PROOF OF PURCHASE

Software Giveaway with Purchase of this Magazine
See back cover

HOME COMPUTER™ magazine

FOCUSING EXCLUSIVELY ON ● APPLE ● COMMODORE ● IBM ● TEXAS INSTRUMENTS

Vol. 5 No. 1

$3.50 in USA
$4.50 in Canada

Thought Processing
—A New Frontier in Home Computing

Featuring A Ready-To-Use Thought Processor for:

- Organizing Your Thinking
- Structuring Written Text
- Producing Outlines
- Planning Projects

Special Apple, Commodore, IBM, & TI Software Programs in this Issue:

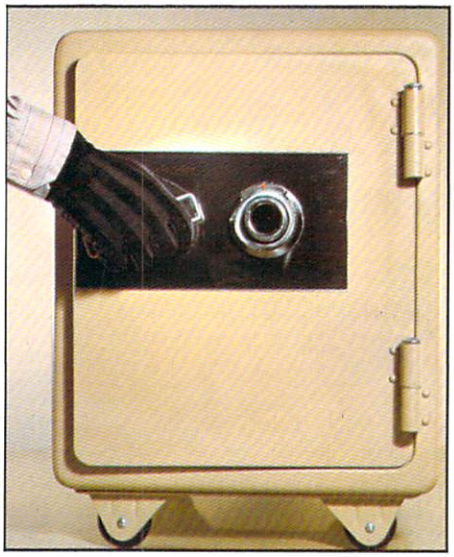
- Another Tool for Quiz Construction Set
- Electronic Backgammon
- Typing Fantasy Game
- Sailing Adventures in LOGO

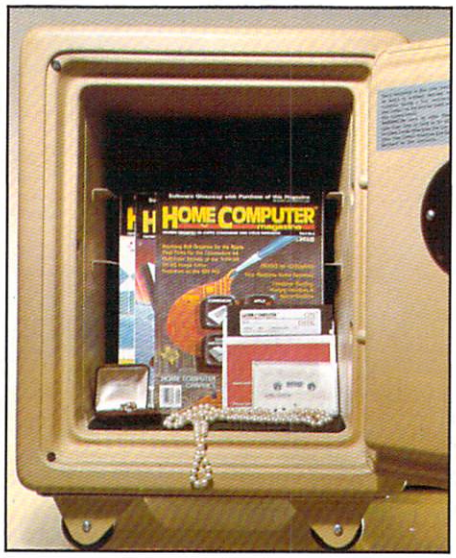
—Reviews Galore:

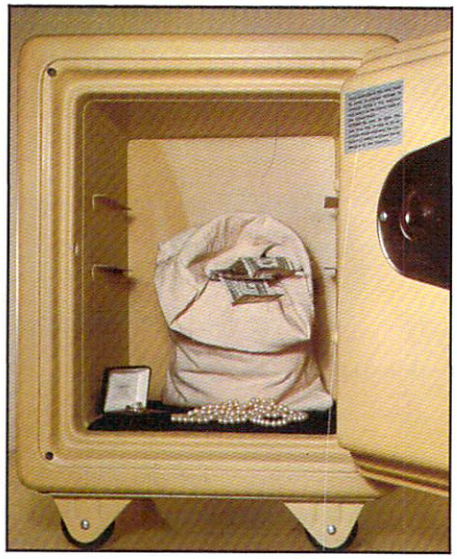
- ★ Building Robots
- ★ Ink-Jet Printing
- ★ Biofeedback Therapy
- ★ Computer Breakdancing
- ★ Ham Radio Interfacing

Contains 34 Type-In Programs!

0 12509 01
477465







MISSING ANY VALUABLES?

If you're missing any back issues of HOME COMPUTER™ magazine you're missing more than you'll ever know . . .

Having each issue of *Home Computer Magazine* readily at hand provides you with direct access to a valuable reference library of home computer knowledge —unequaled anywhere!

A valuable reference library of each *Home Computer Magazine* issue is the One Essential Peripheral™ for your home computer.

Back issues of HCM's program service— ON DISK™ or ON TAPE™ are also available.

