?
      :? I;":";:FOR K=0 TO
       5:N=PEEK(ADR(BUFFER$
      )+I+K-BEG):T$="000":T
      $(4-LEN(STR$(N)))=STR
      $(N)
MA 460 IF PEEK(764)<255 THEN
       GET #1,A:POP :POP :?
      :GOTO 370
FM 470 ? T$;",";:NEXT K:? CH
      R$(126);:NEXT I:? :?
      :GOTO 370
GA 480 IF N<0 THEN ? :GOTO 3
      70
MH 490 A(J)=N:NEXT J
JM 500 CKSUM=ADDR-INT(ADDR/2
      56)*256:FOR I=1 TO 6:
      CKSUM=CKSUM+A(I):CKSU
      M=CKSUM-256*(CKSUM>25
      5):NEXT I
KK 510 RF=128:SOUND 0,200,12
      ,8:GOSUB 570:SOUND 0,
      0,0,0:RF=0:? CHR$(126
      )
CN 520 IF N<>CKSUM THEN ? :?
       "Incorrect";CHR$(253
      );:? :GOTO 370
EK 530 FOR W=15 TO 0 STEP -1
      :SOUND 0,50,10,W:NEXT
      W
FL 540 FOR I=1 TO 6:POKE ADR
      (BUFFER$)+ADDR-BEG+I-
      1,A(I):NEXT I
HB 550 ADDR=ADDR+6:IF ADDR<=
      FIN THEN 370
GM 560 GOTO 710
FI 570 N=0:Z=0
PH 580 GET #1,A:IF A=155 OR
      A=44 OR A=32 THEN 670
FB 590 IF A<32 THEN N=-A:RET
      URN
EB 600 IF A<>126 THEN 630
ML 610 GOSUB 690:IF I=1 AND
      T=44 THEN N=-1:? CHR$
      (126);:GOTO 690
GN 620 GOTO 570
GJ 630 IF A<48 OR A>57 THEN
      580
AN 640 ? CHR$(A+RF);:N=N*10+
      A-48
EB 650 IF N>255 THEN ? CHR$(
      253);:A=126:GOTO 600
EH 660 Z=Z+1:IF Z<3 THEN 580
JH 670 IF Z=0 THEN ? CHR$(25
      3);:GOTO 570
KC 680 ? ",";:RETURN
NO 690 POKE 752,1:FOR I=1 TO
       3:? CHR$(30);:GET #6
      ,T:IF T<>44 AND T<>58
       THEN ? CHR$(A);:NEXT
       I
PI 700 POKE 752,0:? " ";CHR$
      (126);:RETURN
KM 710 GRAPHICS 0:POKE 710,2
      6:POKE 712,26:POKE 70
      9,2
FF 720 IF MEDIA=ASC("T") THE
      N 890
OJ 730 REM  D:CSK
OK 740 IF READ THEN ? :? "Lo
      ad File":?
IG 750 IF DTYPE<>70 THEN 104
      0
AE 760 ? :? "Enter AUTORUN.S
      YS for automatic use"
      :? :? "Enter filename
      ":INPUT T$
GF 770 F$=T$:IF LEN(T$)>2 TH
      EN IF T$(1,2)<>"D:" T
      HEN F$="D:":F$(3)=T$
NJ 780 TRAP 870:CLOSE #2:OPE
      N #2,8-4*READ,0,F$:?
      :? "Working..."
JM 790 IF READ THEN FOR I=1
      TO 6:GET #2,A:NEXT I:
      GOTO 820
PO 800 PUT #2,255:PUT #2,255
DJ 810 H=INT(BEG/256):L=BEG-
      H*256:PUT #2,L:PUT #2
      ,H:H=INT(FIN/256):L=F
      IN-H*256:PUT #2,L:PUT
      #2,H
NF 820 GOSUB 970:IF PEEK(195
      )>1 THEN 870
IF 830 IF STARTADR=0 OR READ
       THEN 850
FD 840 PUT #2,224:PUT #2,2:P
      UT #2,225:PUT #2,2:H=
      INT(STARTADR/256):L=S
      TARTADR-H*256:PUT #2,
      L:PUT #2,H
HH 850 TRAP 32767:CLOSE #2:?
       "Finished.":IF READ
      THEN ? :? :LET READ=0
      :GOTO 360
HF 860 END
FO 870 ? "Error ";PEEK(195);
      " trying to access":?
       F$:CLOSE #2:? :GOTO
      760
MC 880 REM  BOOT TAPE
HN 890 IF READ THEN ? :? "Re
      ad Tape"
HI 900 ? :? :? "Insert, Rewi
      nd Tape.":? "Press PL
      AY ";:IF  NOT READ TH
      EN ? "& RECORD"
LP 910 ? :? "Press  RETURN  wh
      en ready:";
JH 920 TRAP 960:CLOSE #2:OPE
      N #2,8-4*READ,128,"C:
      ":? :? "Working..."
NH 930 GOSUB 970:IF PEEK(195
      )>1 THEN 960
HH 940 CLOSE #2:TRAP 32767:?
       "Finished.":? :? :IF
       READ THEN LET READ=0
      :GOTO 360
HF 950 END
CD 960 ? :? "Error ";PEEK(19
      5);" when reading/wri
      ting boot tape":? :CL
      OSE #2:GOTO 890
MB 970 REM  CIO Load/Save Fil
      e#2 opened  READ=0 fo
      r write, READ=1 for r
      ead
EA 980 X=32:REM File#2,$20
EF 990 ICCOM=834:ICBADR=836:
      ICBLEN=840:ICSTAT=835
MD 1000 H=INT(ADR(BUFFER$)/2
      56):L=ADR(BUFFER$)-H
      *256:POKE ICBADR+X,L
      :POKE ICBADR+X+1,H
FH 1010 L=FIN-BEG+1:H=INT(L/
      256):L=L-H*256:POKE
      ICBLEN+X,L:POKE ICBL
      EN+X+1,H
MD 1020 POKE ICCOM+X,11-4*RE
      AD:A=USR(ADR(CIO$),X
      )
BG 1030 POKE 195,PEEK(ICSTAT
      ):RETURN
KA 1040 REM  SECTOR I/O
GC 1050 IF READ THEN 1100
HE 1060 ? :? "Format Disk In
       Drive 1? (Y/N):";
FC 1070 GET #1,A:IF A<>78 AN
      D A<>89 THEN 1070
EC 1080 ? CHR$(A):IF A=78 TH
      EN 1100
CP 1090 ? :? "Formatting..."
      :XIO 254,#2,0,0,"D:"
      :? "Format Complete"
      :?
AC 1100 NR=INT((FIN-BEG+127)
      /128):BUFFER$(FIN-BE
      G+2)=CHR$(0):IF READ
       THEN ? "Reading..."
      :GOTO 1120
LE 1110 ? "Writing..."
LI 1120 FOR I=1 TO NR:S=I
IO 1130 IF READ THEN GOSUB 1
      220:BUFFER$(I*128-12
      7)=SECTOR$:GOTO 1160
PL 1140 SECTOR$=BUFFER$(I*12
      8-127)
AM 1150 GOSUB 1220
DN 1160 IF PEEK(DSTATS)<>1 T
      HEN 1200
FB 1170 NEXT I
GM 1180 IF  NOT READ THEN EN
      D
DH 1190 ? :? :LET READ=0:GOT
      O 360
JJ 1200 ? "Error on disk acc
      ess.":? "May need fo
      rmatting.":GOTO 1040
KI 1210 REM
BL 1220 REM  SECTOR ACCESS S
      UBROUTINE
IG 1230 REM Drive ONE
IH 1240 REM Pass buffer in S
      ECTOR$
MP 1250 REM sector # in vari
      able S
EG 1260 REM READ=1 for read,
KJ 1270 REM READ=0 for write
BN 1280 BASE=3*256
GL 1290 DUNIT=BASE+1:DCOMND=
      BASE+2:DSTATS=BASE+3
NL 1300 DBUFLO=BASE+4:DBUFHI
      =BASE+5
AI 1310 DBYTLO=BASE+8:DBYTHI
      =BASE+9
JA 1320 DAUX1=BASE+10:DAUX2=
      BASE+11
PN 1330 REM DIM DSKINV$(4)
CA 1340 DSKINV$="hLS":DSKINV
      $(4)=CHR$(228)
PF 1350 POKE DUNIT,1:A=ADR(S
      ECTOR$):H=INT(A/256)
      :L=A-256*H
BP 1360 POKE DBUFHI,H
CO 1370 POKE DBUFLO,L
PD 1380 POKE DCOMND,87-5*REA
      D
AA 1390 POKE DAUX2,INT(S/256
      ):POKE DAUX1,S-PEEK(
      DAUX2)*256
KJ 1400 A=USR(ADR(DSKINV$))
KG 1410 RETURN
```

64 MLX: Machine Language Entry

```
10 REM LINES CHANGED FROM MLX
   {SPACE}VERSION 2.00 ARE 750
   ,765,770 AND 860      :rem 50
20 REM LINE CHANGED FROM MLX V
   ERSION 2.01 IS 300 :rem 147
100 PRINT"{CLR}{6}";CHR$(142);
   CHR$(8);:POKE53281,1:POKE5
   3280,1                :rem 67
101 POKE 788,52:REM DISABLE RU
   N/STOP                :rem 119
110 PRINT"{RVS}{39 SPACES}";
                         :rem 176
120 PRINT"{RVS}{14 SPACES}
   {RIGHT}{OFF}{*}£{RVS}
   {RIGHT} {RIGHT}{2 SPACES}
   {*}{OFF}{*}£{RVS}£{RVS}
   {14 SPACES}";         :rem 250
130 PRINT"{RVS}{14 SPACES}
   {RIGHT} {G}{RIGHT}
   {2 RIGHT} {OFF}£{RVS}£
   {*}{OFF}{*}{RVS}
   {14 SPACES}";         :rem 35
140 PRINT"{RVS}{41 SPACES}"
                         :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK} M
   ACHINE LANGUAGE EDITOR VER
   SION 2.02{5 DOWN}":rem 238
210 PRINT"{5}{2 UP}STARTING AD
   DRESS?{8 SPACES}{9 LEFT}";
                         :rem 143
215 INPUTS:F=1-F:C$=CHR$(31+11
   9*F)                  :rem 166
220 IFS<256OR(S>40960ANDS<4915
   2)ORS>53247THENGOSUB3000:G
   OTO210                :rem 235
225 PRINT:PRINT:PRINT :rem 180
230 PRINT"{5}{2 UP}ENDING ADDR
   ESS?{8 SPACES}{9 LEFT}";:I
   NPUTE:F=1-F:C$=CHR$(31+119
   *F)                   :rem 20
240 IFE<256OR(E>40960ANDE<4915
   2)ORE>53247THENGOSUB3000:G
   OTO230                :rem 183
250 IFE<STHENPRINTC$;"{RVS}END
   ING < START{2 SPACES}":GOS
   UB1000:GOTO 230       :rem 176
260 PRINT:PRINT:PRINT :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S
                         :rem 56
310 A=1:PRINTRIGHT$("0000"+MID
   $(STR$(AD),2),5);":";
                         :rem 33
315 FORJ=ATO6            :rem 33
320 GOSUB570:IFN=-1THENJ=J+N:G
   OTO320                :rem 228
390 IFN=-211THEN 710     :rem 62
400 IFN=-204THEN 790     :rem 64
410 IFN=-206THENPRINT:INPUT"
   {DOWN}ENTER NEW ADDRESS";Z
   Z                     :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETH
   ENPRINT"{RVS}OUT OF RANGE"
   :GOSUB1000:GOTO410:rem 225
417 IFN=-206THENAD=ZZ:PRINT:GO
   TO310                 :rem 238
420 IF N<>-196 THEN 480
                         :rem 133
430 PRINT:INPUT"DISPLAY:FROM";
   F:PRINT,"TO";:INPUTT
                         :rem 234
440 IFF<SORF>EORT<SORT>ETHENPR
   INT"AT LEAST";S;"{LEFT}, N
   OT MORE THAN";E:GOTO430
                         :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINT
   RIGHT$("0000"+MID$(STR$(I)
   ,2),5);":";           :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRIN
   TRIGHT$("00"+MID$(STR$(N),
   2),3);",";            :rem 66
460 GETA$:IFA$>""THENPRINT:PRI
   NT:GOTO310            :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI
   :PRINT:PRINT:GOTO310
                         :rem 50
480 IFN<0 THEN PRINT:GOTO310
                         :rem 168
490 A(J)=N:NEXTJ         :rem 199
500 CKSUM=AD-INT(AD/256)*256:F
   ORI=1TO6:CKSUM=(CKSUM+A(I)
   )AND255:NEXT          :rem 200
510 PRINTCHR$(18);:GOSUB570:PR
   INTCHR$(146);         :rem 94
511 IFN=-1THENA=6:GOTO315
                         :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHE
   N530                  :rem 122
520 PRINT:PRINT"LINE ENTERED W
   RONG : RE-ENTER":PRINT:GOS
   UB1000:GOTO310        :rem 176
530 GOSUB2000            :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):
   NEXT:POKE54272,0:POKE54273
   ,0                    :rem 227
550 AD=AD+6:IF AD<E THEN 310
                         :rem 212
560 GOTO 710             :rem 108
570 N=0:Z=0              :rem 88
580 PRINT"{£}";          :rem 81
581 GETA$:IFA$=""THEN581
                         :rem 95
582 AV=-(A$="M")-2*(A$=",")-3*
   (A$=".")-4*(A$="J")-5*(A$=
   "K")-6*(A$="L")       :rem 41
583 AV=AV-7*(A$="U")-8*(A$="I"
   )-9*(A$="O"):IFA$="H"THENA
   $="0"                 :rem 134
584 IFAV>0THENA$=CHR$(48+AV)
                         :rem 134
585 PRINTCHR$(20);:A=ASC(A$):I
   FA=13ORA=44ORA=32THEN670
                         :rem 229
590 IFA>128THENN=-A:RETURN
                         :rem 137
600 IFA<>20 THEN 630     :rem 10
610 GOSUB690:IFI=1ANDT=44THENN
   =-1:PRINT"{OFF}{LEFT}
   {LEFT}";:GOTO690      :rem 62
620 GOTO570              :rem 109
630 IFA<48ORA>57THEN580
                         :rem 105
640 PRINTA$;:N=N*10+A-48
                         :rem 106
650 IFN>255 THEN A=20:GOSUB100
   0:GOTO600             :rem 229
660 Z=Z+1:IFZ<3THEN580 :rem 71
670 IFZ=0THENGOSUB1000:GOTO570
                         :rem 114
680 PRINT",";:RETURN     :rem 240
690 S%=PEEK(209)+256*PEEK(210)
   +PEEK(211)            :rem 149
691 FORI=1TO3:T=PEEK(S%-I)
                         :rem 67
695 IFT<>44ANDT<>58THENPOKES%-
   I,32:NEXT             :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1)
   ;:RETURN              :rem 7
710 PRINT"{CLR}{RVS}*** SAVE *
   **{3 DOWN}"           :rem 236
715 PRINT"{2 DOWN}(PRESS {RVS}
   RETURN{OFF} ALONE TO CANCE
   L SAVE){DOWN}"        :rem 106
720 F$="":INPUT"{DOWN} FILENAM
   E";F$:IFF$=""THENPRINT:PRI
   NT:GOTO310            :rem 71
730 PRINT:PRINT"{2 DOWN}{RVS}T
   {OFF}APE OR {RVS}D{OFF}ISK
   : (T/D)"              :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"T
   HEN740                :rem 36
750 DV=1-7*(A$="D"):IFDV=8THEN
   F$="0:"+F$:OPEN15,8,15,"S"
   +F$:CLOSE15           :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK
   (54)-LEN(T$):POKE782,ZK/25
   6                     :rem 3
762 POKE781,ZK-PEEK(782)*256:P
   OKE780,LEN(T$):SYS65469
                         :rem 109
763 POKE780,1:POKE781,DV:POKE7
   82,1:SYS65466         :rem 69
765 K=S:POKE254,K/256:POKE253,
   K-PEEK(254)*256:POKE780,25
   3                     :rem 17
766 K=E+1:POKE782,K/256:POKE78
   1,K-PEEK(782)*256:SYS65496
                         :rem 235
770 IF(PEEK(783)AND1)OR(191AND
   ST)THEN780            :rem 111
775 PRINT"{DOWN}DONE.{DOWN}":G
   OTO310                :rem 113
780 PRINT"{DOWN}ERROR ON SAVE.
   {2 SPACES}TRY AGAIN.":IFDV
   =1THEN720             :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E
   2$:PRINTE1$;E2$:CLOSE15:GO
   TO720                 :rem 103
790 PRINT"{CLR}{RVS}*** LOAD *
   **{2 DOWN}"           :rem 212
795 PRINT"{2 DOWN}(PRESS {RVS}
   RETURN{OFF} ALONE TO CANCE
   L LOAD)"              :rem 82
800 F$="":INPUT"{2 DOWN} FILEN
   AME";F$:IFF$=""THENPRINT:G
   OTO310                :rem 144
810 PRINT:PRINT"{2 DOWN}{RVS}T
   {OFF}APE OR {RVS}D{OFF}ISK
   : (T/D)"              :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"T
   HEN820                :rem 34
830 DV=1-7*(A$="D"):IFDV=8THEN
   F$="0:"+F$            :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK
   (54)-LEN(T$):POKE782,ZK/25
   6                     :rem 2
841 POKE781,ZK-PEEK(782)*256:P
   OKE780,LEN(T$):SYS65469
                         :rem 107
845 POKE780,1:POKE781,DV:POKE7
   82,1:SYS65466         :rem 70
850 POKE780,0:SYS65493 :rem 11
860 IF(PEEK(783)AND1)OR(191AND
   ST)THEN870            :rem 111
865 PRINT"{DOWN}DONE.":GOTO310
                         :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.
   {2 SPACES}TRY AGAIN.{DOWN}
   ":IFDV=1THEN800       :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E
   2$:PRINTE1$;E2$:CLOSE15:GO
   TO800                 :rem 102
1000 REM BUZZER          :rem 135
1001 POKE54296,15:POKE54277,45
   :POKE54278,165        :rem 207
1002 POKE54276,33:POKE 54273,6
   :POKE54272,5          :rem 42
1003 FORT=1TO200:NEXT:POKE5427
   6,32:POKE54273,0:POKE5427
   2,0:RETURN            :rem 202
2000 REM BELL SOUND      :rem 78
2001 POKE54296,15:POKE54277,0:
   POKE54278,247         :rem 152
2002 POKE 54276,17:POKE54273,4
   0:POKE54272,0         :rem 86
2003 FORT=1TO100:NEXT:POKE5427
   6,16:RETURN           :rem 57
3000 PRINTC$;"{RVS}NOT ZERO PA
   GE OR ROM":GOTO1000
                         :rem 89
```

Saving Time And Memory: An Atari Variable Utility

P. E. Thompson

Here's a utility—actually three separate programs—which can help programmers save time and conserve memory. With them, you can list, rename, and abbreviate all variable names in a BASIC program. A thorough explanation is included.

One valuable feature of Atari BASIC is its provision for long variable names—up to 128 characters, with every character significant. Naming variables for what they represent, such as AVERAGE, rather than using a cryptic code, like A, makes programs self-documenting and more readable.

However, there are two disadvantages. First, if you want to rename a variable, it is time-consuming to go back through an entire program to edit long variable names. Second, long names lengthen program lines and make it difficult to add statements to the lines later. (Long variable names, however, don't consume much more memory; the Atari stores every character of a name only for the first reference, and uses a lookup table for subsequent references.)

The utility programs following this article solve both problems. In addition, the program steps are explained in detail so you can understand what's happening. If you wish, you can readily modify the programs or use some of the same techniques in your own programming.

The Variable Name Table

Changing variable names in Atari BASIC is actually very easy. Each name is stored in a lookup table called the Variable Name Table. When a program is being listed, BASIC references this table each time a variable appears. When you change a name in the table, every name in the program listing also changes.

You can locate the Variable Name Table by examining memory locations 130 and 131 (decimal) for the start of the table, and locations 132 and 133 for the end of the table. Try this example. Load a BASIC program, type the following line in immediate mode (no line number), and press RETURN:

```
FOR X=PEEK(130)+PEEK(131)*256
  TO PEEK(132)+PEEK(133)*256:
  PRINT CHR$(PEEK(X));:NEXT X
```

This line converts the bytes in those addresses to decimal locations by adding the least significant byte (LSB) to the product of the most significant byte (MSB) times 256. Then it displays the character representations of each memory position between those locations. These character representations are the Variable Name Table.

The table does not look quite as you might expect. Sprinkled throughout are characters in inverse video. These characters are flags which signal the end of a variable name and indicate the variable type. If the type is a scalar variable (that is, a number), the last character of the name is in inverse video. For string variables, an inverse-video dollar sign is appended. For an array variable, an inverse-video left parenthesis is added.

By scanning the table, you may see variable names that no longer appear in the program itself. This can happen for two reasons. First, mistyped commands entered in immediate mode while you're programming may be inadvertently interpreted by BASIC as variable names, and therefore added to the table. Second, variable names used in a program but later removed are not deleted from the name table.

The only way to remove these unused names is to LIST the program to tape or disk, type NEW to erase the program in memory, and then re-ENTER the program. When you load a program with ENTER, BASIC reinterprets each line as if you were typing the program manually. (That's why ENTER takes longer than LOAD.)

Using The Utilities

Follow these steps to use each utility:

1. Type each one into the computer individually from the listings here. REM lines are included strictly for reference and can be eliminated to save typing.

2. Store each utility on tape or disk using the LIST command, not SAVE.

3. Type NEW to erase any program in memory. Load the program on which the utility will operate. Make sure the program has no line numbers greater than 31999.

4. Load the appropriate utility using the ENTER command. For example, ENTER"C:" for tape or ENTER"D:*filename*" for disk. This appends the utility to the end of the program. (If your program has line numbers greater than 31999, they will be replaced by the utility.)

5. Run the utility by typing GOTO 32000 and pressing RETURN.

6. Write down the two starting addresses of the Variable Name Table. If a utility has run but an error has been made or a change is required, these addresses must be restored before any computer operations can take place. To restore the addresses, POKE 130 with the location 130 value listed by the utility, and POKE 131 with the location 131 value listed by the utility.

7. Execute the utility by responding to the screen prompts.

8. Two of the utilities—"Changer" and "Squeezer"—require that you immediately save the newly modified version of your program on tape or disk. However, you can't use the SAVE command for this purpose because the utility is merged with your program, so both would be saved together. Nor can you save the program with an immediate mode command, because the Variable Name Table would become garbled. Therefore, line 32380 in Changer and Squeezer automatically LISTs the modified program to tape or disk, separating it from the utility in the process. The utilities currently are set up to LIST your program to disk with the filename D:XXXXXXXX.XXX. You can change this filename by modifying line 32380 in both Changer and Squeezer. Also, change line 32380 in both utilities to LIST"C:",0,31999 for cassette.

9. After Changer or Squeezer has automatically saved your program, clear the computer by turning it off, then on again. Then you can load your program with the ENTER command for a test run. This assures that all pointers and the Variable Name Table will be reset to proper values.

Lister

The first utility, "Lister," lists the variable names and types. It scans the Variable Name Table looking for inverse characters to determine the type of variable. Each variable and its type are listed in the order of appearance in the table. More specific descriptions of the utility's steps are included in the program listing.

If you want hardcopy, change the PRINT statements in lines 32040, 32140, 32160, and 32180 to LPRINT.

Changer

The second utility, "Changer," displays each variable on the screen and gives you the opportunity to change it. Press RETURN to retain the variable name.

Changer operates by adding either the existing name or the changed name to a string variable called VARNAME$. This string emulates the format of the Variable Name Table, including the inverse video flags. When you've been given a chance to change all the names, Changer makes VARNAME$ the new name table. It does this by finding the starting memory location of VARNAME$ with the ADR function, then computing revised values for locations 130 and 131 and POKEing them into place.

Immediately after Changer has LISTed your program to disk or tape, reboot the computer as described in step 9.

You may want to expand the size of the new Variable Name Table. A program using many variables or long names may have insufficient space dimensioned for the new name table. If all the space in the new table is used before the utility has completed, an Error 5, String Length Error, will result. To allocate more space, change the dimensioned value for VARNAME$ in line 32020 from 500 to a larger number. You'll have to use your judgment as to the size of the number based on the number of variables and the length of the names.

Squeezer

The third utility is "Squeezer." It is similar to Changer except that each variable name is automatically replaced by a unique one- or two-letter name. This shortens the Variable Name Table to its minimum length, yet preserves the ability to LIST or modify the program. It's intended for use after a program is completely developed and debugged, particularly when the program requires as much free memory as possible. It's also helpful for shortening long program lines so you can add more statements. During testing, Squeezer reduced the size of one program by 400 bytes—an impressive figure, especially if you're working on a 16K computer.

Squeezer lists the variable type, original name, and revised name. If you want a hardcopy, add the following line:

```
32001 OPEN #1,8,0,"P:"
```

and change the PRINT statements in lines 32045, 32050, 32060, 32160, 32181, 32201, 32220, 32260, and 32300 to PRINT #1;.

As with Changer, after Squeezer has LISTed your program on disk or tape, immediately reboot the computer as described in step 9.

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

Program 1: Lister

```
FO 32000 PRINT CHR$(125):? :
         ?
GF 32010 REM INITIALIZE VARI
         ABLES
ME 32011 REM NAME$=VARIABLE
         NAME
FB 32012 REM LOCATION=MEMORY
          ADDRESS
BN 32020 CLR :DIM NAME$(128)
NP 32030 GOSUB 32040:GOTO 32
         060
BJ 32040 NAME$="":? "Type  :
          Variable Name":RET
         URN
NA 32050 REM BEGIN FOR-NEXT
         LOOP
NI 32051 REM FROM STARTING L
         OCATION
EE 32052 REM OF VARIABLE NAM
         E TABLE
JC 32053 REM TO ENDING LOCAT
         ION
IL 32060 FOR LOCATION=PEEK(1
         30)+PEEK(131)*256 T
         O PEEK(132)+PEEK(13
         3)*256-1
IN 32070 REM CHECK FOR INVER
         SE CHAR.
PB 32071 REM IF NOT,ADD TO N
         AME STRING
CN 32072 REM AND GET NEXT LO
         CATION
PI 32080 IF PEEK(LOCATION)<1
         28 THEN NAME$(LEN(N
         AME$)+1)=CHR$(PEEK(
         LOCATION)):NEXT LOC
         ATION
DM 32090 REM IF LOCATION IS
         NOT A S
KB 32091 REM THEN JUMP AHEAD
JE 32100 IF PEEK(LOCATION)<>
         164 THEN 32160
LB 32110 REM IF VARIABLE IS
         "NAME"
CB 32111 REM VARIABLES IN TH
         E UTILITY
FH 32112 REM HAVE BEEN ENCOU
         NTERED
BK 32113 REM SO WE ARE DONE
HK 32120 REM IF NAME$="NAME"
          THEN 32220
KF 32130 REM SINCE LAST CHAR
         ACTER OF

OP 32131 REM THE NAME IS S P
         RINT TYPE
PN 32132 REM "STRING" AND TH
         E NAME.
FI 32133 REM GET NEXT LOCATI
         ON
MG 32140 PRINT "STRING: ";NA
         ME$:GOTO 32200

BC 32150 REM SINCE LAST CHAR
         ACTER
LL 32151 REM OF THE NAME IS
         K
FD 32152 REM PRINT "ARRAY" A
         ND NAME.
FK 32153 REM GET NEXT LOCATI
         ON
IL 32160 IF PEEK(LOCATION)=1
         68 THEN ? "ARRAY: "
         ;NAME$:GOTO 32200
BE 32170 REM SINCE LAST CHAR
         ACTER
HM 32171 REM OF NAME IS INVE
         RSE,
CD 32172 REM CHANGE TO NORMA
         L.
IN 32173 REM PRINT "SCALAR"
         AND NAME.
IL 32174 REM GET NEXT LOCATI
         ON.
LI 32180 NAME$(LEN(NAME$)+1)
         =CHR$(PEEK(LOCATION
         )-128):? "SCALAR: "
         ;NAME$
CN 32190 REM IF SCREEN IS FU
         LL,
NF 32191 REM STOP AND WAIT F
         OR INPUT,
CI 32192 REM RESET SCREEN
KC 32193 REM FOR MORE NAMES.
AI 32200 IF PEEK(84)>20 THEN
          ? :? "PRESS RETURN
         TO CONTINUE";:INPU
         T NAME$:? CHR$(125)
         :GOSUB 32040
KE 32210 REM RESET NAME$
HH 32211 REM FOR NEXT VARIAB
         LE.
IE 32212 REM GET NEXT LOCATI
         ON.
JB 32220 NAME$="":NEXT LOCAT
         ION
NC 32240 END
```

Program 2: Changer

```
BA 32000 ? CHR$(125):? :?
GF 32010 REM INITIALIZE VARI
         ABLES
LH 32011 REM ZNAME$  =OLD NA
         ME
LK 32012 REM VARNAME$=NEW NA
         ME TABLE
AB 32013 REM RENAME$ =NEW NA
         ME
FD 32014 REM LOCATION=MEMORY
          ADDRESS
KL 32020 CLR :DIM ZNAME$(128
         ),VARNAME$(500),REN
         AME$(128)
LA 32022 ? "VALUE AT LOCATIO
         N 130: ";PEEK(130):
         ? "VALUE AT LOCATIO
         N 131: ";PEEK(131):
         ?
FI 32030 GOSUB 32040:? :GOTO
          32060
HD 32040 ZNAME$="":? "Type
         : Variable Name":RE
         TURN
NA 32050 REM BEGIN FOR-NEXT
         LOOP
NI 32051 REM FROM STARTING L
         OCATION
EE 32052 REM OF VARIABLE NAM
         E TABLE
JC 32053 REM TO ENDING LOCAT
         ION
IL 32060 FOR LOCATION=PEEK(1
         30)+PEEK(131)*256 T
         O PEEK(132)+PEEK(13
         3)*256-1
IN 32070 REM CHECK FOR INVER
         SE CHAR.
PB 32071 REM IF NOT,ADD TO N
         AME STRING
CN 32072 REM AND GET NEXT LO
         CATION
KM 32080 IF PEEK(LOCATION)<1
         28 THEN ZNAME$(LEN(
         ZNAME$)+1)=CHR$(PEE
         K(LOCATION)):NEXT L
         OCATION
PL 32090 REM IF LOCATION IS
         NOT S
KB 32091 REM THEN JUMP AHEAD
MN 32100 IF PEEK(LOCATION)<>
         164 THEN GOTO 32160
LB 32110 REM IF VARIABLE IS
         "NAME"
AE 32111 REM VARIABLES IN CH
         ANGER
FH 32112 REM HAVE BEEN ENCOU
         NTERED
BK 32113 REM SO WE ARE DONE
IG 32120 IF ZNAME$="ZNAME" T
         HEN GOTO 32340
BA 32130 REM SINCE LAST CHAR
         ACTER
NE 32131 REM OF NAME IS S
KJ 32132 REM PRINT "STRING"
         AND NAME.
FI 32133 REM GET NEXT LOCATI
         ON
NC 32140 ? "STRING: ";ZNAME$
         :GOTO 32200
BC 32150 REM SINCE LAST CHAR
         ACTER
NK 32151 REM OF NAME IS K
FD 32152 REM PRINT "ARRAY" A
         ND NAME.
FK 32153 REM GET NEXT LOCATI
         ON
OF 32160 IF PEEK(LOCATION)=1
         68 THEN ? "ARRAY :
         ";ZNAME$:GOTO 32200
BE 32170 REM SINCE LAST CHAR
         ACTER
NG 32171 REM OF ZNAME IS INV
         ERSE,
CD 32172 REM CHANGE TO NORMA
         L.
IN 32173 REM PRINT "SCALAR"
         AND NAME.
FN 32174 REM GET NEXT LOCATI
         ON
MG 32180 ZNAME$(LEN(ZNAME$)+
         1)=CHR$(PEEK(LOCATI
         ON)-128):? "SCALAR:
         ";ZNAME$
PP 32190 REM INPUT NEW NAME
         OR RETURN
LG 32191 REM IF NO CHANGE
CF 32200 ? :? "NEW NAME OR R
         ETURN":INPUT RENAME
         $
MN 32210 REM USE DOWN-ARROW
         TO SLIDE
JJ 32211 REM NAME OFF SCREEN
AO 32220 POSITION 0,7:FOR LI
         NE=1 TO 15:? CHR$(1
         57):NEXT LINE:POSIT
         ION 2,7

IP 32230 REM IF RETURN PRESS
         ED,
JN 32231 REM ADD OLD NAME TO
          NEW TABLE
OK 32240 IF LEN(RENAME$)=0 T
         HEN RENAME$=ZNAME$
NA 32250 REM IF VARIABLE IS
         ARRAY
AP 32251 REM OR STRING ADD K
          OR S
AM 32260 IF PEEK(LOCATION)=1
         64 OR PEEK(LOCATION
         )=168 THEN RENAME$(
         LEN(RENAME$)+1)=CHR
         $(PEEK(LOCATION)):G
         OTO 32300
AJ 32270 REM IF VARIABLE IS
         SCALAR
```

```
NL 32271 REM CHANGE LAST CHA
         R
KD 32272 REM TO INVERSE
FJ 32280 RENAME$(LEN(RENAME$
         ))=CHR$(ASC(RENAME$
         (LEN(RENAME$)))+128
         ):GOTO 32300
FL 32290 REM ADD NAME TO NEW
LE 32291 REM VARIABLE NAME T
         ABLE
EK 32300 VARNAME$(LEN(VARNAM
         E$)+1)=RENAME$
PP 32310 REM RESET ZNAME$
HI 32311 REM FOR NEXT VARIAB
         LE.
HC 32312 REM GET NEXT VARIAB
         LE.
ID 32320 ZNAME$="":RENAME$="
         ":NEXT LOCATION
HC 32330 REM ALL VARIABLE NA
         MES
AO 32331 REM REVISED. ADD CH
         R$(0) TO
AE 32332 REM TABLE TO INDICA
         TE END
PE 32340 VARNAME$(LEN(VARNAM
         E$)+1)=CHR$(0)
EE 32350 REM CHANGE ORIGINAL
          TABLE
NN 32351 REM ADDRESS TO NEW
         TABLE
MK 32360 POKE 131,INT(ADR(VA
         RNAME$)/256):POKE 1
         30,ADR(VARNAME$)-PE
         EK(131)*256
AC 32370 ? CHR$(125):? "NOW
         LISTING TO TAPE OR
         DISK.":? "CHANGE LI
         NE 32380 IF DESIRED
         ."
IL 32380 LIST "D:XXXXXXXX.XX
         X",0,31999
NI 32390 END
```

Program 3: Squeezer