ON DISK™ and ON TAPE™ are the convenient, accurate and affordable ways to save hundreds of typing hours.

Collect all the programs from each magazine issue on a ready-to-RUN quality floppy disk or cassette tape available in separate versions for Apple, Commodore, IBM, and Texas Instruments home computers.

"Safeguard" Your Home Computer Knowledge— Order Valuable Back Issues Today!

To Order, Use Bind-In Card at Center of Magazine.

FOR NEW READERS

The Plain & Simple Truth About HOME COMPUTER™ magazine

Chock Full of Valuable Software & How-To Articles Without Filler

Every issue is a software "horn of plenty" with dozens of type-in-and-RUN programs printed in an easy-to-read listings format. Our programs are also available on inexpensive disks or cassettes for those who prefer the convenience of ready-to-RUN software. Step-by-step tutorials round out each issue, providing the solid facts you need without fluff or filler. Thus, each issue functions as an excellent reference work, as well as a valuable software source.

No Outside Advertising

Freed from the pressures of servicing *advertisers*, we concentrate on serving our *readers*. Each issue provides uninterrupted editorial flow and graphic layouts for better comprehension—plus unbiased product reviews which focus on true strengths and weaknesses, wherever the chips may fall . . . And we don't have to worry about losing advertisers because of publishing software in the magazine that is "too good." Consequently, we can provide the best free software available anywhere.

Focused on the 4 Hot Home Brands

We are 4 system-specific magazines under one wrapper—not a sprawling, "general interest" publication which attempts to cover too wide a field, only to spread itself too thin. The other side of the coin to this focused approach is the knowledge you gain from being exposed to the many tips, ideas, and techniques we provide for 3 of the 4 systems you may not even have. You'll learn more about your Apple, Commodore, IBM, or Texas Instruments home computer from this one magazine than from a host of more limited sources.

A Balanced Mix For a Perfect Recipe

In each issue we strive for a perfect balance of productivity, entertainment, education, utilities, and computer literacy—serving the needs of novice and pro alike. Every issue is a full-course meal, with a smorgasboard of tasty dishes for all palates. Whereas other computer magazines may dish out lumps of "editorial indigestion," we serve up a satisfying blend—one digestible byte at a time.

—Welcome to Our World of Home Computing

HOME COMPUTER magazine

Home Computer Magazine (ISSN 0747-055X) is published ten times per year by Emerald Valley Publishing Co., P.O. Box 70288, Eugene, OR 97401. The editorial office is located at 1500 Valley River Drive, Suite 250, Eugene, OR 97401 (Tel. 503-485-8796). Subscription rates in U.S. and its possessions are $25 for one year, $45 for two years, and $63 for three years. In Canada and Mexico add $11 per year. Other foreign countries $43 for one year surface mail. Inquire for air delivery. Single copy price in U.S. and its possessions is $3.50, and $4.50 in Canada and Mexico. Foreign subscription payment should be in United States funds drawn on a U.S. bank. Second-class postage paid at Eugene, OR 97401, and Columbia, MO 65201.

POSTMASTER: Send all address changes to **Home Computer Magazine**, P. O. Box 70288, Eugene, OR 97401. Subscribers should send all correspondence about subscriptions to above address.

Address all editorial correspondence to the Editor at **Home Computer Magazine**, 1500 Valley River Drive, Suite 250, Eugene, OR 97401. Unacceptable manuscripts will be returned if accompanied by sufficient first class postage and self-addressed envelope. Not responsible for lost manuscripts, photos, or program media. Opinions expressed by the authors are not necessarily those of **Home Computer Magazine**. All mail directed to the Editor or to the "Letters to the Editor" column will be treated as unconditionally assigned for publication, copyright purposes, and use in any other publication or brochure, and are subject to **Home Computer Magazine's** unrestricted right to edit and comment. **Home Computer Magazine** assumes no liability for errors in articles, programs, or advertisements. Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes endorsement of the product or products by **Home Computer Magazine** or the publisher unless explicitly stated.

Publisher/Editor-in-Chief	Gary M. Kaplan
Executive Editor	David G. Brader
Managing Editor	Walter Hego
Associate Editor	Wayne Koberstein
Sr. Technical Editors	
William K. Balthrop, Roger Wood	
Technical Editors	
D. Donaldson, Tom Green, G.R. Michaels, Steven P. Nelson, Patricia Swift	
User Group Editor	Judy Campbell
Assistant Editor	Dana M. Campbell
Program Translators	
Justin Bottero, Hendrik Broekhoff, Stephen A. Cordon, Ann Dahm, Jeff Fund, Jay Jones, Robert Paschelke, Randy Thompson, Nancy Vendelin	
Asst. to the Publisher	Rhea J. Grundy
Production Manager	Norman Winney, Jr.
Creative Director	Gei-Lei Gom
Photography	
Nelson Stevens, K.D. Wainsworth	
Production Assistant	Rachel Knight
Customer Service	**Tel. (503) 341-1029**
Dealer Sales & Distribution	**Tel. (503) 341-1036**
Ken Reiling	
Main Switchboard	**Tel. (503) 485-8796**

Outside HCM

Food for Thought! It's in our magazine—and it's in our software. Even the simplest interaction with a computer involves thinking, however random—or organized—that process may be. Yet you might say that when it comes to software, you are what you boot. At *Home Computer Magazine*, we try to serve a balanced, healthy meal every issue—so that you can leave the table with your thought processes functioning even more vigorously than before. With such hearty fare to power your thinking, you may even grow to reach the stars . . .

Inside HCM

Think about it! Intelligence accumulates; it's additive. As humans evolve, they not only add to their collective *knowledge*, but they also develop new ways to solve problems—thus adding to their collective *intelligence*. Computers are but one hallmark of this process; though for our time, they may be the most important. They extend our powers of thought by doing some of our more mundane thinking *for* us, giving us the time to concentrate on other, more creative matters. They may also make us *better* thinkers, by teaching us to *organize* our thoughts.

Inside this issue of *Home Computer Magazine*, you will find a particularly powerful piece of mind-stretching software in *The Organizer*, a program that combines word-processing abilities with a built-in outline structure. Whether you are planning a project, drawing up a contract, or writing a term paper, you can place your ideas in this template and create a formidable game-plan—or even finished documents.

In future issues, we will continue to add even more useful features to this program in a similar way to what we have done with this issue's print option for *Quiz Construction Set* (originally published last issue). This versatile add-on allows teachers and others to hand out their quizzes on paper. We then follow through with a *Quiz-Print Tutorial: Accessing and Formatting Printer Output*. This article will not only provide you with more insight into the operation of *Quiz-Print*, but it also fosters a general understanding of how your computer can command a printer to produce a hard copy of virtually any document in its memory or storage media.

Don't forget to take the time to relax with a few of our thought-provoking games—such challenges as *Orbital Defender*, a space-age thriller that will make you think *fast*; and *Electronic Backgammon*, a modernized classic in which you'll have to think *hard* to beat your cybernetic opponent. And to further mix thought with pleasure, the fast-paced *Kors-Elf* game will initiate young and old alike to learning quick and accurate typing on the keyboard.

A lot of thought also goes into our product reviews. Check out our evaluation of two fascinating educational programs, *Rocky's Boots* and *Robot Odyssey I*—a pair of trend-setting games that will surely make anyone think. Then take a look with us at two other packages designed to cool troubled thoughts: *Relax* and *Calmpute*, activities that "stress" biofeedback training in the home.

Along the way, we also review some hot new games: *Breaksteet*, a refreshing break from gang violence; and *King of the Castle*, an assembly-language arcade adventure.

Some interesting hardware items help to firm-up our review section. We examine several products that link your computer to the fascinating world of ham radio communications and international wire services. We also focus our sights on a state-of-the-art laser-jet printer.

This issue premieres two new mini-features to complement *Simon Sez* for the C-64, and *Razzle Dazzle* for the 99/4A. We're proud to present *Apple Seedlings* and *IBMpressions*.