```
BA 32000 ? CHR$(125):? :?
GK 32011 REM COUNT(0)= NUM.
         STRINGS
BE 32012 REM COUNT(1)= NUM.
         ARRAYS
EN 32013 REM COUNT(2)= NUM.
         SCALARS
EI 32014 REM COUNT(3)= COUNT
         ER
AO 32015 REM COUNT(4)= ARGUM
         ENT IN SUB
LO 32016 REM VARNAME$= NEW N
         AME TABLE
EM 32019 CLR :DIM VARNAME$(3
         84),COUNT(4)
KO 32020 ? "VALUE AT LOCATIO
         N 130: ";PEEK(130):
         ? "VALUE AT LOCATIO
         N 131: ";PEEK(131):
         ?
FM 32022 COUNT(0)=0:COUNT(1)
         =0:COUNT(2)=0:COUNT
         (3)=0:COUNT(4)=0:GO
         SUB 32040:GOTO 3212
         0
GP 32030 REM SUBROUTINES TO
         PRINT
JH 32031 REM VARIABLE NAMES
HA 32040 IF PEEK(84)<22 THEN
          GOTO 32045
PC 32041 ? "PRESS RETURN TO
         CONTINUE"
OB 32042 IF PEEK(764)<>12 TH
         EN GOTO 32042
PG 32043 POKE 764,155:? CHR$
         (125)
DB 32045 ? "  NAME: ";:RETUR
         N
ME 32050 ? "RENAME: ";:RETUR
         N
GM 32060 ? VARNAME$(LEN(VARN
         AME$));:RETURN
DA 32070 REM SUBROUTINE TO D
         ETERMINE
OP 32071 REM NEW VARIABLE NA
         ME.   IF
MB 32072 REM ALL SINGLE LETT
         ER NAMES
HO 32073 REM HAVE BEEN USED,
HK 32074 REM ADD A SECOND LE
         TTER
FH 32080 GOSUB 32050:IF COUN
         T(4)<25 THEN GOTO 3
         2090
DB 32085 COUNT(3)=1+INT(COUN
         T(4)/25):VARNAME$(L
         EN(VARNAME$)+1)=CHR
         $(64+COUNT(3)):GOSU
         B 32060
PE 32090 COUNT(3)=1+COUNT(4)
         -INT(COUNT(4)/25)*2
         5:VARNAME$(LEN(VARN
         AME$)+1)=CHR$(64+CO
         UNT(3))
IL 32100 GOSUB 32060:RETURN
LO 32110 REM CHECK ALL LOCAT
         IONS
BI 32111 REM FROM START TO E
         ND
PL 32112 REM OF NAME TABLE
CK 32120 FOR LOCATION=PEEK(1
         30)+PEEK(131)*256 T
         O PEEK(132)+PEEK(13
         3)*256
EG 32130 REM IF CHARACTER IS
          CHR$(0) THEN
DK 32131 REM END OF TABLE IS
          REACHED
CF 32140 IF PEEK(LOCATION)=0
          THEN GOTO 32300
II 32150 REM IF CHARACTER IS
          NOT
CM 32151 REM INVERSE THEN GE
         T NEXT ONE
JC 32152 REM IF INVERSE THEN
          END
MC 32153 REM OF NAME IS REAC
         HED SO
AI 32154 REM DETERMINE VARIA
         BLE TYPE
GM 32160 IF PEEK(LOCATION)<1
         27 THEN ? CHR$(PEEK
         (LOCATION));:GOTO 3
         2280
HA 32170 REM IF CHARACTER IS
          [ THEN
FJ 32171 REM TYPE IS ARRAY.
         SET
LK 32172 REM ARGUMENT TO COU
         NT, CALL
DE 32173 REM SUBROUTINE TO D
         ETERMINE
IO 32174 REM VARIABLE NAME.
         ADD [ TO
IF 32175 REM NAME, ADD 1 TO
         COUNT,
CH 32176 REM GET NEXT NAME
JL 32180 IF PEEK(LOCATION)<>
         168 THEN 32200
KK 32181 ? "("
DH 32182 COUNT(4)=COUNT(1):G
         OSUB 32080:VARNAME$
         (LEN(VARNAME$)+1)="
         (":GOSUB 32060:COUN
         T(1)=COUNT(1)+1:GOT
         O 32260
PP 32190 REM IF CHAR IS ; TH
         EN
LD 32191 REM TYPE IS STRING.
          SET
LM 32192 REM ARGUMENT TO COU
         NT, CALL
DI 32195 REM SUBROUTINE TO D
         ETERMINE
IO 32196 REM VARIABLE NAME.
         ADD ; TO
IJ 32197 REM NAME, ADD 1 TO
         COUNT,
CL 32198 REM GET NEXT NAME
ML 32200 IF PEEK(LOCATION)<>
         164 THEN GOTO 32220
JP 32201 ? "$"
CJ 32202 COUNT(4)=COUNT(0):G
         OSUB 32080:VARNAME$
         (LEN(VARNAME$)+1)="
         $":GOSUB 32060:COUN
         T(0)=COUNT(0)+1:GOT
         O 32260
OE 32210 REM VARIABLE TYPE I
         S SCALAR.

FB 32211 REM PRINT NORMAL CH
         AR
CB 32220 ? CHR$(PEEK(LOCATIO
         N)-128)
DI 32230 REM SET ARGUMENT EQ
         UAL TO NUM
DP 32231 REM OF SCALAR VARIA
         BLES FOUND
LF 32232 REM SO FAR. CALL SU
         BROUTINE
FK 32233 REM TO DETERMINE NE
         W NAME.
FB 32234 REM ADD 1 TO NUMBER
          SCALARS
PE 32240 COUNT(4)=COUNT(2):G
         OSUB 32080:COUNT(2)
         =COUNT(2)+1
CC 32250 REM SET LAST CHARAC
         TER OF
MB 32251 REM NAME TO INVERSE
NJ 32260 VARNAME$(LEN(VARNAM
         E$))=CHR$(ASC(VARNA
         ME$(LEN(VARNAME$)))
         +128):? :? :GOSUB 3
         2040
ML 32270 REM END OF FOR-NEXT
          LOOP
FF 32271 REM FOR NEXT CHAR.
JH 32280 NEXT LOCATION
AM 32290 REM HOLD LAST PARTI
         AL SCREEN
BA 32291 REM FOR DISPLAY.
EL 32292 REM ADD CHR$(0) TO
         END OF NEW NAME
CB 32293 REM NAME TABLE INDI
         CATING END
KO 32300 ? "END OF TABLE":?
         :GOSUB 32041:VARNAM
         E$(LEN(VARNAME$)+1)
         =CHR$(0)
PD 32330 REM CHANGE TABLE AD
         DRESS
BL 32340 POKE 131,INT(ADR(VA
         RNAME$)/256):POKE 1
         30,ADR(VARNAME$)-IN
         T(ADR(VARNAME$)/256
         )*256
BC 32350 REM DISPLAY WARNING
          MESSAGE
ND 32360 ? CHR$(125):? "NOW
         LISTING TO TAPE OR
         DISK":? "CHANGE LIN
         E 32380 IF DESIRED.
         "
IL 32380 LIST "D:XXXXXXXX.XX
         X",0,31999
NI 32390 END
```

©

Commodore 64 Disk Commander

Michael Kunkel

Disk access can be clumsy on the Commodore 64 because it has no special disk commands like those found on the Commodore Plus/4, 16, and PET/CBM computers. "Disk Commander" is a powerful new utility which adds the missing commands, plus a few more. It works with any 1541-compatible disk drive. Together with "TurboDisk" (COMPUTE!, April 1985), it transforms your 64 into a much faster and friendlier computer.

Because the Commodore 64 contains BASIC 2.0, designed primarily for cassette storage, disk access is a little inconvenient. For instance, you have to type LOAD"$",8 and LIST to view a disk directory—thereby wiping out a resident BASIC program—or OPEN15,8,15,"S0:*filename*": CLOSE15 just to scratch a file. If you merely want to check the disk drive error channel, you have to write a short BASIC program. Other disk operations are equally awkward. Quite a few 64 users have pined for the more powerful BASIC 3.5 or 4.0 found in some other Commodore computers. Now that wish can come true.

"Commodore 64 Disk Commander" adds 18 commands to BASIC to simplify use of the 1541 disk drive. Furthermore, the commands are flexible enough to be included within BASIC programs, and some of the commands can't be found even in BASIC 4.0. In addition, Disk Commander resides in the Random Access Memory (RAM) hidden beneath the Commodore 64's Read Only Memory (ROM), so it's relatively protected from interference with other BASIC and machine language programs. In fact, nearly all of the commands are compatible with "TurboDisk," the high-speed disk loader published in the April 1985 issue of COMPUTE!.

Typing The Program

Disk Commander is easy to prepare. Type it in with the MLX machine language entry program found elsewhere in this issue. MLX makes it easier to type machine language programs without errors because it detects most typos after you enter each program line. (See instructions in the MLX article.)

Before using MLX to enter the data for Disk Commander, clear the computer by turning it off, then on again. Then enter the following line and press RETURN:

```
POKE 44,20:POKE 5120,0:NEW
```

Now load and run MLX. Enter these responses to the prompts:

```
Starting Address? 2049
Ending Address?  4760
```

When you're done typing, MLX automatically prompts you to save the program. You can also enter the listing in multiple sittings by following the instructions in the MLX article. If you do enter the listing in more than one sitting, remember to reset the computer and enter the above POKEs and NEW each time before loading the MLX program.

Once you've saved a copy of Disk Commander, load and run it like any BASIC program. (The POKEs are *not* necessary to run the finished program.) It will copy itself into a safe place in memory and then delete its loader program from memory. Once Disk Commander is activated, even pressing RUN/STOP–RESTORE for a warm-start reset will not disable it. Disk Commander can be turned off only by a cold-start reset (shutting off the computer or typing SYS 64738).

Command Summary

Following is a list of the new commands added by Disk Commander. Each command can be abbreviated as shown in the parentheses.

DIRECTORY (DI SHIFT-R) Calls up a disk directory without erasing a resident BASIC program.

DISKST (DI SHIFT-S) Prints the error message from the disk drive error channel.

DSAVE *"filename"* (D SHIFT-S) Saves a BASIC or machine language program with the specified filename.

DLOAD *"filename"* (D SHIFT-L) Loads a BASIC or machine language program with the specified filename.

DVERIFY *"filename"* (D SHIFT-V) Compares the program specified by the filename with the program in memory.

SCRATCH *"filename"* (S SHIFT-C) Deletes the specified file from the disk. First it asks, ARE YOU SURE? If you respond by typing YES or Y, the file is scratched.

RENAME *"oldfile"* TO *"newfile"* (RE SHIFT-N) Changes the filename from *oldfile* to *newfile*.

COPY *"file1"* TO *"file2"* (CO SHIFT-P) Makes a copy of *file1* as *file2* on the same disk. However, it does not allow you to copy a file from one disk to another.

COLLECT (CO SHIFT-L) Validates the disk by reconstructing the Block Allocation Map as explained in the disk drive manual (equivalent to OPEN 15,8,15: PRINT#15,"V0:": CLOSE 15).

HEADER *"diskname,ID"* (HE SHIFT-A) Formats a disk as described in the disk drive manual. (HEADER corresponds to the disk NEW command.) The disk is given the title *diskname* for directory purposes, and the ID should be a unique two-character combination. Any files currently on the disk will be erased when this command is executed.

DOPEN#*x*,*"filename"* (D SHIFT-O) Opens a file to the disk drive as specified by *x* and the *filename*. The filename can also specify the type (P for program, S for sequential, or L and the record length for relative files) and whether the file is being opened for reading (R) or writing (W). If these parameters are not specified, certain default values are assumed. For example, DOPEN#1, "TEST" opens file 1 for reading if TEST is an existing sequential or program file, and for both reading and writing if TEST is an existing relative file. Examples: DOPEN#1, "TEST,W" opens the sequential file TEST for writing. DOPEN#1, "TEST,P,R" opens the program file TEST for reading. DOPEN#1, "TEST,L20" creates a relative file with the filename TEST and a record length of 20. (When using the abbreviated form of the command, it is not necessary to type the #. For example, you would use D SHIFT-O 1,"TEST".)

APPEND#*x*,*"filename"* (A SHIFT-P) Allows you to add data to an existing sequential file. The specified file *x* is opened for the sequential file specified by *filename*. Any data written to file *x* will be added at the end of the existing sequential file. Example: APPEND#1,"TEST": PRINT#1,"NAME": CLOSE1. This command is only for sequential files; it cannot be used to append lines to a program file. (When using the abbreviated form of the command, it is not necessary to type the #. For example, you would use A SHIFT-P 1,"TEST".)

RECORD#*x*,*y*,*z* (RE SHIFT-C) Selects record *y* and character *z* in the relative file currently open as file *x*. Examples: RECORD#1,3 selects the third record in the relative file opened as file 1. RECORD#1,3,5 selects the fifth character in the third record. (When using the abbreviated form of the command, it is not necessary to type the #. For example, you would use RE SHIFT-C 1,3,5.)

SEND (S SHIFT-E) This command has the same effect as OPEN1,8,15: PRINT#1,"*string*": CLOSE1. Example: SEND"I0" initializes the disk drive. SEND "M-R"+CHR$(3)+CHR$(5) reads the byte at location $0503 in the disk drive's memory.

BLOCKS (B SHIFT-L) Displays the number of free blocks remaining on the disk without calling up the entire directory.

PROTECT *"filename"* (PR SHIFT-O) Protects the specified file so that it cannot be scratched. Protected files are denoted on the disk directory with a less-than sign (<). Even a protected file, however, can be erased by reformatting the entire disk. Also, protected program files cannot be read by the TurboDisk utility from the April issue. Attempting to load a protected program with TurboDisk results in a ?FILE NOT FOUND ERROR.

RELEASE *"filename"* (RE SHIFT-L) Unprotects the specified file.

TRANSPOSE *"file1"* WITH *"file2"* (T SHIFT-R) Transposes the positions of two files in the disk directory. **WITH** can be abbreviated **W SHIFT-I**.

Disk Commander is extremely versatile. In addition to letting you imbed the new commands in your programs, it also lets you use them with variables, too. For instance, instead of typing this:

```
DOPEN#1,"filename"
```

you can type this:

```
A=1:A$="filename":DOPEN#A,A$
```

Together with TurboDisk, or just by itself, Disk Commander greatly enhances the power of your Commodore 64.

Commodore 64 Disk Commander

Please refer to the "MLX" article before entering this listing.

```
2049 :011,008,010,000,158,050,238
2055 :048,054,049,000,000,000,158
2061 :169,012,133,251,169,160,139
2067 :133,252,162,002,160,000,216
2073 :177,251,145,251,200,208,233
2079 :249,230,252,202,208,244,136
2085 :169,233,133,251,169,161,129
2091 :133,252,169,091,133,253,050
2097 :169,008,133,254,160,000,005
2103 :162,010,177,253,145,251,029
2109 :200,208,249,230,252,230,150
2115 :254,202,208,242,185,091,225
2121 :018,153,184,002,200,192,054
2127 :062,208,245,032,184,002,044
2133 :032,068,166,076,116,164,195
2139 :169,158,133,251,169,160,107
2145 :133,252,169,157,133,253,170
2151 :169,160,133,254,096,032,179
2157 :233,161,076,022,162,169,164
2163 :158,133,251,169,160,133,095
2169 :252,169,157,133,253,169,230
2175 :160,133,254,166,122,160,098
2181 :004,132,015,189,000,002,219
2187 :016,007,201,255,240,062,152
2193 :232,208,216,201,032,240,250
2199 :055,133,008,201,034,240,054
2205 :086,036,015,112,045,201,140
2211 :063,208,004,169,153,208,200
2217 :037,201,048,144,004,201,036
2223 :060,144,029,132,113,160,045
2229 :000,132,011,136,134,122,204
2235 :202,200,232,189,000,002,244
2241 :056,241,251,234,240,245,180
2247 :201,128,208,048,005,011,032
2253 :164,113,232,200,153,251,038
2259 :001,185,251,001,240,057,178
2265 :056,233,058,240,004,201,241
2271 :073,208,002,133,015,056,198
2277 :233,085,208,131,133,008,003
2283 :189,000,002,240,223,197,062
2289 :008,240,219,200,153,251,032
2295 :001,232,208,240,166,122,192
2301 :230,011,200,177,253,234,078
2307 :016,250,177,251,234,208,115
2313 :180,076,170,162,189,000,018
2319 :002,016,187,153,253,001,115
2325 :198,123,169,255,133,122,253
2331 :096,165,251,201,158,208,082
2337 :235,169,000,133,251,169,222
2343 :164,133,252,169,255,133,121
2349 :253,169,163,133,254,160,153
2355 :000,076,076,162,076,096,025
2361 :163,076,109,163,016,248,064
2367 :201,255,240,244,036,015,030
2373 :048,240,056,233,127,170,175
2379 :132,073,160,255,224,077,228
2385 :176,022,202,240,008,200,161
2391 :185,158,160,016,250,048,136
2397 :245,200,185,158,160,048,065
```