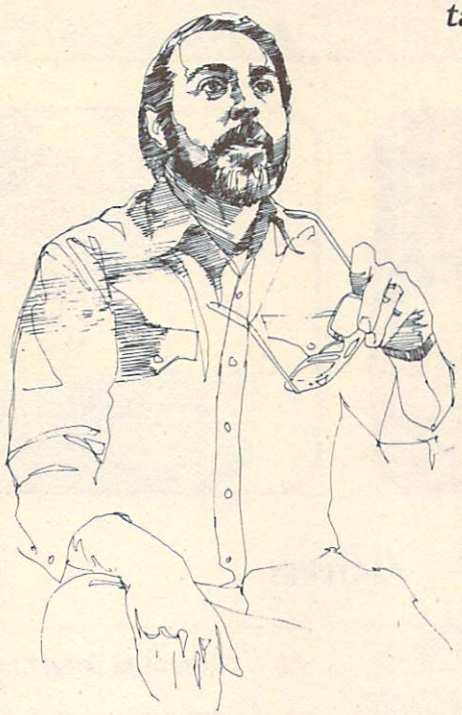
After all this, we hope you still feel adventurous enough to venture further into part 2 of our tutorial series, *Build a LOGO Adventure*. Or, if sprites are your cup of tea, try your hand at *LOGO Sailing* to regain America's Cup. Now that's something to think about!

Until next time, have fun reading, learning, and RUNing HCM

On Screen

By Gary M. Kaplan
Publisher & Editor-in-Chief

". . . Tramiel has caught a touch of 'Iacocca Fever' . . . trying to resurrect Atari while taking aim at 'The Big 3'. . ."



January's Consumer Electronics Show (CES) in Las Vegas was indeed large—with more than 100,000 attendees. It was also largely boring: Timex and Mattel are no longer players in the game. Apple and IBM did not exhibit. Texas Instruments—without a home computer line—was a skeleton of its former self. The Microsoft-sponsored MSX presence was of yawning excitement and minimal impact. And Coleco's pre-show exit from the home computer business transformed their large CES floor area into a virtual ghost town.

Stand-alone video game hardware and software is definitely *out* this year. In the past, this product category dominated the shows. Former video superstars Activision and Imagic were at the show, but had metamorphosed into smaller, "respectable-looking" home computer software houses.

Judging from CES activity, it appears that educational software has finally caught up with and overtaken entertainment software as a category. CBS and Spinnaker virtually slugged it out in their large adjoining floor areas at the show. The message was clear that Spinnaker will not have an easy time keeping its early lead in the face of the CBS conglomerate's supposed $40 million media onslaught scheduled for 1985.

Productivity software also made a much stronger showing this time around, but the "new" category that made the most impact was music—both peripherals and software. 1985 will undoubtedly be the Year of Music in the home computer world, with increasing numbers of C-64's being purchased as music system *components* rather than as stand-alone *computers*. And if Atari actually delivers its new music machines (mentioned below), the growth of this product category should escalate well into 1986.

The *real* gambling in town didn't take place in the casinos. The millions of chips being wagered were of the silicon variety—with Commodore pitted against Jack Tramiel's new Atari Corporation. Apple and IBM were clearly in the Commodore crew's sights as they uncovered and aimed their two new "big guns": the Commodore LCD portable, and the Commodore 128 Personal Computer. Designed to sell for about $500, the LCD is a souped-up (with 8 ROM-resident application programs) Commodore Plus 4, with an 80-character by 16-line LCD viewing screen, and a built-in 300-baud modem.

The Commodore 128 is really three machines in one—with three separate microprocessors and operating systems. First, there is a C-64 compatibility mode with all of the critical chips present, so the large library of C-64 software can run unchanged. There's also an enhanced 128K mode to take advantage of the larger memory and higher-resolution (up to 640 x 200) video circuitry, through a much richer BASIC (version 7.0) and a machine language facilitated by a resident monitor. In 128K mode, the machine utilizes a faster microprocessor (the 8502), and sports 80-column capability with 16 colors.