```
2403 :214,032,000,168,208,245,198
2409 :056,233,076,170,202,240,058
2415 :008,200,185,000,164,016,172
2421 :250,048,245,200,185,000,021
2427 :164,048,188,032,000,168,211
2433 :208,245,032,115,000,032,249
2439 :026,163,076,015,168,240,055
2445 :062,233,128,144,017,201,158
2451 :035,176,023,010,168,185,232
2457 :013,160,072,185,012,160,243
2463 :072,076,028,168,076,048,115
2469 :168,201,058,240,217,076,101
2475 :086,168,201,075,208,003,144
2481 :076,067,168,176,003,076,231
2487 :086,168,201,095,176,249,134
2493 :233,075,010,168,185,129,221
2499 :163,072,185,128,163,072,210
2505 :076,115,000,096,082,069,127
2511 :065,068,089,072,169,243,145
2517 :133,247,169,166,133,248,029
2523 :104,076,193,002,072,169,067
2529 :239,133,247,169,166,133,032
2535 :248,104,076,193,002,000,086
2541 :000,000,000,000,000,000,237
2547 :165,123,165,180,165,194,211
2553 :165,197,165,247,165,216,124
2559 :166,224,166,232,166,250,179
2565 :166,105,167,198,167,211,251
2571 :167,055,169,070,169,140,013
2577 :169,153,169,081,170,085,076
2583 :168,095,133,034,168,165,018
2589 :091,229,096,170,232,152,231
2595 :240,035,165,090,056,229,082
2601 :034,133,090,176,003,198,163
2607 :091,056,165,088,229,034,198
2613 :133,088,176,008,198,089,233
2619 :144,004,177,090,145,088,195
2625 :136,208,249,177,090,145,046
2631 :088,198,091,198,089,202,169
2637 :208,242,096,010,105,062,032
2643 :176,053,133,034,186,228,125
2649 :034,144,046,096,196,052,145
2655 :144,040,208,004,197,051,227
2661 :144,034,072,162,009,152,162
2667 :072,181,087,202,016,250,147
2673 :032,068,073,082,069,067,248
2679 :084,079,082,217,068,073,210
2685 :083,075,083,212,068,083,217
2691 :065,086,197,068,086,069,190
2697 :082,073,070,217,068,076,211
2703 :079,065,196,083,067,082,203
2709 :065,084,067,200,082,069,204
2715 :078,065,077,197,067,079,206
2721 :080,217,067,079,076,076,244
2727 :069,067,212,072,069,065,209
2733 :068,069,210,068,079,080,235
2739 :069,078,163,065,080,080,202
2745 :069,078,068,163,082,069,202
2751 :067,079,082,068,163,083,221
2757 :069,078,196,066,076,079,249
2763 :067,075,211,080,082,079,029
2769 :084,069,067,212,082,069,024
2775 :076,069,065,083,197,084,021
2781 :082,065,078,083,080,079,176
2787 :083,197,087,073,084,200,183
2793 :000,000,000,000,000,000,233
2799 :000,000,000,107,169,032,035
2805 :121,165,132,011,032,019,213
2811 :166,144,068,160,001,177,199
2817 :095,133,035,165,045,133,095
2823 :034,165,096,133,037,165,125
2829 :095,136,241,095,024,101,193
2835 :045,133,045,133,036,165,064
2841 :046,105,255,133,046,229,071
2847 :096,170,056,165,095,229,074
2853 :045,168,176,003,232,198,091
2859 :037,024,101,034,144,003,130
2865 :198,035,024,177,034,145,150
2871 :036,200,208,249,230,035,245
2877 :230,037,202,208,242,032,244
2883 :089,166,032,051,165,173,231
2889 :000,002,240,136,024,165,128
2895 :045,133,090,101,011,133,080
2901 :088,164,046,132,091,144,238
2907 :001,200,132,089,032,184,217
2913 :163,165,020,164,021,141,003
2919 :254,001,140,255,001,165,151
2925 :049,164,050,133,045,036,074
2931 :169,001,162,000,160,165,004
2937 :032,189,255,169,001,162,161
2943 :008,160,096,032,186,255,096
2949 :032,192,255,162,001,032,039
2955 :198,255,169,000,133,144,014
2961 :162,005,032,207,255,164,202
2967 :144,208,056,202,208,246,191
2973 :133,251,032,207,255,164,175
2979 :144,208,044,166,251,032,240
2985 :097,168,169,032,032,210,109
2991 :255,032,207,255,164,144,208
2997 :208,027,170,240,006,032,096
3003 :210,255,076,062,165,169,100
3009 :013,032,210,255,032,237,204
3015 :246,240,005,162,003,076,163
3021 :033,165,076,112,168,032,023
3027 :204,255,169,001,076,195,087
3033 :255,162,000,189,118,165,082
3039 :157,004,003,232,224,006,081
3045 :208,245,096,219,002,234,209
3051 :002,240,002,169,008,032,176
3057 :180,255,169,111,032,150,114
3063 :255,032,165,255,032,210,172
3069 :255,201,013,208,246,076,228
3075 :171,255,169,000,032,189,051
3081 :255,162,008,160,001,169,252
3087 :221,141,208,002,169,225,213
3093 :141,209,002,032,199,002,094
3099 :186,169,169,157,003,001,200
3105 :169,167,157,004,001,096,115
3111 :032,147,165,169,089,133,006
3117 :247,169,225,133,248,076,119
3123 :193,002,169,001,044,169,117
3129 :000,133,010,032,147,165,032
3135 :169,111,133,247,169,225,093
3141 :133,248,076,193,002,165,118
3147 :251,208,003,076,086,168,099
3153 :169,008,032,177,255,169,123
3159 :111,032,147,255,160,000,024
3165 :177,252,032,168,255,200,153
3171 :196,251,208,246,076,174,226
3177 :255,032,129,168,032,047,000
3183 :166,169,083,141,000,190,092
3189 :169,058,141,001,190,160,068
3195 :000,177,098,153,002,190,231
3201 :200,196,097,208,246,169,221
3207 :000,133,252,169,190,133,244
3213 :253,200,200,132,251,032,185
3219 :223,165,165,157,016,012,117
3225 :169,013,032,210,255,076,140
3231 :124,165,165,157,048,001,051
3237 :096,160,000,185,113,166,117
3243 :032,210,255,200,192,014,050
3249 :208,245,032,204,255,032,129
3255 :207,255,201,089,208,025,144
3261 :032,207,255,201,013,240,113
3267 :225,201,069,208,014,032,176
3273 :207,255,201,083,208,007,138
3279 :032,207,255,201,013,240,131
3285 :207,201,013,240,006,032,144
3291 :207,255,208,247,096,104,056
3297 :104,096,065,082,069,032,161
3303 :089,079,085,032,083,085,172
3309 :082,069,063,032,032,129,132
3315 :160,160,000,177,098,153,231
3321 :000,191,200,196,097,144,053
3327 :246,132,250,160,000,177,196
3333 :122,201,164,240,003,076,043
3339 :086,168,032,115,000,032,188
3345 :129,168,160,000,177,098,237
3351 :153,002,190,200,196,097,093
3357 :144,246,169,061,153,002,036
3363 :190,200,200,200,132,252,185
3369 :169,190,133,253,160,000,178
3375 :185,000,191,145,252,200,252
3381 :196,250,144,246,152,024,041
3387 :101,252,133,251,169,000,197
3393 :133,252,169,058,141,001,051
3399 :190,076,223,165,169,082,208
3405 :141,000,190,076,127,166,009
3411 :169,067,141,000,190,076,214
3417 :127,166,169,008,032,177,000
3423 :255,169,111,032,147,255,040
3429 :169,086,032,168,255,076,119
3435 :174,255,032,129,168,032,129
3441 :047,166,169,078,141,000,202
3447 :190,169,058,141,001,190,100
3453 :160,000,177,098,153,002,203
3459 :190,200,196,097,208,246,244
3465 :169,000,133,252,169,190,026
3471 :133,253,200,200,132,251,032
3477 :032,223,165,165,157,016,139
3483 :008,169,013,032,210,255,074
3489 :076,124,165,096,032,152,038
3495 :168,160,000,177,122,201,227
3501 :044,240,003,076,086,168,022
3507 :134,184,032,165,168,165,003
3513 :097,133,183,169,000,133,132
3519 :187,169,190,133,188,160,194
3525 :000,177,098,145,187,200,236
3531 :196,183,208,247,169,044,226
3537 :145,187,200,169,083,145,114
3543 :187,200,132,183,096,032,021
3549 :051,167,032,121,000,201,025
3555 :044,208,076,032,115,000,190
3561 :201,087,208,040,169,044,214
3567 :145,187,200,169,087,145,148
3573 :187,200,132,183,032,115,070
3579 :000,169,008,133,186,160,139
3585 :097,200,152,166,152,202,202
3591 :048,007,221,045,006,240,062
3597 :244,208,246,132,185,076,080
3603 :192,255,201,076,208,194,121
3609 :032,171,168,169,076,164,037
3615 :183,136,145,187,200,169,027
3621 :044,145,187,200,138,145,128
3627 :187,200,132,183,076,138,191
3633 :167,198,183,198,183,076,030
3639 :138,167,032,051,167,169,011
3645 :044,145,187,200,169,065,103
3651 :076,184,167,032,152,168,078
3657 :032,121,000,201,044,240,199
3663 :003,076,086,168,134,251,029
3669 :032,177,168,169,001,133,253
3675 :252,032,121,000,201,044,229
3681 :208,005,032,171,168,134,047
3687 :252,076,000,169,000,000,088
3693 :000,000,000,000,000,072,181
3699 :169,071,141,208,002,169,107
3705 :171,141,209,002,104,076,056
3711 :199,002,072,169,174,133,108
3717 :247,169,167,133,248,104,177
3723 :076,193,002,186,169,233,230
3729 :157,003,001,169,167,157,031
3735 :004,001,169,000,072,169,054
3741 :114,072,076,225,002,104,238
3747 :104,169,167,072,169,233,053
3753 :072,169,165,133,247,169,100
3759 :169,133,248,076,193,002,228
3765 :104,104,169,167,072,169,198
3771 :233,072,169,018,133,247,035
3777 :169,168,133,248,076,193,156
3783 :002,169,008,133,247,169,159
3789 :175,133,248,076,193,002,008
3795 :072,169,205,141,208,002,240
3801 :169,189,141,209,002,104,007
3807 :076,199,002,169,052,133,086
3813 :247,169,168,133,248,169,083
3819 :001,032,195,255,056,076,082
3825 :193,002,169,158,141,208,088
3831 :002,169,173,141,209,002,175
3837 :032,199,002,072,169,143,102
3843 :141,208,002,104,076,221,243
3849 :168,169,158,141,208,002,087
3855 :169,183,141,209,002,076,027
3861 :199,002,032,115,000,076,189
3867 :129,168,032,115,000,076,035
3873 :152,168,032,115,000,169,157
3879 :138,141,208,002,169,173,102
3885 :141,209,002,032,199,002,118
3891 :072,169,247,141,208,002,122
3897 :169,183,141,209,002,104,097
3903 :076,199,002,169,055,133,185
3909 :247,169,164,133,248,162,168
3915 :003,076,193,002,032,199,068
3921 :002,072,169,163,141,208,068
3927 :002,169,182,141,209,002,024
3933 :104,032,199,002,133,097,148
3939 :134,098,132,099,096,000,146
3945 :000,000,016,000,000,000,121
3951 :000,255,000,165,251,166,180
3957 :152,202,016,003,076,208,006
3963 :168,221,089,002,208,245,032
3969 :189,109,002,141,001,190,249
3975 :169,080,141,000,190,165,112
3981 :020,141,002,190,165,021,168
```

```
3987 :141,003,190,165,252,141,015
3993 :004,190,169,005,133,251,137
3999 :169,000,133,252,169,190,048
4005 :133,253,076,223,165,032,023
4011 :129,168,162,003,181,096,142
4017 :149,250,202,208,249,076,031
4023 :223,165,169,073,141,000,186
4029 :190,169,001,032,043,169,025
4035 :162,000,189,135,169,157,239
4041 :000,190,232,224,006,208,037
4047 :245,169,006,032,043,169,103
4053 :169,008,032,180,255,169,002
4059 :111,032,150,255,169,013,181
4065 :032,210,255,032,165,255,150
4071 :170,032,165,255,032,165,026
4077 :255,032,097,168,032,171,224
4083 :255,169,013,076,210,255,197
4089 :077,045,082,250,002,003,196
4095 :169,009,141,048,170,169,193
4101 :064,141,049,170,076,164,157
4107 :169,169,041,141,048,170,237
4113 :169,191,141,049,170,032,001
4119 :061,170,169,002,162,008,083
4125 :160,170,032,189,255,162,229
4131 :008,160,002,032,186,255,166
4137 :032,192,255,162,000,189,103
4143 :010,170,157,000,190,232,038
4149 :224,008,208,245,134,251,099
4155 :032,045,169,162,002,032,245
4161 :201,255,162,000,189,018,122
4167 :170,032,210,255,232,224,170
4173 :043,208,245,032,204,255,040
4179 :032,129,168,160,000,177,237
4185 :098,153,003,190,200,196,161
4191 :097,208,246,152,024,105,159
4197 :003,133,251,169,085,141,115
4203 :000,190,169,051,141,001,147
4209 :190,169,058,141,002,190,095
4215 :076,074,170,035,050,066,078
4221 :045,080,032,050,032,048,156
4227 :013,032,238,193,032,152,023
4233 :195,032,032,195,032,202,057
4239 :195,169,000,133,134,032,038
4245 :157,196,048,016,032,183,013
4251 :221,144,014,160,000,177,103
4257 :148,009,064,145,148,076,239
4263 :187,200,076,225,202,076,109
4269 :087,217,032,231,255,169,140
4275 :073,141,000,190,169,001,241
4281 :076,043,169,032,045,169,207
4287 :169,002,076,195,255,032,152
4293 :061,170,169,002,162,008,001
4299 :160,170,032,189,255,162,147
4305 :008,160,002,032,186,255,084
4311 :032,192,255,162,000,189,021
4317 :010,170,157,000,190,232,212
4323 :224,008,208,245,134,251,017
4329 :032,045,169,162,002,032,163
4335 :201,255,162,000,189,041,063
4341 :171,032,210,255,232,224,089
4347 :159,208,245,032,204,255,074
4353 :032,129,168,160,000,177,155
4359 :098,153,003,190,200,196,079
4365 :097,208,246,152,024,105,077
4371 :003,133,251,169,085,141,033
4377 :000,190,169,052,141,001,066
4383 :190,169,058,141,002,190,013
4389 :032,251,170,032,121,000,131
4395 :201,222,240,003,076,086,103
4401 :168,032,115,000,032,129,013
4407 :168,160,000,177,098,153,043
4413 :003,190,200,196,097,208,187
4419 :246,152,024,105,003,133,218
4425 :251,169,085,141,000,190,141
4431 :169,053,141,001,190,169,034
4437 :058,141,002,190,032,251,247
4443 :170,169,002,133,251,169,217
4449 :085,141,000,190,169,051,221
4455 :141,001,190,076,074,170,243
4461 :032,045,169,169,008,032,052
4467 :180,255,169,111,032,150,244
4473 :255,032,165,255,201,048,053
4479 :208,005,032,165,255,201,225
4485 :048,008,032,171,255,040,175
4491 :208,001,096,162,004,169,011
4497 :055,133,247,169,164,133,022
4503 :248,076,193,002,076,069,047
4509 :005,160,000,044,160,033,047
4515 :140,176,005,032,238,193,179
4521 :032,152,195,032,032,195,039
4527 :032,202,195,032,157,196,221
4533 :016,003,076,225,202,165,100
4539 :148,174,176,005,157,177,000
4545 :005,165,149,010,168,185,107
4551 :000,000,157,178,005,185,212
4557 :001,000,157,179,005,160,195
4563 :000,177,148,157,180,005,110
4569 :232,200,192,030,208,245,044
4575 :096,173,178,005,133,006,046
4581 :173,179,005,133,007,169,127
4587 :128,133,000,162,000,032,178
4593 :153,213,160,036,174,177,130
4599 :005,185,177,005,157,000,008
4605 :003,232,200,192,066,208,130
4611 :244,169,144,133,000,162,087
4617 :000,032,153,213,173,211,023
4623 :005,133,006,173,212,005,037
4629 :133,007,169,128,133,000,079
4635 :162,000,032,153,213,160,235
4641 :003,174,210,005,185,177,019
4647 :005,157,000,003,232,200,124
4653 :192,033,208,244,169,144,011
4659 :133,000,162,000,076,153,063
4665 :213,239,255,255,255,255,249
4671 :255,255,255,255,255,255,057
4677 :255,255,255,255,255,255,063
4683 :255,255,255,255,255,255,069
4689 :255,000,000,000,000,000,080
4695 :000,000,000,000,032,210,073
4701 :002,032,104,165,076,225,185
4707 :002,032,225,002,108,247,203
4713 :000,072,165,001,009,001,097
4719 :133,001,104,032,205,189,007
4725 :072,165,001,041,254,133,015
4731 :001,104,096,032,210,002,056
4737 :032,000,162,072,165,001,049
4743 :009,001,133,001,104,096,223
4749 :032,210,002,076,203,162,058
4755 :032,210,002,076,017,163,135
```

Apple Fractals

Paul W. Carlson

Fractals are receiving a great deal of attention in mathematics and computer graphics these days. They're being used for everything from simulating random plant growth to generating realistic planetary landscapes for science-fiction films. This article introduces the fascinating world of fractals with three programs that demonstrate a particular type of fractal that can be plotted on a personal computer.

The word *fractal* was coined by Benoit Mandelbrot, a pioneer in their study, to denote curves or surfaces having *fractional dimension*. The concept of fractional dimension can be illustrated as follows: A straight curve (a line) is one-dimensional, having only length. However, if the curve is infinitely long and curves about in such a manner as to completely fill an area of the plane containing it, the curve could be considered two-dimensional. A curve partially filling an area would have a fractional dimension between one and two.

Many types of fractals are *self-similar*, which means that all portions of the fractal resemble each other. Self-similarity occurs whenever the whole is an expansion of some basic building block. In the language of fractals, this basic building block is called the *generator*. The generator in the accompanying programs consists of a number of connected line segments. The curves that the programs plot are the result of starting with the generator and then repeatedly replacing each line segment with the whole generator according to a defined rule. Theoretically, these replacement cycles would continue indefinitely. In practice, the screen resolution limits the number of cycles.

The programs illustrate two types of fractal curves. The curves generated by Program 1 and Program 2 are *self-contacting*, while the curve generated by Program 3 is *self-avoiding*. A self-contacting curve touches itself but does not cross itself. A self-avoiding curve never actually touches itself although it may appear to because of the limited screen resolution.

The Dragon Sweep

Program 1 plots what Mandelbrot refers to as a "dragon sweep." It demonstrates in a step-by-step fashion how a fractal curve is filled. The generator consists of two-line segments of equal length forming a right angle. During each replacement cycle, the generator is substituted for each segment on alternating sides of the segments, that is, to the left of the first segment, to the right of the second segment, and so on. Figure 1 shows the first few cycles of substitution. The program is written in BASIC so the plotting is slow enough to let you observe the development of the curve.

The program prompts you to enter an even number of cycles (for reasons of efficiency and screen resolution, only even numbers of cycles are plotted). When a plot is complete, pressing any key clears the screen and returns you to the prompt. I recommend starting with two cycles, then four, six, etc. It takes fourteen cycles to completely fill in the "dragon," but since this requires almost two hours, you will probably want to quit after about ten cycles. You can see the complete dragon by running Program 2, which always plots the dragon first in less than 30 seconds.

Since it's not at all obvious how the program works, here's a brief explanation. NC is the number of cycles; C is the cycle number; SN is an array of segment numbers indexed by cycle number; L is the segment length; D is the segment direction, numbered clockwise from the positive x direction; and X and Y are the high-resolution screen coordinates.

Lines 100–140	Get number of cycles from user.
Line 150	Computes segment length.
Line 160	Sets starting coordinates.
Line 170	Sets segment numbers for all cycles to the first segment.
Lines 180–220	Find the direction of the segment in the last cycle by rotating the segment in each cycle that will contain the segment in the last cycle.
Lines 230–260	Increase or decrease X or Y by the segment length, depending on the segment direction.

Lines 270–290	Plot the segment and update the current segment number for each cycle.
Lines 300–320	If the segment number for cycle zero is still zero, do the next segment; otherwise, we're done.

Eight Thousand Dragons

Program 2 plots more than 8,000 different dragons. It does this by randomly determining on which side of the first segment the generator will be substituted for all cycles after the first cycle. The generator is always substituted to the left of the first segment in the first cycle to avoid plotting off the screen. Other than the randomization, this program uses the same logic as Program 1. The main part of this program is written in machine language to reduce the time required to plot a completely filled-in dragon from about two hours to less than half a minute.

All the dragons are plotted after fourteen cycles of substitution. All have exactly the same area, which equals half of the square of the distance between the first and last points plotted. All the dragons begin and end at the same points.

When a plot is complete, press the space bar to plot another dragon, or press the Q key to quit.

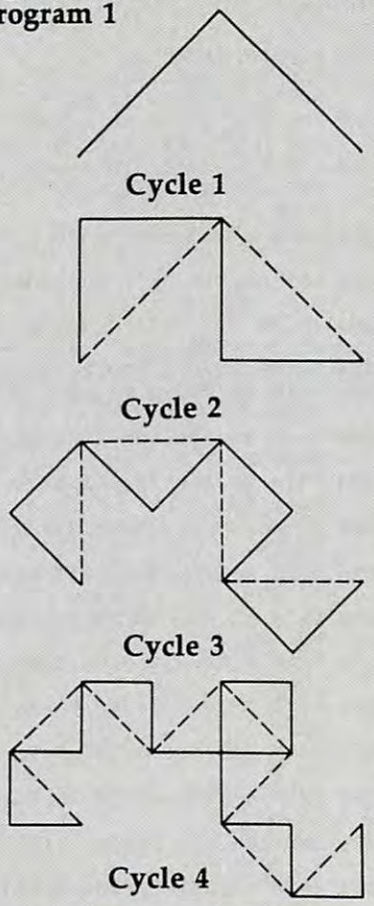
Snowflakes

Program 3 plots what Mandelbrot refers to as a "snowflake sweep." The generator, shown in Figure 2, was discovered by Mandelbrot. The segments are numbered zero through six, starting at the right. The program is basically the same as Program 1. The variables NC, C, SN, D, X, and Y represent the same values except that the direction D is numbered counterclockwise from the negative x direction. For each segment, the accompanying table gives the value of RD (relative direction), LN (length factor), and SD (flags indicating which side of the segment the generator is to be placed).

Line 20	Reads values of SD and RD. Compute LN values.
Lines 30–50	Compute delta x and delta y factors for each direction.
Lines 60–100	Get number of cycles from user.
Line 120	Sets starting coordinates.
Line 130	Sets the segment numbers for all cycles to the first segment.
Lines 140–170	Find the direction of the segment in the last cycle.
Lines 180–190	Compute the coordinates of the end of the segment, plot the segment, and update the segment numbers for each cycle.
Lines 200–220	Same as lines 300–320 in Program 1.

Like Program 1, pressing any key when a plot is complete clears the screen and brings another prompt.

Figure 1: Substitution Cycles, Program 1

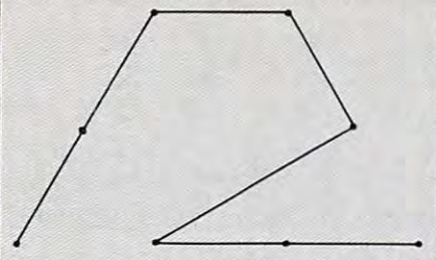


Experiment!

I hope these programs encourage you to look further into the fascinating world of fractals. Don't be afraid to experiment with the programs—try modifying the shape of the generator in Program 3, for example. Better yet, design your own generator.

These programs just begin to explore the possibilities of fractal computer graphics. There is another whole class of fractals, those generated by functions of complex variables. And then there are three-dimensional fractals. And then . . . .

Figure 2: Generator, Program 3



Values For Program 3

Segment Number SN	Relative Direction RD	Length Factor LN	Side Flag SD
0	0	1/3	0
1	0	1/3	1
2	7	√1/3	1
3	10	1/3	0
4	0	1/3	0
5	2	1/3	0
6	2	1/3	1

Program 1: The Dragon Sweep

```
1E 10 REM  PROGRAM 1
6A 20 REM
7B 30 REM  THIS PROGRAM PLOTS A
     FRACTAL "DRAGON SWEEP"
D0 40 REM  FOR AN EVEN NUMBER OF
     CYCLES (14 MAX).
6D 50 REM
9D 90 DIM SN(14)
54 100 TEXT : HOME
F1 110 PRINT "ENTER AN EVEN NO.
     OF CYCLES (2 TO 14)"
90 120 INPUT "           OR ENTER
     A ZERO TO QUIT: ";NC
A7 130 IF NC = 0 THEN END
E4 140 IF INT (NC / 2) * 2 < > N
     C OR NC < 2 OR NC > 14 TH
     EN 100
1D 150 L = 128: FOR C = 2 TO NC
     STEP 2:L = L / 2: NEXT
E8 160 X = 77:Y = 128: HGR2 : HC
     OLOR= 3: HPLOT X,Y
81 170 FOR C = 0 TO NC:SN(C) = 0
     : NEXT
43 180 D = 0: FOR C = 1 TO NC: I
     F SN(C - 1) = SN(C) THEN
     D = D - 1: GOTO 200
46 190 D = D + 1
ED 200 IF D = - 1 THEN D = 7
1C 210 IF D = 8 THEN D = 0
FD 220 NEXT
9D 230 IF D = 0 THEN X = X + L:
     GOTO 270
F0 240 IF D = 2 THEN Y = Y + L:
     GOTO 270
A4 250 IF D = 4 THEN X = X - L:
     GOTO 270
9A 260 Y = Y - L
35 270 HPLOT TO X,Y:SN(NC) = SN(
     NC) + 1
10 280 FOR C = NC TO 1 STEP - 1:
     IF SN(C) < > 2 THEN 300
9F 290 SN(C) = 0:SN(C - 1) = SN(
     C - 1) + 1: NEXT
BA 300 IF SN(0) = 0 THEN 180
D6 310 GET A$: IF A$ = "" THEN 3
     10
90 320 GOTO 100
```

Program 2: Eight Thousand Dragons

```
2E 10 REM  PROGRAM 2
6A 20 REM
6B 30 REM
92 40 REM  THIS PROGRAM PLOTS RA
     NDOM FRACTAL "DRAGON SWEEP
     S."
7C 50 REM  THE "STANDARD" DRAGON
     IS ALWAYS PLOTTED FIRST.
6E 60 REM
5F 70 REM  WHEN A PLOT IS COMPLE
     TE, PRESS THE SPACE BAR
D1 80 REM  TO PLOT ANOTHER DRAGO
     N, OR PRESS THE "Q" KEY
97 90 REM  TO EXIT THE PROGRAM.
82 100 REM
88 130 REM
6B 140 HIMEM: 16383
DB 150 FOR N = 24612 TO 24912: R
     EAD I: POKE N,I: NEXT
9F 160 FOR N = 24591 TO 24605: P
     OKE N,0: NEXT : GOTO 180
17 170 FOR N = 24593 TO 24605: P
     OKE N, INT ( RND (1) * 2)
     : NEXT
24 180 HGR2 : HCOLOR= 3: CALL 24
     619
B5 190 GET A$: IF A$ = " " THEN
     170
DB 200 IF A$ < > "Q" THEN 190
FF 210 TEXT : END
F0 220 DATA 1,2,4,8,16,32,64,169
64 230 DATA 0,141,16,96,160,14,1
     53,0
1C 240 DATA 96,136,192,255,208,2
     48,141,32
AF 250 DATA 96,162,77,142,31,96,
     160,128
22 260 DATA 140,33,96,32,248,96,
     169,0
A5 270 DATA 141,30,96,162,0,160,
     1,185
DB 280 DATA 15,96,208,20,238,30,
     96,189
26 290 DATA 0,96,217,0,96,208,26
     ,206
2B 300 DATA 30,96,206,30,96,76,1
     25,96
A8 310 DATA 206,30,96,189,0,96,2
     17,0
26 320 DATA 96,208,6,238,30,96,2
     38,30
B5 330 DATA 96,173,30,96,16,5,16
     9,7
AF 340 DATA 141,30,96,201,8,208,
     5,169
16 350 DATA 0,141,30,96,232,200,
     224,14
DB 360 DATA 208,189,170,208,20,1
     73,31,96
07 370 DATA 24,105,1,141,31,96,1
     73,32
4D 380 DATA 96,105,0,141,32,96,7
     6,210
7A 390 DATA 96,224,2,208,6,238,3
     3,96
44 400 DATA 76,210,96,224,4,208,
     20,173
0C 410 DATA 31,96,56,233,1,141,3
     1,96
53 420 DATA 173,32,96,233,0,141,
     32,96
E1 430 DATA 76,210,96,206,33,96,
     32,248
15 440 DATA 96,238,14,96,160,14,
     162,13
6B 450 DATA 185,0,96,201,2,208,1
     2,169
B4 460 DATA 0,153,0,96,254,0,96,
     202
CF 470 DATA 136,208,237,173,0,96
     ,208,3
E1 480 DATA 76,74,96,96,173,33,9
     6,10
D1 490 DATA 10,41,28,9,64,133,27
     ,173
28 500 DATA 33,96,74,74,74,74,41
     ,3
FF 510 DATA 5,27,133,27,173,33,9
     6,41
45 520 DATA 192,72,106,133,26,10
     4,74,74
1F 530 DATA 74,5,26,133,26,173,3
     1,96
8F 540 DATA 141,34,96,173,32,96,
     141,35
66 550 DATA 96,56,160,255,200,17
     3,34,96
0C 560 DATA 233,7,141,34,96,173,
     35,96
35 570 DATA 233,0,141,35,96,16,2
     37,173
FC 580 DATA 34,96,105,7,170,189,
     36,96
71 590 DATA 17,26,145,26,96
```

Program 3: The Snowflake Sweep

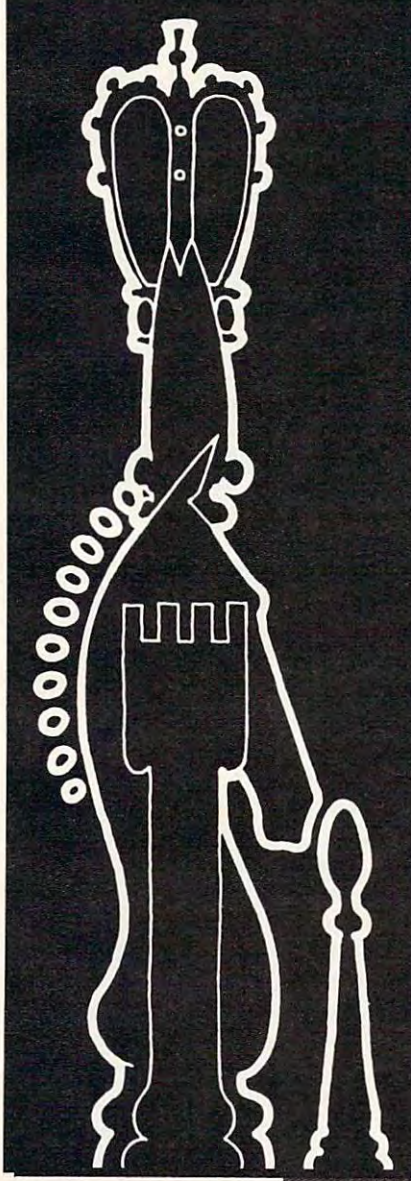
```
3E 10 REM  PROGRAM 3
6A 20 REM
B0 30 REM  THIS PROGRAM PLOTS A
     FRACTAL "SNOWFLAKE SWEEP"
6C 40 REM
9C 50 DIM DX(11),DY(11):M = 7 /
     6
1C 60 FOR N = 0 TO 6: READ SD(N)
     ,RD(N):LN(N) = 1 / 3: NEXT
     :LN(2) = SQR (LN(1))
F1 70 A = 0: FOR D = 6 TO 11:DX(
     D) = COS (A):DY(D) = SIN (
     A)
BC 80 A = A + 0.52359879: NEXT
EB 90 FOR D = 0 TO 5:DX(D) = - D
     X(D + 6):DY(D) = - DY(D +
     6): NEXT
54 100 TEXT : HOME
B5 110 PRINT "ENTER NUMBER OF CY
     CLES ( 1 - 4 )"
90 120 INPUT "           OR ENTER
     A ZERO TO QUIT: ";NC
A7 130 IF NC = 0 THEN END
1A 140 IF NC > 4 THEN 100
9D 150 HGR2 : HCOLOR= 3
BE 160 X = 235:Y = 142:TL = 162:
     HPLOT X,Y
81 170 FOR C = 0 TO NC:SN(C) = 0
     : NEXT
04 180 D = 0:L = TL:NS = 0: FOR
     C = 1 TO NC:I = SN(C):L =
     L * LN(I):J = SN(C - 1):
     NS = NS + SD(J):K = INT (
     NS / 2): IF K * 2 < > NS
     THEN D = D + 12 - RD(I):
     GOTO 200
61 190 D = D + RD(I)
92 200 IF D > 11 THEN D = D - 12
FB 210 NEXT
70 220 X = X + M * L * DX(D):Y =
     Y - L * DY(D): HPLOT TO
     X,Y:SN(NC) = SN(NC) + 1:
     FOR C = NC TO 1 STEP - 1:
     IF SN(C) < > 7 THEN 240
93 230 SN(C) = 0:SN(C - 1) = SN(
     C - 1) + 1: NEXT
C1 240 IF SN(0) = 0 THEN 180
4E 250 GET A$: IF A$ = "" THEN 2
     50
97 260 GOTO 100
41 270 DATA 0,0,1,0,1,7,0,10,0,0
     ,0,2,1,2
```

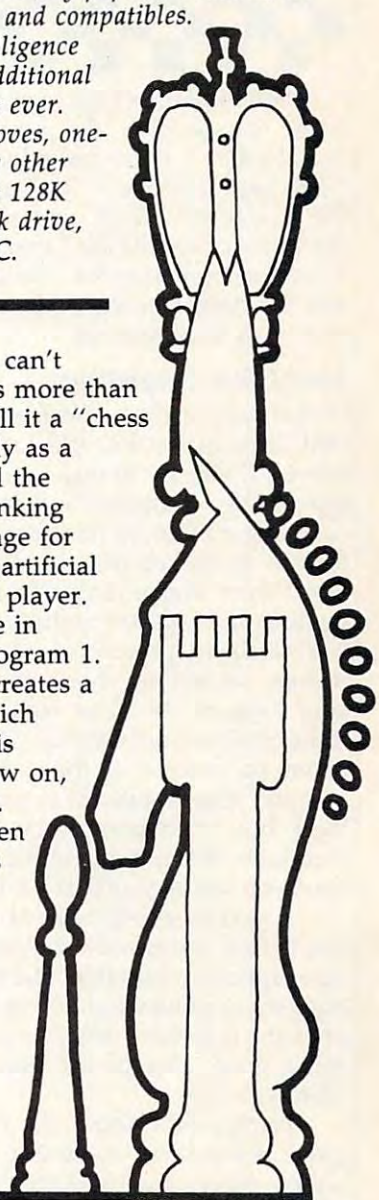
For IBM PC & PCjr

Chess

John Krause, Assistant Technical Editor

In the December 1984 issue, COMPUTE! *published "Chess" for the Commodore 64, VIC-20, Atari, and Apple computers. This month, by popular demand, we present an all-new version for the IBM PC, PCjr, and compatibles. Like the original Chess, the IBM version has intelligence routines written entirely in machine language. Additional features make it our most powerful chess program ever. It has multiple skill levels, checking for illegal moves, one- and two-player modes, reverse moving, and many other features. The program requires a PC with at least 128K RAM, color/graphics adapter, BASICA, and a disk drive, or an Enhanced Model PCjr with Cartridge BASIC.*





A computer chess game is great for those who can't always find a human opponent. But "Chess" is more than just a substitute for a live player. You might call it a "chess processor." It processes chess positions as easily as a word processor manipulates text. It contains all the features a chess player could ever want. Its thinking routines are written entirely in machine language for greater speed, and they use basic principles of artificial intelligence to simulate an actual human chess player.

Chess consists of two programs. First, type in and save each program. Then load and run Program 1. You'll have to wait about 15 seconds while it creates a BLOAD file on the disk called CHESS.BLD which contains the machine language. Once this file is created, Program 1 is no longer used. From now on, to play Chess, simply load and run Program 2.

After running Chess, you'll see a title screen for a few seconds while the computer prepares itself. Then the board is displayed with the pieces in their starting positions. You're in command of the white pieces versus the computer's black pieces on skill level 1, the easiest level. You should see a frame around the square in the lower-left corner of the board. This is the cursor which takes the place of your hand for moving and capturing pieces.

"Chess" for the IBM PC and PCjr is COMPUTE!'s *most powerful chess program to date.*

Use the cursor keys to move the frame cursor atop the piece you wish to move. Press and release the Enter key. Now move the cursor to the square on which you want to place the piece and hit Enter again. Your piece moves to the new square, and the computer responds instantly with a countermove.

Sorry, No Cheating

One of the most valuable features of IBM Chess is that it checks for illegal moves. If you try to make an illegal move, the computer buzzes and keeps your piece on its square. This feature is not perfect, however. It won't catch illegal moves involving castling or *en passant* captures. But it will catch 99 percent of all illegal moves, including those that put your king in check, as well as the more obvious ones such as moving a pawn backwards. If the computer accepts your move, it's probably legal, but not necessarily so. If the computer rejects your move, however, you can be sure that it is illegal.

If you're a beginner at chess, you'll find the move-checking feature especially valuable. Just by trying various moves and noting which ones the computer accepts, you can get a good idea of the way each piece can move.

Information about the current game is displayed at the top of the screen. *Move#* indicates the number of the move currently being made, counting from the start of the game. In chess, a move by both sides is considered one move. So, the move number is changed only after both sides have moved.

To Move indicates which side has the move. W means it is white's turn, and B means it is black's.

Normally after you move, the computer automatically makes the next move. This can be turned off by pressing the T key to switch to two-player mode. Now you can play against another person with the computer acting as referee to check for illegal moves. To switch back to one-player mode, press T again.

You can also let the computer make moves for you by pressing the M key. The side that the computer plays depends on whose turn it is. By repeatedly pressing M, you can watch the computer play itself.

Five Skill Levels

One of the advantages of a computer opponent over a human is that you can tell the computer exactly how hard you want it to try to beat you, and it obediently plays at that level of difficulty. This is important because it's no fun if you always lose or always win effortlessly.

Level shows the current skill level from 1 to 5. You can change the level at any time by pressing keys 1–5. The difference between levels is the number of moves ahead that the computer looks. On level 1, for example, it looks ahead one full move or two half-moves (its move and your reply). Each succeeding level looks ahead one more half-move than the previous level.

Alas, the smarter play on the higher levels doesn't come without a price. The further ahead the computer looks, the more moves it must examine and, hence, the longer it thinks. Here's a rundown of the five levels:

Level 1: Beginner. Thinking time: one second. Look-ahead: two half-moves. Fast but dumb.

Level 2: Intermediate. Thinking time: five seconds. Look-ahead: three half-moves. Provides a reasonable challenge for impatient players.

Level 3: Tournament. Thinking time: two minutes. Look-ahead: four half-moves. Since the usual time limit for tournament play is 40 moves in two hours, an average of three minutes per move, this level is best suited for serious players.

Level 4: Mate in two. Thinking time: 20 minutes. Look-ahead: five half-moves. Capable of solving most mate-in-two problems.

Level 5: Postal chess. Thinking time: two hours. Look-ahead: six half-moves. Simulates chess by mail where there is no time limit. Can avoid checkmate in two moves.

These thinking times are averages. The actual thinking time varies greatly depending on the position. For example, level 5 takes only five seconds with just two kings on the board. Also, these times are for the PC only. Since the PCjr runs at about two-thirds the speed of the PC, the thinking times for the PCjr are greater than the values shown above.

A Spectacular Blunder

It happens to everyone. It's inevitable. You've played for an hour, somehow managing to maneuver into a superior position in what you consider to be the best game of your life, only to throw it all away in a single, spectacular blunder.

Don't panic. You can take back the last half-move by pressing the B key. If you're in one-player mode, you need to press B again to take back your move and the computer's reply. In fact, you can press B repeatedly to take back several moves until you reach the starting position. This is possible because the computer records every move made in the game.

Another use for this feature is to allow the computer to suggest a move for you. If you don't have a good idea of where to move next, press M and the computer will move for you. If you like that move, press M again to continue with the computer's next move. But if you think you've found a better move, press B to take back the suggested move and make your own move.

Pressing the F key does the opposite of B. It moves forward through the move list up to the most advanced position. Note that every time a new move is made, the resulting position becomes the most advanced. So if you use B to backtrack to a previous position, and then make a new move, all subsequent stored moves are erased because they are no longer relevant.

If you have a printer, you can print the move list by pressing the P key. The list appears in three columns: the move numbers, white's moves, and black's moves. Each move is indicated by the square the piece moved from followed by the square it moved to. Each square is specified by its coordinates according to the numbers along the left side of the board and letters along the bottom.

You can also dump the screen image to the printer to get a hardcopy of a particularly interesting position. Before loading BASIC from DOS, type GRAPHICS with the DOS master disk in the drive. Then run Chess and press Shift-PrtSc (Fn-PrtSc on the PCjr) whenever you want to print the position.

Checkmate

The computer thinks by analyzing thousands of possible moves and countermoves and choosing what it considers to be the best move based on the relative value of the pieces. Most positions don't have just one best move but several which are equally good, in which case the computer chooses among them at random. This random factor insures that every game will be different, and makes for varied and interesting play.

The computer announces checkmate when it occurs. However, there are a few quirks in the way the computer evaluates a checkmate. On levels 3–5, it announces checkmate prematurely. When this happens, the computer has determined that it's impossible to avoid checkmate on the *next* move or two—assuming both sides make the best moves.

Also, the computer doesn't know the subtle difference between checkmate and stalemate. Consequently, when a game is stalemated, the computer announces checkmate even though the game is a draw. Since the computer tries as hard as it can to checkmate its opponent, it also tries to achieve stalemate, possibly forcing a draw when it could have won. Fortunately, this rarely happens, because a stalemate requires unusual circumstances, such as when one side has only the king remaining.

You can start a new game at any time by pressing the N key. This sets up the pieces in the starting position with white on the bottom. If you want to play the black pieces, you can press the I key to invert the board, so you still play from the bottom. As with the N command, the board is reset to the starting position. However, the N and I commands retain the move list from the previous game. This allows you to replay the game using the F command. When replaying a game, be sure to reset the board by pressing I if the game was played in the inverted mode, or N if normal mode was used.

Set Up Any Position

You don't have to begin a game from the starting position. You can set up any position and begin playing from that point. If you want, you can first clear the board by pressing the C key. To add a piece or change a piece to a different one, move the cursor to the appropriate square, hold down either Shift or Ctrl, and press P, N, B, R, Q, or K for pawn, knight, bishop, rook, queen, or king, respectively. Holding down Shift adds one of the lower player's pieces, and Ctrl adds one of the upper player's pieces. (Just remember that Ctrl is above Shift on the keyboard.) A piece can be removed from the board by pressing the space bar. Note that these changes are not stored in the move list.

These commands allow you to experiment with hypothetical or downright ridiculous positions. The position doesn't even have to be legal. Live out your fantasy by giving yourself ten queens versus the computer's lone king. Or invent your own type of chess by giving each side two kings, for example (although in this case the computer might get confused trying to determine a checkmate).

You can also set up a problem for the computer to solve, such as the mate-in-two problems published in many newspapers. To solve a mate-in-two problem, press C to clear the board, set up the position, press 4 to select level 4, and press M to start the computer thinking. After several minutes of deep thought, the computer will make a move (the solution) and announce checkmate. The only mate-in-two problems that the computer cannot solve are those which involve castling, *en passant* captures, or pawn promotion.

Special Moves

The computer never castles or captures *en passant* because, due to their complexity, these moves are not included in its thinking routine. But *you* can make these special moves. To castle, move the king *two* squares to the left or right. The rook moves automatically. To capture *en passant*, move your pawn diagonally to the proper square. The opponent's pawn is removed automatically. Remember, the computer doesn't check for illegal moves involving castling or *en passant* captures, so if you're a beginner, you should familiarize yourself with the rules on these special moves.

When a pawn reaches the opposite side of the board, it's automatically promoted to a queen. In the rare event that you would rather promote to a knight, bishop, or rook, you can easily make the change by positioning the cursor over the new queen and pressing N, B, or R with Shift or Ctrl. Note, however, that underpromotions are not stored in the move list.

Saving A Game

If you want to stop the present game and continue later, you can save the game on disk (in drive A) by pressing the S key. You'll see the prompt *Save:*. Type in a filename for your game and press Enter. The filename can be up to eight characters long. Don't type an extender; .CHS is added automatically. If a file on the disk already has the same name, it will be replaced.

To load a previously saved game, press the L key. Answer the *Load:* prompt with the filename and press Enter. (Don't type the .CHS extender.) The L command restores the game exactly as it was when it was saved. Not only the position is restored, but also the move list and even the position of the cursor.

If the computer is unable to save or load a game, an error number is displayed. See Appendix A of the *BASIC Reference Manual* for a description of the error.

Besides allowing you to continue a game at a later time, the S and L commands can be used to create a library of your best games. To do this, press N or I just before saving. The game will come up in the starting position when loaded and can be replayed using the F command.

IBM Chess Commands

B: Move backward
C: Clear board
F: Move forward
I: New game (inverted)
L: Load game
M: Computer's move
N: New game
P: Print move list
S: Save game
T: Two players
1–5: Level
Cursor Keys: Move cursor
Enter: Your move
Space Bar: Remove piece
Shift-P: Lower player's pawn
Shift-N: Lower player's knight
Shift-B: Lower player's bishop
Shift-R: Lower player's rook
Shift-Q: Lower player's queen
Shift-K: Lower player's king
Ctrl-P: Upper player's pawn
Ctrl-N: Upper player's knight
Ctrl-B: Upper player's bishop
Ctrl-R: Upper player's rook
Ctrl-Q: Upper player's queen
Ctrl-K: Upper player's king

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

Program 1: IBM Chess (Machine Language)