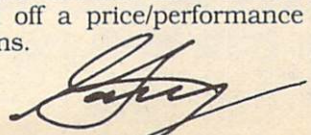
The third operating mode is CP/M 3.0, running under a Z80A microprocessor (at 4 Mhz). Perfect Software has ported their productivity series to this machine's CP/M environment. Commodore was also actively hyping its Arktronics connection with Jane 2.0—an integrated, icon-laden software package first introduced in the Apple world. Although Commodore's much-talked-about 16-bit Amiga machine wasn't introduced at this show, the firm is apparently counting on the Perfect/Jane products to carry its C-128 into the low-end business world until the more powerful machine is ready.

The 92-key Commodore 128 is slated to sell for about $300 (sans disk drive) and is said to be expandable to 512K of memory with a RAM disk option. Peripherals announced include a mouse, 300/1200 baud modems, and a "high-speed" double-sided/single-density disk drive (with 350K storage capacity in C-128 mode, 140K in C-64 mode, and up to 410K in CP/M formats).

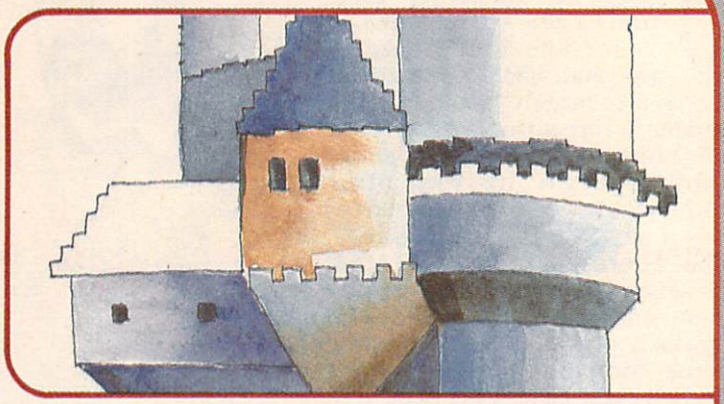
Atari announced two new hardware lines at the show. At the low end is the XE series of 8-bit machines (with 6502C microprocessors). A 64K version with 256 colors (and 320 x 192 resolution) will sell for approximately $120. For $150, there's a music version with 8 voices; $200 gets you an expanded-memory machine with 128K; and $400 buys a 64K portable with built-in 5-inch monitor and 3-1/2 inch disk drive.

Atari's second new line is its ST series of 16-bit machines (with the 68000 microprocessor). The ST sports Digital Research's GEM operating system—a Macintosh-lookalike desktop environment, complete with mouse and icons. The 128K version (sans disk drive) of Tramiel's ST "Jackintosh" will sell for about $400; a 256K version will be available for $500; and a 512K version for $600. Each ST machine is capable of displaying 512 colors, a 640 x 400 pixel resolution in monochrome mode, and contains a built-in MIDI interface to provide digital control of musical instruments. A 3-1/2 inch disk drive and a color monitor will be sold for about $200 each. The new Atari crew was also boasting of a forthcoming 15-megabyte hard disk drive to sell for $400!

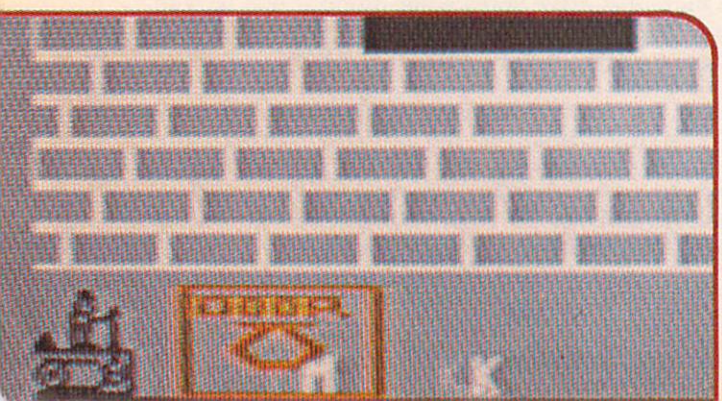
It's clear Mr. Tramiel has caught a touch of "Iacocca Fever." In trying to resurrect the new Atari Corporation, he is squarely taking aim at "The Big 3" with a full, aggressively-priced product line that—if delivered as promised—could set off a price/performance war of mammoth proportions.

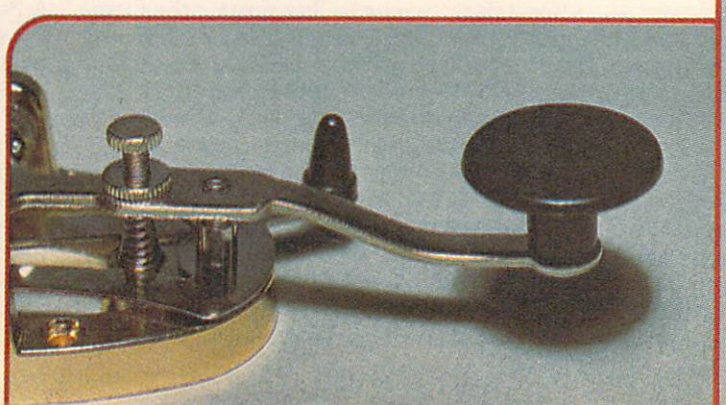


HOME COMPUTER™ magazine



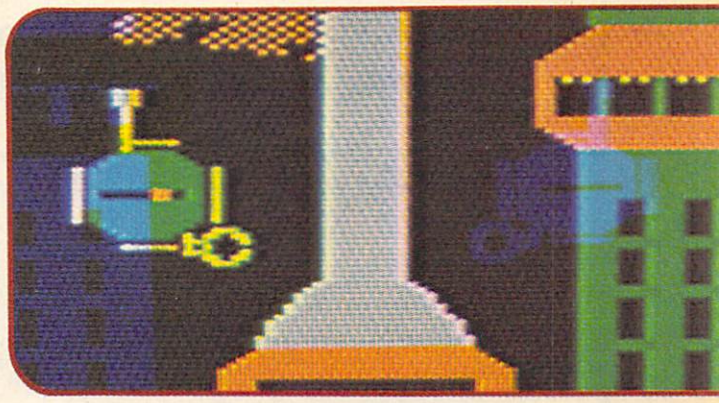
FEATURES





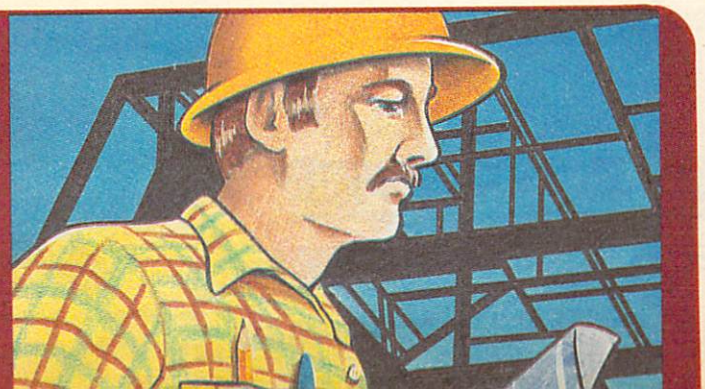
CONTENTS

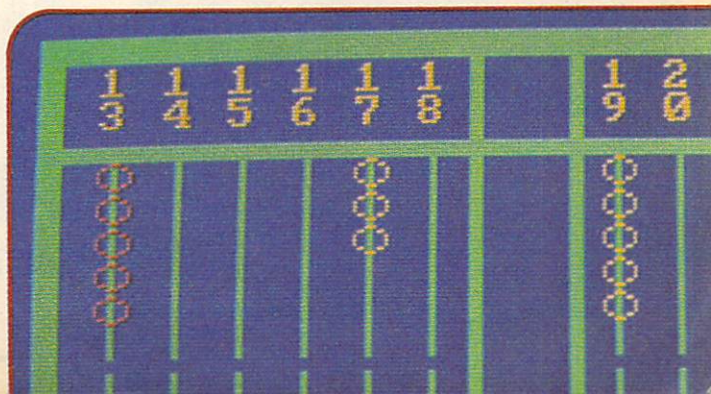
VOLUME 5 NUMBER 1



PRODUCT REVIEWS

DEPARTMENTS





HISTORICAL NOTE
99'er Magazine (founded in December, 1980) was the forerunner of Home Computer Magazine.