```
IG 10 DEF SEG=&HFFFF:IF PEEK(14)
      =253 THEN DEF SEG=&H1700:G
      OTO 30
IE 20 DEF SEG=&H1C00
EE 30 FOR I=1 TO 31:READ A$:FOR
      J=1 TO 143 STEP 2
BB 40 POKE K,VAL("&h"+MID$(A$,J,
      2)):K=K+1:IF K<825 THEN NE
      XT:NEXT
KC 50 BSAVE"chess.bld",0,825
JG 60 DATA 1EB8311C8ED88C16E1008
      926E300B8401C8ED0BC0001E80
      A008E16E1008B26E3001FCBFAB
      908008BD9C6875E00C0E2F7C60
      65E0000C606E00000B80000BFF
      FFFE908018A854C0002
OJ 70 DATA 8554005080AD88A8767008
      A9D4C00888767008A852C005B8
      88767000406508B1E2900C6876
      000005B8A8F10002A8D6000C68
      56000C083FF0075523A0E5F007
      C4B7511B000E643E440
IB 80 DATA E4403A065E00723BA25E0
      0803EDF0000741DA04C003A065
      C007528020654003A065D00751
      E80F9E57E19FE06E000C3880E5
      F008A0E4C00880E5C008A0E540
      0880E5D00C33A8D5F00
ND 90 DATA 7EF9888D5F008A9D2B008
      0C3068A871000 2A855F003A855
      E007C4083FF0174DB3A855E007
      435C38A8D4C0002BD54008AD98
      A876700803E2B00007 5063C017
      DBB7C083C007CB53C07
PF 100 DATA 74B188852C003C067404
       3CFA750AC6855F002E5A5AEB5
       C908A9D4C008A876700C68767
       00008AD9888767003B3E29007
       503E9EFFE47C6854C00148036
       2B0001FE854C008A9D4C008A
EK 110 DATA 9F6700803E2B0000750D
       80FB017C1580FB077410EB089
       080FB007D08F6DBD0E3FF971B
       0080BD4C00627CCC83FF00740
       980362B00014FE9A2FEA05C00
       00065D00FBC3803E2B000075
HO 120 DATA 5E8A9D4C0080C30A80BF
       6700007523C68554000AE838F
       F8A9D4C0080FB277D1680C314
       80BF6700007508C6855400 14E
       81DFF8A9D4C0080C30980BF67
       00007D08C685540009E807FF
DN 130 DATA 8A9D4C0080C30B80BF67
       00007D08C68554000BE8F1FEC
       38A9D4C0080C3F680BF670000
       7523C6855400F6E8DAFE8A9D4
       C0080FB517C1680C3EC80BF67
       00007508C6855400ECE8BFFE
FL 140 DATA 8A9D4C0080C3F780BF67
       00007E08C6855400F7E8A9FE8
       A9D4C0080C3F580BF6700007E
       08C6855400F5E893FEC3C6853
       40000B3008A87000088855400
       E880FEFE8534008A9D340080
OB 150 DATA FB087CE8C3C685440004
       C685340000EB1890C68544000
       8C685340004EB0B90C6854400
       08C685340000 8A9D34008A870
       80088853C0088855400E83BFE
       8A854C0002855400 8AD880BF
MI 160 DATA 670000750A8A8554000 2
       853C00EBDEFE8534008A9D340
       03A9D44007CC8C3C685340000
       B3008A87080088855400E8FFF
       DFE8534008A9D340080FB087C
       E8C30000150CF8EDEBF40813
KB 170 DATA 0BF7F5090A01F6FF2E09
       05030301000103030 5092EAA0
       16D028D029A02A702EE02
```

Program 2: IBM Chess (Main Program)

```
KM 10 CO=&H1C00:DEF SEG=&HFFFF:I
      F PEEK(14)=253 THEN CO=&H1
      700:I=1
AB 20 DA=CO+49:DEF SEG=CO:BLOAD"
      chess.bld",0:IF I THEN POK
      E 3,23:POKE 16,23
JD 30 DEF SEG=DA:GOSUB 690
GF 40 M=40:N=158:K=21
NF 50 POKE 43,1-BB:GOTO 180
KO 60 IF C2 THEN 180
NC 70 POKE 223,0:DEF SEG=CO:SOUN
      D 99,0:CALL ML:DEF SEG=DA
LO 80 IF PEEK(95)<229 AND PEEK(9
      5)>150 THEN I=0:GOTO 120
BF 90 K1=PEEK(92):K=PEEK(93):SOU
      ND 500,1:GOSUB 1190:GOSUB
      950
QI 100 IF PEEK(95)>99 OR PEEK(95
       )<28 THEN 180
NF 110 I=1
OF 120 X=I+BB+PEEK(43):IF I=0 TH
       EN POKE 43,-(PEEK(43)=0)
BN 130 GOSUB 1410:PRINT"Checkmat
       e!  ";
LO 140 IF X/2-INT(X/2) THEN PRIN
       T"White wins.":GOTO 160
IB 150 PRINT"Black wins."
FD 160 SOUND 999,9:FOR J=0 TO 20
       0:NEXT
HP 170 SOUND 260,9:FOR J=0 TO 20
       0:NEXT
KG 180 F=0:M=M-8:N=N-3
NB 190 GOSUB 680
NM 200 C$=INKEY$:IF C$="" THEN 2
       00
EP 210 IF LEN(C$)=1 THEN 270
PO 220 C=ASC(RIGHT$(C$,1)):IF C=
       75 AND M>32 THEN GOSUB 68
       0:M=M-31:K=K-1:GOTO 190
KB 230 IF C=77 AND M<249 THEN GO
       SUB 680:M=M+31:K=K+1:GOTO
        190
DN 240 IF C=72 AND N>8 THEN GOSU
       B 680:N=N-21:K=K+10:GOTO
       190
DB 250 IF C=80 AND N<155 THEN GO
       SUB 680:N=N+21:K=K-10:GOT
       O 190
BD 260 GOTO 200
QP 270 C=ASC(C$):GOSUB 1400:IF C
       <>13 OR F=0 THEN 360
GF 280 POKE 92,K1:POKE 93,K:J=PE
       EK(41):POKE 41,1:POKE 223
       ,1
FF 290 DEF SEG=CO:CALL ML:DEF SE
       G=DA
DO 300 POKE 41,J:IF PEEK(224)=0
       THEN 320
CK 310 GOSUB 1190:GOSUB 950:GOTO
        60
MG 320 X=PEEK(103+K1):IF (X=6 OR
        X=250) AND ABS(K-K1)=2 T
       HEN GOSUB 1190:GOSUB 950:
       Y=K1:K1=21-70*(X>6)-7*(K>
       K1):K=K+(K>Y)-(Y>K):MM=MM
       -1:GOSUB 1190:PR(MV)=1:GO
       SUB 950:GOTO 60
DO 330 IF PEEK(103+K) THEN 350
MI 340 IF (X=1 OR X=255) AND (AB
       S(K-K1)=9 OR ABS(K-K1)=11
       ) THEN GOSUB 1190:GOSUB 9
       50:K=K+10*(X=1)-10*(X>1):
       MM=MM-1:GOSUB 1190:PR(MV)
       =1:GOSUB 950:GOTO 60
PE 350 SOUND 100,4:F=0:POKE 43,-
       (PEEK(43)=0):GOTO 200
GA 360 IF F THEN 200
JO 370 IF C<>13 OR PEEK(103+K)=0
        THEN 410
QJ 380 IF PEEK(43) AND PEEK(103+
       K)<7 THEN 400
DH 390 IF PEEK(43) OR PEEK(103+K
       )<7 THEN 410
NO 400 K1=K:F=1:SOUND 500,1:GOTO
        200
AD 410 S=0
JL 420 IF D(S)=C THEN 450
EN 430 S=S+1:IF S<28 THEN 420
BB 440 GOTO 200
JA 450 IF S>22 THEN SOUND 500,1:
       LOCATE 1,22:PRINT C$:POKE
       41,VAL(C$):GOTO 200
HA 460 IF S=13 THEN SOUND 500,1:
       GOSUB 680:M=M+8:N=N+3:GOT
       O 70
IF 470 IF S=14 THEN SOUND 500,1:
       FOR I=0 TO 70 STEP 10:FOR
        J=0 TO 7:POKE 124+I+J,0:
       NEXT:NEXT:MX=0:MV=0:MM=0:
       BB=0:GOSUB 900:GOTO 40
NN 480 IF S<>15 OR MV=0 THEN 530
LJ 490 SOUND 500,1:POKE 43,-(PEE
       K(43)=0):GOSUB 680:GOSUB
       1200:MM=MM-1:GOSUB 1430
GD 500 IF ABS(PC(MV)-128)=122 AN
       D ABS(FR(MV)-T(MV))=2 THE
       N GOSUB 1200
FG 510 IF ABS(PC(MV)-128)=127 AN
       D PC(MV+1)=0 AND MV<MX TH
       EN GOSUB 1200
GM 520 GOTO 180
HM 530 IF S<>16 OR MV>=MX THEN 5
       80
FH 540 SOUND 500,1:POKE 43,-(PEE
       K(43)=0):GOSUB 680:GOSUB
       1210:MM=MM+1:GOSUB 1430
LO 550 IF ABS(PC(MV)-128)=122 AN
       D ABS(FR(MV)-T(MV))=2 THE
       N GOSUB 1210
KC 560 IF ABS(PC(MV)-128)=127 AN
       D PC(MV+1)=0 AND MV<MX TH
       EN GOSUB 1210
GG 570 GOTO 180
NJ 580 IF S=17 THEN BB=0:GOTO 67
       0
PO 590 IF S=18 THEN 1280
HA 600 IF S=19 THEN 1220
```

```
FE 610 IF S=20 THEN 1340
KL 620 IF S=21 THEN BB=1:GOTO 67
       0
MG 630 IF S=22 THEN SOUND 500,1:
       C2=1-C2
BI 640 IF S>12 THEN 200
HN 650 SOUND 500,1:IF S>6 THEN S
       =262-S
PI 660 POKE 103+K,S:GOSUB 950:M=
       M-8:N=N-3:GOTO 190
EG 670 SOUND 500,1:MV=0:MM=0:FOR
        I=0 TO 77:POKE I+124,BD(
       I):NEXT:GOSUB 890:GOTO 40
QL 680 PUT (M,N),F,XOR:RETURN
NH 690 KEY OFF:SCREEN 1,0:COLOR
       0,1:CLS
BG 700 POKE 41,1
AB 710 DEFINT P,N,B,R,Q,K,F
LD 720 DIM A(64),C(64),D(27),P(3
       0),N(30),B(30),R(30),Q(30
       ),K(30),F(82),FR(200),T(2
       00),PC(200),CA(200),PR(20
       0),BD(77)
PP 730 FOR I=0 TO 27:READ D(I):N
       EXT
GJ 740 LINE (0,0)-(29,19),1,BF
NF 750 GET (0,0)-(29,19),A:CLS
HF 760 LINE (0,0)-(29,19),2,BF
AH 770 GET (0,0)-(29,19),C:CLS
QO 780 LOCATE 10,18:PRINT "CHESS
       "
ON 790 LOCATE 12,15:PRINT"John K
       rause"
MN 800 FOR I=103 TO 222:POKE I,7
       :NEXT
GL 810 FOR I=0 TO 77:READ BD(I):
       POKE I+124,BD(I):NEXT
PN 820 FOR K=0 TO 30:READ P(K):N
       EXT
MP 830 FOR K=0 TO 30:READ N(K):N
       EXT
KB 840 FOR K=0 TO 30:READ B(K):N
       EXT
CD 850 FOR K=0 TO 30:READ R(K):N
       EXT
BN 860 FOR K=0 TO 30:READ Q(K):N
       EXT
IP 870 FOR K=0 TO 30:READ K(K):N
       EXT
PL 880 FOR K=0 TO 82:READ F(K):N
       EXT:CLS
JC 890 IF BB THEN POKE 127,6:POK
       E 128,5:POKE 197,250:POKE
        198,251
HM 900 LOCATE 1,5:PRINT"Move#
          Level"PEEK(41)"  To mo
       ve:":GOSUB 1430
EE 910 FOR I=0 TO 7:FOR J=0 TO 7
IM 920 H=70-10*I+J:GOSUB 960:NEX
       T:NEXT
CD 930 FOR I=1 TO 8:LOCATE 3*I-1
       +(I>4),2:PRINT 9-I:NEXT
DI 940 GOSUB 1400:RETURN
OO 950 H=K-21:I=INT(H/10):J=H-10
       *I:I=7-I
PA 960 M=31*J+40:N=21*I+11
FF 970 IF INT((I+J)/2)-(I+J)/2 T
       HEN PUT (M-8,N-3),C,PSET:
        GOTO 990
MI 980 PUT (M-8,N-3),A,PSET
HC 990 L=PEEK(124+H):IF I=0 AND
       L=1 THEN L=5:POKE 124+H,L
JF 1000 IF I=7 AND L=255 THEN L=
        251:POKE 124+H,L
PM 1010 IF L>6 THEN L=L-256
OM 1020 ON ABS(L) GOTO 1040,1050
       ,1060,1070,1080,1090
IL 1030 GOTO 1100
NF 1040 PUT (M,N),P,OR:GOTO 1100
MI 1050 PUT (M,N),N,OR:GOTO 1100
QL 1060 PUT (M,N),B,OR:GOTO 1100
QO 1070 PUT (M,N),R,OR:GOTO 1100
PB 1080 PUT (M,N),Q,OR:GOTO 1100
IG 1090 PUT (M,N),K,OR
PG 1100 IF BB THEN L=-L
KF 1110 IF L>=0 THEN RETURN
PF 1120 ON -L GOTO 1130,1140,115
       0,1160,1170,1180
FD 1130 PUT (M,N),P,XOR:RETURN
DG 1140 PUT (M,N),N,XOR:RETURN
IJ 1150 PUT (M,N),B,XOR:RETURN
IM 1160 PUT (M,N),R,XOR:RETURN
HP 1170 PUT (M,N),Q,XOR:RETURN
BC 1180 PUT (M,N),K,XOR:RETURN
QN 1190 K2=K:K=K1:MV=MV+1:PR(MV)
       =0:MM=MM+1:MX=MV:FR(MV)=
       K:PC(MV)=PEEK(103+K):POK
       E 103+K,0:GOSUB 950:K=K2
       :T(MV)=K:CA(MV)=PEEK(103
       +K):POKE 103+K,PC(MV):GO
       SUB 1430:RETURN
QN 1200 POKE 103+FR(MV),PC(MV):P
       OKE 103+T(MV),CA(MV):K=T
       (MV):GOSUB 950:K=FR(MV):
       GOSUB 950:MV=MV-1:RETURN
FO 1210 MV=MV+1:POKE 103+T(MV),P
       EEK(103+FR(MV)):POKE 103
       +FR(MV),0:K=FR(MV):GOSUB
        950:K=T(MV):GOSUB 950:R
       ETURN
DM 1220 SOUND 500,1:GOSUB 1410:I
       NPUT"Save:",N$
KO 1230 ON ERROR GOTO 1420
LA 1240 OPEN N$+".chs" FOR OUTPU
       T AS #1
NC 1250 FOR I=124 TO 201:PRINT #
       1,PEEK(I):NEXT
GM 1260 PRINT #1,PEEK(41),PEEK(4
       3),MV,MX,MM,BB,M,N,K,C2
BO 1270 FOR I=1 TO MX:PRINT #1,T
       (I),FR(I),PC(I),CA(I),PR
       (I):NEXT:CLOSE #1:ON ERR
       OR GOTO 0:GOSUB 1400:GOT
       O 200
BC 1280 SOUND 500,1:GOSUB 1410:I
       NPUT"Load:",N$
LA 1290 ON ERROR GOTO 1420
GN 1300 OPEN N$+".chs" FOR INPUT
        AS #1
FJ 1310 FOR I=124 TO 201:INPUT #
       1,J:POKE I,J:NEXT
EM 1320 INPUT #1,X,J,MV,MX,MM,BB
       ,M1,N1,K1,C2:POKE 41,X:P
       OKE 43,J
MI 1330 FOR I=1 TO MX:INPUT #1,T
       (I),FR(I),PC(I),CA(I),PR
       (I):NEXT:CLOSE #1:ON ERR
       OR GOTO 0:GOSUB 900:M=M1
       :N=N1:K=K1:GOTO 190
PC 1340 SOUND 500,1:X=0:FOR I=1
       TO MX:IF PR(I) THEN 1370
BD 1350 X=X+1:IF X/2-INT(X/2) TH
       EN LPRINT(X+1)/2" ";:GOS
       UB 1380:GOTO 1370
JH 1360 LPRINT"    ";:GOSUB 1380:
       LPRINT
HH 1370 NEXT:LPRINT:GOTO 200
HE 1380 J=INT(FR(I)/10):LPRINT C
       HR$(64+FR(I)-10*J);MID$(
       STR$(J-1),2,1)"-";
MD 1390 J=INT(T(I)/10):LPRINT CH
       R$(64+T(I)-10*J);MID$(ST
       R$(J-1),2,1);:RETURN
LC 1400 LOCATE 23,6:PRINT"A   B
         C   D   E   F   G   H"
       :RETURN
LE 1410 LOCATE 23,6:PRINT"
                           "
       :LOCATE 23,9:RETURN
LD 1420 GOSUB 1410:PRINT"Error #
       "ERR:RESUME 200
LF 1430 LOCATE 1,10:PRINT INT(MM
       /2+1)" ":LOCATE 1,35:IF
       INT(MM/2)=MM/2 THEN PRIN
       T CHR$(87):RETURN
FE 1440 PRINT CHR$(66):RETURN
NB 1450 DATA 32,80,78,66,82,81,7
       5,16,14,2,18,17,11,109,9
       9,98,102,110,108,115,112
       ,105,116,49,50,51,52,53
CL 1460 DATA 4,2,3,5,6,3,2,4,7
FO 1470 DATA 7,1,1,1,1,1,1,1,1,7
LB 1480 DATA 7,0,0,0,0,0,0,0,0,7
LE 1490 DATA 7,0,0,0,0,0,0,0,0,7
KL 1500 DATA 7,0,0,0,0,0,0,0,0,7
KO 1510 DATA 7,0,0,0,0,0,0,0,0,7
BJ 1520 DATA 7,255,255,255,255,2
       55,255,255,255,7
HO 1530 DATA 7,252,254,253,251,2
       50,253,254,252
BO 1540 DATA 28,14,0,0,0,0,3840,
       0
OE 1550 DATA 16128,192,16128,192
       ,3840,0,16128,192
AF 1560 DATA 3840,0,3840,0,16128
       ,192,-256,240
DG 1570 DATA -256,240,0,0,0,0,12
       8
PK 1580 DATA 28,14,3,0,-16381,0,
       -1021,0
LC 1590 DATA-241,192,-244,240,-2
       41,240,-241,252
PL 1600 DATA -193,252,-12481,255
       ,3852,255,16128,255
KF 1610 DATA -256,255,-253,255,-
       253,255,-253
PB 1620 DATA 28,14,-4096,240,-40
       96,240,-1021,252
CF 1630 DATA -253,60,-253,204,-2
       53,204,-253,204
KK 1640 DATA -256,240,-16384,48,
       -256,240,-16384,48
LL 1650 DATA -193,-16129,-3841,-
       3841,192,12288,-253
OP 1660 DATA 28,14,16143,207,161
       43,207,-241,255
IJ 1670 DATA 3,12,-253,252,-253,
       252,-253,252
NO 1680 DATA -253,252,-253,252,-
       253,252,3,12
EG 1690 DATA -241,255,-193,-1612
       9,-193,-16129,-193
CE 1700 DATA 28,14,-16384,192,-1
       6384,192,-16384,192
FA 1710 DATA -16192,-16192,-3133
       ,-16144,-3277,243,-3277,
       243
ED 1720 DATA -193,255,12,12,-241
       ,252,-3313,252
KI 1730 DATA -241,252,12,12,-241
       ,252,0
CD 1740 DATA 28,14,-256,192,-133
       12,192,-3268,207
KP 1750 DATA -13057,-16129,-1,-1
       6129,-16129,-16129,-3265
       ,255
FP 1760 DATA -193,255,12,12,-241
       ,252,-3313,252
KE 1770 DATA -241,252,12,12,-241
       ,252,0
CJ 1780 DATA 60,20,-1,-1,-1,-384
       1,-1,-1
GB 1790 DATA -1,-3841,252,0,0,-4
       093,252,0
DF 1800 DATA 0,-4093,252,0,0,-40
       93,252,0
EI 1810 DATA 0,-4093,252,0,0,-40
       93,252,0
EL 1820 DATA 0,-4093,252,0,0,-40
       93,252,0
EO 1830 DATA 0,-4093,252,0,0,-40
       93,252,0
EB 1840 DATA 0,-4093,252,0,0,-40
       93,252,0
EE 1850 DATA 0,-4093,252,0,0,-40
       93,252,0
EH 1860 DATA 0,-4093,252,0,0,-40
       93,252,0
II 1870 DATA 0,-4093,-1,-1,-1,-3
       841,-1,-1
IF 1880 DATA -1,-3841,0
```

Commodore Bootstrapping

Jim Butterfield, Associate Editor

Large programs are often divided into several parts and started up by a separate program called a bootstrap. This article explains how the technique works and provides a simple demonstration. The demo programs run on the Commodore 64, VIC-20, 16, Plus/4, 128 (in 64 mode), and PET/CBM, and require a disk drive.

Many complex programs—especially commercial software packages—appear on disk or tape as a collection of files. The program is broken into several pieces, and each file is one of the pieces. It's the job of a *bootstrap* program (often called a *boot)* to put all these pieces together. This makes your job easier: Just load the boot program and enter RUN. The boot brings in the other programs and gets everything going for you.

When you see a cluster of programs with similar names on a disk, look for one with BOOT in the name. That's the one to load and run. For instance, you might see these filenames in a disk directory:

```
GAME.BOOT
+GAME.SCREEN
+GAME.MUSIC
+GAME.SPRITES
+GAME.ML
+GAME.MAIN
```

In this case, you run GAME.BOOT. The boot loads each of the remaining files in turn: +GAME.SCREEN, which contains a drawing of a high-resolution screen; +GAME.MUSIC, a tune that plays during the game; +GAME.SPRITES, which contains pictures of moving objects; +GAME.ML, a machine language routine used by the main program; and finally, +GAME.MAIN, which is the actual game program. When the bootstrap program has finished its job, often it erases itself from memory.

Notice in the above example how all the filenames other than the bootstrap start with a nonalphabetic character. The computer doesn't care what the filenames look like; the symbols are a signal to you, the human part of the system, that you shouldn't load these programs directly.

In other cases, you don't get any hints from the filenames. The word BOOT doesn't appear in any filename, and the names are not distinguished by any special symbols. With a commercial program, you could try LOAD "*",8,1 to see if this starts a bootstrap sequence. If all else fails, you may have to try desperate measures: Read the instructions.

A Little History

Early computers had no Read Only Memory. The marvelous ROM that computers now use to store "canned" instructions didn't exist. When the computer was turned on, it knew nothing—not even how to load a program. Thus, early computer users were faced with a chicken-and-egg paradox: In order to load a program, they needed a program in the computer that told it how to load. How did they get this first program in? Sometimes toggle switches were used to enter individual bytes. Sometimes the computer could read a punched card and transfer a tiny program from the card into its memory.

Whatever the method, one thing was certain: The first program would be very small, containing just enough instructions to do the simplest possible loading job. And the first program to be loaded would usually be a bigger and better loading program. You had to start with a tiny loading program whose job was to bring in a bigger loading program. It seemed as though the computer was coming into action by pulling itself up "by its own bootstraps." And the term *bootstrap* came to signify any program whose job is to bring in a larger program.

Once you open the door to program-loading programs, new possibilities arise. For example, a bootstrap program can bring in several disconnected modules, each of a different type (a screen, a main BASIC program, a machine language routine, and so on). Since the modules may load into different memory areas, it's usually far easier to create them as separate files rather than paste them into one big package that loads as a single file.

A bootstrap program can also reconfigure the computer. To make room for a high-resolution graphics screen or extra sprite definitions, you may need to change the locations where BASIC starts and ends. The boot program can reconfigure BASIC memory, then load the main BASIC program into the newly defined area.

The bootstrap can make changes to allow for a particular

model of computer. If the boot program finds it is running in an 80-column machine, it might decide to load an 80-column program module instead of the 40-column one. Or, the boot could let the user decide what modules to load, depending on what peripherals are in use. Thus, the program might ask if the user has a color or black-and-white monitor, or call for the identity of any printer that is connected.

Writing A Simple Boot

Let's write a small program that uses a bootstrap technique. We'll make the program do a simple task: read a sequential file from disk. If you don't happen to have a sequential file on disk, you can create a short one called XFILE by typing the following statements in direct mode (without a line number).

```
OPEN 8,8,8,"0:XFILE,S,W"
PRINT#8,"HELLO THERE"
PRINT#8,"GOODBYE NOW"
CLOSE 8
```

Now for the program itself. Here's the plan: We'll put a main program in BASIC's usual memory area. In another area (the cassette buffer), we'll put a machine language (ML) routine that reads the file quickly and displays it on the screen. Finally, we'll need a bootstrap program to install the other two modules. We'll be using several advanced techniques, including machine language programming, program overlays, and dynamic keyboard. If you haven't seen them before, don't worry. There's no space here to explain the techniques in detail, but you can still run the programs and enjoy the view.

First you need to put an ML routine on disk. The following program is not an ML routine itself, but a generator program that creates one for you. Type in and save the program, then run it. *(Be sure to type the semicolon at the end of line 220.)* This program puts a short machine language program named "+ML" on your disk. If the computer prints ** ERROR **, you've made a typing mistake in the DATA statements. After you correct the error in the generator program and resave it, scratch the incorrect ML file by typing OPEN 15,8,15,"S0:+ML": CLOSE 15. Then reload the generator program and run it again.

If you have a Commodore 128, you can type in and save the programs in 128 mode, but before running the boot you must switch to 64 mode as explained below. The value of 144 in line 150 is correct for the VIC-20, Commodore 64 (and 128 in 64 mode), 16, and Plus/4. It needs fixing for the PET/CBM, but we'll let the boot program do that.

```
100 DATA 60,3
110 DATA 162,1
120 DATA 32,198,255
130 DATA 32,228,255
140 DATA 32,210,255
150 DATA 166,144
160 DATA 240,246
170 DATA 76,204,255
180 OPEN 4,8,4,"0:+ML,P,W"
190 FOR J=1 TO 20
200 READ X
210 T=T+X
220 PRINT#4,CHR$(X);
230 NEXT J
240 CLOSE 4
250 IF T<>3054 THEN PRINT "**
    {SPACE}ERROR **"
```

Creating The Main Program

The BASIC program is quite straightforward. Type NEW and enter:

```
100 PRINT "NAME OF SEQUENTIAL
    {SPACE}FILE":INPUTN$
110 OPEN 1,8,2,N$
120 SYS 828
130 CLOSE1
```

Now save this program by typing SAVE "0:+BASIC",8 so that the boot program can call it up when needed. *Do not try to run this program yet.* First we have to put the machine language routine it uses into memory.

Creating The Bootstrap

Type NEW again. Since the boot program varies slightly depending on the computer, we'll take care of the differences in the first line of the program. Enter line 100 as listed below for your computer.

For the 64 and VIC-20 (or 128 in 64 mode):

```
100 DATA 144,198,631
```

For the Commodore 16 or Plus/4:

```
100 DATA 144,239,1319
```

For the PET/CBM:

```
100 DATA 150,158,623
```

The three values in line 100 represent the memory locations of the computer's status variable (ST), keyboard buffer counter, and keyboard buffer, respectively. The first value adjusts the ML program to work on different machines. The other two are used to load the main BASIC program with the dynamic keyboard technique. After you enter line 100, type in the following lines as well:

```
110 IF X=1 GOTO 200
120 X=1
130 LOAD"+ML",8,1
140 STOP
```

We're using a program overlay technique here. The computer never reaches line 140, since the boot program restarts at its first statement with all variable values intact after the LOAD in line 130. Since the variable X equals 1 on the second pass, the computer leaps ahead to the rest of the program at line 200. The technique is called program overlay because it was designed to allow a second BASIC program to be loaded over an existing program while maintaining variable values. Whenever a LOAD command is executed within a program, whatever BASIC program is in memory after the LOAD is finished will begin running at its first line. We're not actually using an overlay here, since the machine language program doesn't overwrite the BASIC boot program in memory, hence the need for using X to skip the LOAD on the second pass. Without it, the program would do nothing but LOAD again and again.

Now enter the following lines, which adjust the ML program to run on different machines.