THE BEST OF 99'er™

SUPER CLOSE-OUT SPECIAL for TI-99/4A USERS

A Giant Home Computer Compendium™ for the Texas Instruments 99/4A

The largest, most comprehensive collection of programs and articles ever assembled for the TI Home Computer

VOLUME 1

- Over 200 thoroughly tested key-in-and-RUN programs and sub-programs typeset in a grid format for maximum clarity.
- Programming instruction in 4 languages—learn to use BASIC, Extended BASIC, LOGO and Assembly Language—for everything from record keeping and money management to arcade-quality action games.
- A selection of sensational game software featuring full-color graphics, animation, and sound effects.
- Beyond the owner's manual—tips and techniques for getting the most out of your computer system.
- Computer-Assisted Instruction—The home computer becomes your private tutor.
- Page after page of innovative applications—transforming your computer into a home productivity center.

Regular (Pre-Close-out) Prices:

Best Of 99'er (Book alone)	$19.95	+ $3.00 shipping
Best Of 99'er (Tape Set alone)	$35.00	+ $2.50 shipping

USE BIND-IN CARD AT CENTER OF MAGAZINE

SPECIAL OFFER Buy the Tape Set for ONLY $35. And Get the Book FREE + FREE SHIPPING

FREE BONUS WHILE SUPPLIES LAST

ORDER THE BOOK & TAPE PACKAGE AND RECEIVE A Simon's Saucer AND A 99'er Programmer's Guide **ABSOLUTELY FREE!** This Additional $18 gift IS YOURS IF YOU ACT TODAY!

PLUS

FREE BONUS WHILE SUPPLIES LAST

THE BEST OF 99'er ON TAPE

A Choice Selection of 37 Full-length Programs On 5 Quality Cassette Tapes

- ★ Save Typing Time and Frustrating Key-in Errors.
- ★ Own the Most Comprehensive Software Library for the TI-99/4A
- ★ Enjoy Hundreds of Hours of Exciting Computer Activity.

TO ORDER—USE BIND-IN CARD AT CENTER OF MAGAZINE

People Who Know the Home Computer Best™

HOME COMPUTER™ magazine

AN EMERALD VALLEY BOOK

IF CENTER BIND-IN ORDER CARD IS MISSING— MAIL WRITTEN ORDER TO:
P.O. Box 70288 Eugene, OR 97401
FOR CREDIT CARD ORDERS CALL 1-(800)-828-2212

Letters to the Editor

Keeping Your Head Clean

Dear Sir:

I read several months ago that one manufacturer of disk head-cleaning kits had a program that caused the disk head to stay in contact with the disk cleaning pad for 30 seconds, but I was not able to find one of the kits. Since then I have been trying to figure out a program that accomplished this and believe that the following program does just that.

```
10 ON ERROR 30
20 RUN "DSK1.LOAD"
30 ON ERROR 40
40 RETURN
```

This program will run the disk drive for 30 seconds, although the head may not be in contact with the cleaning pad for that long. I hope this information may be of some use. I like your new format. Keep up the good work.

Steve Lisonbee
Orem, UT 84057

There are many ways, Steve, to get the disk to turn for 30 seconds while cleaning the heads. Probably the simplest way is to try to load something from that particular drive and count the number of seconds that the drive turns, subtract that from 30, and repeat the load procedure until you have actually reached the 30 seconds. It is easy to tell when the head is in contact with the cleaning disk because you will hear a definite rasping sound as the rough surface passes over the disk head. We have noted that in cases where a disk drive is used extensively and is not cleaned regularly, the disk drive cleaning kit must be used much longer than just 30 seconds to remove the deposits from the head. This seems to be a trial and error situation. If your drive is actually producing errors, then we suggest that you run the head cleaner for 30 seconds, and see whether the errors have gone away. If not, run it for another 30 seconds and repeat this until the errors have cleared up. If after a collective period of five minutes the errors persist, have the drive checked by a service center.