```
200 READ A,B,C
210 POKE 840,A
```

Loading the ML required a special overlay technique. Loading the BASIC program is even trickier. Since BASIC programs normally load into the same space, the new program will destroy the bootstrap as it comes in. There are several ways we can cope with this. Perhaps the easiest is to use the dynamic keyboard technique. Here goes:

```
220 D$=CHR$(17)
```

```
230 R$=CHR$(147)+D$+D$+D$+"LOA
    D"
240 N$=CHR$(34)+"+BASIC"+CHR$(
    34)
250 PRINT R$+N$+",8"+D$+D$
260 PRINT D$+D$+"RUN"+CHR$(19)
270 POKE B,2:POKE C,13:POKE C+
    1,13
```

If you've never used the dynamic keyboard technique, the above lines may look confusing. Briefly, we are telling the computer to type two commands on the screen for us. You'll see the commands when the program runs:

LOAD "+BASIC",8
RUN

The commands are carefully arranged on lines 3 and 8 of the screen. If you pressed RETURN twice—assuming the cursor was in the right place—the commands would execute, loading and running the program named +BASIC. But the boot program can press RETURN for us by putting RETURN characters, CHR$(13), into the keyboard buffer. This is a familiar trick for making Commodore computers do things that would otherwise be difficult.

Our bootstrap program is complete. Save it on disk with the name BOOT. *Be sure to save a copy of the program before you run it,* since it erases itself from memory after performing its work. (Users of the 128 *must* switch to 64 mode before running the program. Type GO64 and then enter Y at the prompt.) You should now have the following files on your disk:

BOOT (the boot program you just entered)
+ML
+BASIC
The sequential file you wish to read (**XFILE**, for example)

When you run the boot program, it loads in the ML and BASIC modules and starts things up. You'll be asked for a filename (enter XFILE if you created the sample file as shown above). After the program is finished, you can look at another file without using the boot again. Since everything's in place, just enter RUN.

This simple demonstration only hints at what a bootstrap program can do. The small but mighty bootstrap can call together many program elements to create an elegant and effective software package. ©

Atari Animation With P/M Graphics

Part 1

Robert J. Powell

Here's an easy-to-grasp explanation of how to use the Atari computer's built-in system for advanced graphics animation. This month, Part 1 takes you step by step through the fundamentals of setting up player/missile graphics in BASIC. It's intended for those with an intermediate knowledge of BASIC programming.

One of the reasons you probably bought an Atari computer was for its fine graphics capabilities. By now, maybe you've tried to write some programs with graphics and discovered that it takes considerable work to achieve the special effects you've admired in commercial software. Smooth animation seems impossible with ordinary character graphics, and moving any object across the screen using BASIC is difficult and often disappointingly slow.

The alternative is that mysterious Atari feature known as *player/missile graphics*. With P/M graphics, you can create shapes in any color and move them smoothly around the screen with relative ease. You can simulate three-dimensional movement by making some shapes pass over or beneath other shapes and the screen background. You can even detect when a shape has collided with another shape or with anything else on the screen. P/M graphics is the key to sophisticated animation on Atari computers.

Unfortunately, too many people are intimidated by P/M graphics. Although it isn't the Atari's easiest to use feature, it isn't the most difficult, either. The mystery surrounding P/M graphics started soon after the original Atari 400 and 800 computers were introduced in 1979. It was obvious from early commercial games like *Star Raiders* that some innovative graphics were involved, but Atari didn't even mention the feature in any of its manuals. Indeed, the first explanation of how P/M graphics works didn't appear until January 1981, when Atari programmer Chris Crawford wrote an article entitled "Player/Missile Graphics with the Atari Personal Computer System," which appeared in COMPUTE!. Until then, most programmers were in the dark.

A number of magazine articles and books followed, most notably *De Re Atari* by Crawford and his colleagues at Atari. But since the latest generation of Atari XL and XE owners has missed all this history, it's time for another look at P/M graphics and how it can help you add the professional touch to your programs.

A Layer Of Cellophane

First of all, P/M graphics isn't part of BASIC; there aren't even any Atari BASIC commands or keywords for handling P/M graphics. Instead, P/M graphics is built into the hardware of the computer, specifically the dedicated graphics chips unique to the Atari. Therefore, all P/M manipulation in BASIC must be done with PEEK and POKE statements.

A good way to think of P/M graphics is as a second video image overlapped onto the regular screen, like a layer of colored cellophane. That's why P/M objects can seem to travel over or behind other screen objects without erasing or disturbing them.

This system is known as *sprite graphics* on most other computers, such as the Commodore 64 and TI-99/4A. On these machines, each movable object is called a sprite; the Commodore can display up to eight at a time without special tricks, and the TI can display up to 32. Atari P/M graphics, an earlier system, consists of eight movable objects, but they're a little different than sprites. On the 64 and TI, sprites are all the same size and are roughly square (although they can be redefined as any shape, of course). On the Atari, there are four full-sized objects called *players* and four miniature objects called *missiles*. If you want, the four missiles can be grouped together to form a fifth player. And instead of being square, players and missiles are narrow strips taller than the height of the screen.

If you've never seen these strips, don't be surprised. Most programs that use P/M graphics render all but a small part of the strip invisible on the screen. The small visible part is the player or missile object you actually see. Its shape is determined by numbers POKEd by the program into a section of memory called *P/M graphics memory*. It's up to your program to set aside and protect this memory when it runs. When your program fills this memory with zeros, the whole P/M strip becomes invisible. By POKEing a few nonzero numbers into P/M memory, your program defines the shape of the visible part of the strip. This shape could be an alien, a spaceship, a cursor for a spreadsheet, or almost anything you want.

In P/M memory, each player strip is eight bits (one byte) wide, and each missile strip is two bits wide. (That's why grouping together the four two-bit missiles results in a fifth player.) All the strips are either 128 or 256 bytes tall (as described below) and extend off the visible screen in both directions.

Later, we'll explain how to determine which numbers to POKE to redefine the strips into your own shapes.

P/M Memory

Once defined, players and missiles can appear in any graphics or text mode and can be quickly moved about the screen without affecting the background graphics or text. Each player can be a different color, and P/M colors can be different than the regular screen colors—thus allowing more simultaneous colors than are normally available. With a few PEEKs, you can check for collisions between players, players and missiles, and players and screen objects (including characters). Before creating a player, let's take a look at how P/M memory is organized.

Your program must set up P/M memory to store the shape data for players. The amount of memory you set aside depends on the degree of P/M resolution desired. Two resolutions are available: single scan-line and double scan-line (a *scan-line* is the thinnest horizontal line visible on your video screen). Single-line resolution allows more detailed shapes but requires twice as much P/M memory. A single-line player is 256 bytes tall and a double-line player is 128 bytes tall. Single-line resolution requires a total of 2K, or 2,048 bytes; double-line resolution requires a total of 1K, or 1,024 bytes.

To protect P/M memory against intrusions, it's generally established near the top of user RAM just below screen memory. Another requirement is that P/M memory must start on an address that is a multiple of eight pages (2K) for single-line resolution or a multiple of four pages (1K) for double-line resolution. (A *memory page* equals 256 bytes.)

The accompanying figure shows a map of P/M memory. By custom, the starting address of P/M memory is assigned to the variable PMBASE. Since the exact memory address of PMBASE varies according to how much RAM is in the computer, which graphics mode you're using, and other factors, the map shows all other addresses as relative offsets from PMBASE. For single-line resolution, the missile data area occupies 256 bytes starting at PMBASE+768. Player data starts at PMBASE+1024 and requires 256 bytes for each player (numbered 0 through 3). For double-line resolution, all these offsets would be halved, since only half as much memory is required. Missile data would start at PMBASE+384 and player data would start at PMBASE+512.

A Bunch Of POKEs

For an example, let's write a program to set up single-line resolution P/M graphics. This requires a bunch of POKEs which may look confusing. Even if you don't fully understand the purpose of the POKEs, however, you can still use them in your programs.

First, you have to determine the number of memory pages to the starting address of P/M memory, or PMBASE. To do this, you use a memory address called RAMTOP. Logically enough, RAMTOP stores the address of the top of available RAM. That is, the computer looks at RAMTOP to calculate how much free memory is available and won't let BASIC use any memory above RAMTOP. By POKEing a lower value into RAMTOP, you can make the computer think there is less RAM and therefore free up some memory above RAMTOP (just as lowering your ceiling would create more room in your attic). The extra RAM freed up by this method is ideal for P/M memory because it's relatively safe from interference.

The value stored in RAMTOP is the number of memory pages available. How far should you lower RAMTOP? Remember that 1K is required for double-line resolution P/M graphics and 2K is required for single-line resolution P/M graphics. Since we're using single-line resolution in our example, we need to protect 2K (2,048 bytes) for P/M memory. That means we must subtract eight pages from the value in RAMTOP (8*256=2,048). The address for RAMTOP is 106 decimal, so the statement looks like this:

```
10 POKE 106,PEEK(106)-8
```

Second, you must store this new page number for RAMTOP in the *P/M base register* at memory location 54279:

```
20 POKE 54279,PEEK(106)
```

Third, select your graphics mode with the usual GRAPHICS statement, then establish the actual starting address for PMBASE. Let's stick with ordinary text mode and make the screen background black for maximum contrast:

```
30 GRAPHICS 0:SETCOLOR 2,0,0
40 PMBASE=PEEK(106)*256
```

Finally, two more POKEs are required to enable the *Direct Memory Access control register* (559 decimal) and another address which turns on P/M graphics (53277 decimal):

```
50 POKE 559,62
60 POKE 53277,3
```

(Note that for double-line P/M resolution, line 50 would be POKE 559,46.)

P/M graphics memory is now set up and activated. Before you can run the program and actually see the players, though, you have to define some shape data, assign colors, and position them on the visible part of the screen. These tasks require a few additional POKEs.

Revealing The Strips

Let's assign the colors first. There aren't any BASIC statements like COLOR or SETCOLOR for P/M graphics, so you have to POKE color values into certain memory locations instead. Each of the four players has its own color location, or *player color register*. These memory locations are 704 for player 0, 705 for player 1, 706 for player 2, and 707 for player 3. (Incidentally, the missiles lack independent color control, so missile 0 takes the same color as player 0, missile 1 takes the same color as player 1, etc.)

To determine which number to POKE into the player color registers, consult the accompanying table of Atari color numbers and use this formula:

Atari Color Numbers

0 Gray	8 Blue
1 Gold	9 Light blue
2 Orange	10 Turquoise
3 Red-orange	11 Green-blue
4 Pink	12 Green
5 Purple	13 Yellow-green
6 Red-orange	14 Orange-green
7 Blue	15 Light orange

P/M color = color number * 16 + luminance

Luminance means brightness; this should be an even number from 0 to 14. To make player 0 appear medium pink, you could POKE 704,72 (72=4*16+8). To make player 3 appear dark green, POKE 707,13*16+4. (The exact hue may vary according to how your TV or monitor is adjusted.) For our example program, we'll make the players red, green, light blue, and dark blue:

```
70 POKE 704,68:POKE 705,198:POKE
   706,168:POKE 707,148
```

Next, we want to make sure the player strips are positioned where we can see them. In addition to a color register, each player also is controlled by a *horizontal position register*. This is a memory address that determines each player's horizontal location. The registers are 53248 for player 0, 53249 for player 1, 53250 for player 2, and 53251 for player 3. You can POKE any value into these registers from 0 to 255; lower values position the player to the left, and higher values position the player to the right. However, values less than 45 begin moving the player off the left edge of the visible screen, and values greater than 205 begin moving the player off the right edge of the screen.

For this example, let's group all four players together near the right edge of the screen:

```
80 POKE 53248,160:POKE 53249,170:
   POKE 53250,180:POKE 53251,190
```

Finally, to make the player strips visible, we must fill P/M memory with shape data. For now, let's not worry about creating a fancy shape such as a spaceship. Instead, we'll reveal the players as they really are by completely filling P/M memory with 255:

```
90 FOR X=PMBASE+1024 TO
   PMBASE+2048:POKE X,255:
   NEXT X
```

Now run the program. In a few seconds, you'll see the four player strips appear on screen as line 90 fills P/M memory with the shape data.

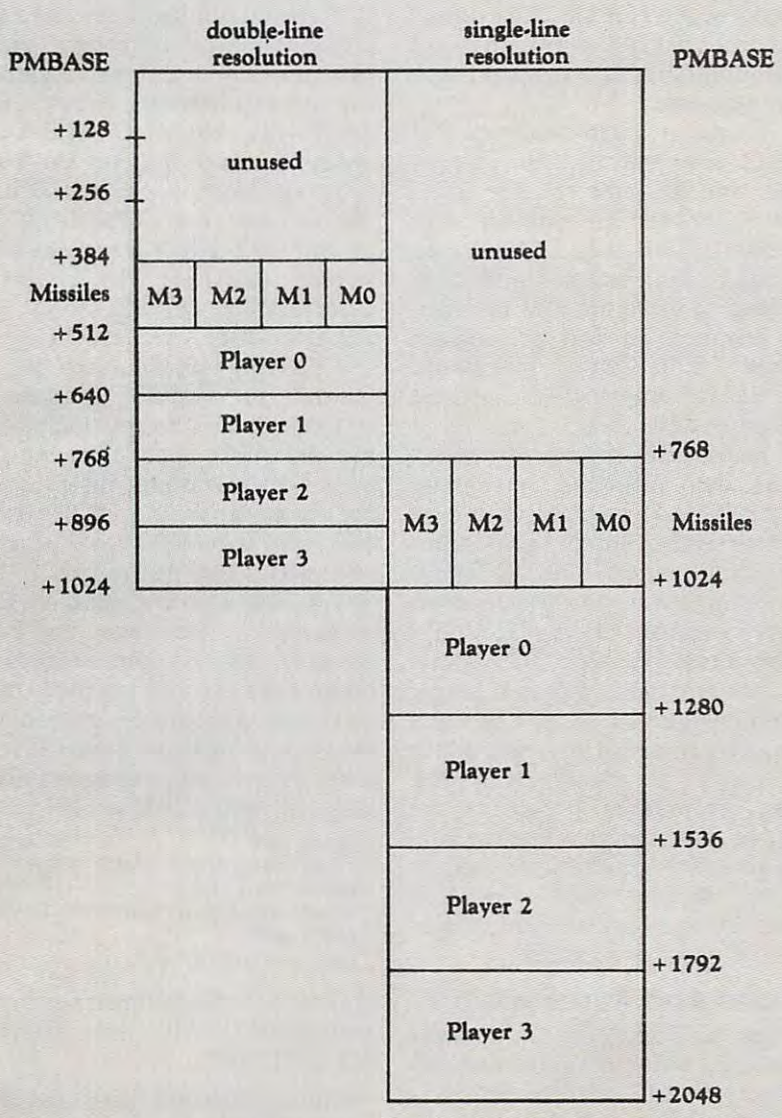
A Few Experiments

After the program stops, the READY prompt reappears and the four players remain on the screen. This is an ideal time to observe how P/M graphics works. Try these experiments:

- Type LIST. Notice how the program listing on the screen overlaps the players.
- Press SHIFT-CLEAR or CTRL-CLEAR. This clears the program listing off the screen but leaves the players undisturbed. P/M graphics, remember, are independent of regular screen graphics and text.
- In direct mode (without a line number), change the color of player 0 by POKEing a different value into the player 0 color register—for example, POKE 704,250. Also change the colors of players 1, 2, and 3 by POKEing color registers 705, 706, and 707.
- In direct mode, relocate player 0 to the left side of the screen by POKEing a lower value into the player 0 horizontal position register—say, POKE 53248,60. Relocate the other players, too, by POKEing their horizontal registers. Make a player disappear from the visible screen by POKEing a value from 0 to 45 or 205 to 255. Try stacking two players atop each other by POKEing the same value into their horizontal registers, and observe which one has display priority.

Next month, we'll show additional ways to manipulate P/M graphics and also how to transform the player strip into a shape of your own design. ©

P/M Graphics Memory Map



All About IBM Batch Files

Part 1

G. Russ Davies

IBM batch programs provide a convenient way to carry out a series of DOS (Disk Operating System) commands at once. This month we'll cover some batch programming fundamentals. Part 2 will show how to add multiple-option menus, color, and graphic displays to batch programs.

In IBM parlance a *batch* program is simply a disk file containing a series (batch) of DOS commands. The batch file executes these commands in sequence, just as if you manually typed them yourself. Batch files are identified with the .BAT filename extension. The most familiar example of a batch program is AUTOEXEC.BAT, used to issue startup commands to configure the system to your liking. Here's what a typical AUTOEXEC.BAT file might contain:

```
MODE CO80
DATE
TIME
CHKDSK
BASICA MENU
```

The first four commands in this batch file are familiar DOS commands to set the display mode to 80 columns, let you input the date and time, and analyze the disk directory. (Note that if the AUTOEXEC.BAT file doesn't include DATE and TIME, the system doesn't ask for date and time inputs when it boots.) The last command activates BASICA, then loads and runs a BASIC program named MENU. A file named AUTOEXEC.BAT differs from other batch files only in that it runs automatically when you turn on the system.

To run a batch program that doesn't automatically run, simply enter the filename at the DOS prompt (you can leave off the .BAT extension). This tells DOS to load the batch file from disk and carry out each of its commands in order. For instance, to run a program named SETUP.BAT you would type SETUP after the DOS prompt and press Enter.

This article presents several example batch programs. Since these are *not* BASIC programs, don't try to enter them with the "IBM Automatic Proofreader." The DOS manual explains how to type in short batch programs using the COPY CON: command from DOS. However, for any batch program longer than a few lines, it's easier to use a word processor or any text editor that creates standard ASCII files. Most commercial programs are suitable. You can also use the EDLIN program (on the *DOS Supplemental Programs* disk), though it lacks the convenient editing features of word processors.

Chains And Parameters

In the AUTOEXEC.BAT example above, the batch program ends by loading BASIC and running a BASIC program. A batch program can also end by returning control to DOS, or by running a second batch program (permitting you to "chain" two or more programs together). For instance, ending a batch program with SECOND causes the system to load and run the batch program named SECOND.BAT. You can also use COMMAND /C to run one batch program from within another: For example, COMMAND /C SECOND runs SECOND.BAT.

Passing parameters (information) to a batch program is straightforward. Simply include the needed information after the filename when running the program. For example, typing FIRST JULIA 123 runs the FIRST.BAT program and passes two parameters to it: a string (JULIA) and a number (123). In much the same way, one batch program can pass parameters to another. Let's use an example to demonstrate parameter passing in chained programs. Enter the following batch program and save it to disk with the filename FIRST.BAT:

```
ECHO OFF
ECHO FIRST.BAT USES FIRST P
ARAMETER: %1
ECHO PASSES %2 AND %3 TO SE
COND.BAT
REM SECOND %2 %3
```

Now enter the following program and save it with the filename SECOND.BAT:

```
ECHO SECOND.BAT USES SECOND
 PARAMETER: %1
```