Companion Review Reviewed

Dear Sir:

I would certainly like to commend you upon several aspects of the Companion word processor review. I was indeed pleased to see Companion so favorably reviewed in the August [Vol. 4, No. 3] issue of *Home Computer Magazine* and I congratulate the reviewer on accurately portraying the philosophy of Companion's design.

However, I would certainly have preferred that she do a more careful reading of the documentation before launching into the review. For example, Companion does have a "handy reference card." It happens to be two of the four sample text files included on the Companion diskette. We felt that this was the best possible way to provide such reference material, because the user may easily reproduce it in unlimited quantities and also edit it to personal preference.

Regarding the operation of the [FCTN][3] key, it is clearly stated both in the manual and in the reference sheets that its purpose is to toggle the visibility of the typed spaces. It is, of course, its own "oops" key.

It was with some disappointment that we saw TI-Writer called the "de facto standard" [word processor in the 99/4A community], and it came as quite a shock to see TI described as "generous" in the middle of a review of Companion. I will leave it to your readers to decide whether this editorial comment was either appropriate or accurate.

Allan Swett
Intelpro
Brossard, Quebec, Canada

Thank you for your comments, Allan. We still feel, however, that the review was accurate and a fair assessment of your product's strengths and weaknesses. As for our statement that "TI has generously provided the 'hooks' for third-party add-on products," we feel that it was both appropriate and prescient. See our review of a third-party spelling checker for TI-Writer in Vol. 4, No. 5.

Word From the Board Room

Dear Sir:

I recently purchased a copy of the Vol. 4, No. 3 issue of *Home Computer Magazine* and was very impressed. I think the editorial style is easy to read, but unlike many other magazines of its kind, it wasn't filled with "computer fluff." As the in-house list manager for Boardroom Reports, I get to see a lot of computer magazines (since most of the "non-technical" ones use our lists) and yours is among the best.

Brian Kurtz
Board Room Reports
New York, NY 10036

It is nice to get an expert opinion, Brian. Thank you for taking time to write.

Innovation Is Not Without Problems

Dear Sir:

I would first of all like to compliment you on *HCM*. I have an Apple *IIc* and a TI-99/4A and *HCM* is unique in the usefulness of the material you present. I especially appreciate the availability of low-cost software in the form of your ON DISK and ON TAPE service. I also appreciate some of the innovative approaches you take to marketing, formatting, and editorial.

However, I have a concern. I have received issues at a slower rate than one per month since subscribing to *HCM*. While I can appreciate the chaos and re-planning that was undoubtedly necessitated by Texas Instruments' discontinuation of the 99/4A last year, and delays in publishing due to some of the revisions of policy you have recently undertaken, I would hope that your issues can be regularly counted on in the future.

Fred K. Rehl
Sedalia, OH 43151

As announced in our last issue, Fred, you can count on HCM being published ten times each calendar year—with approximately 5 to 6 weeks between issues. Both existing and new subscriptions are being fulfilled on an issue basis (instead of by months) so that everyone will receive the correct number of issues they are entitled to. This reduction to a more realistic frequency was necessitated by the comprehensive software content of each HCM issue.

Where Did All the Memory Go, Junior?

Dear Sir:

I recently acquired the IBM Writing Assistant word processor for my IBM PCjr. While it works fine, I find that I can only store about 1600 words in the "working copy" before the program memory is full. This is only about 12K bytes of RAM. I've got 128K RAM installed in my Junior, so I expected to be able to write a much larger document. What do you think?

James McCloskey
Reidsville, NC 27320

The IBM Writing Assistant word processor is a very large program, James. It was originally designed to run on the IBM PC with a minimum of 256K of RAM. In that configuration, it allows space for about 140K of storage for the "working copy." If you need more space to work with larger documents, you have two options: (1) select a word processor that takes up less space and memory, or (2) buy a memory expansion unit (from IBM or third-party firms such as Tecmar, Legacy, Microsoft, etc.) that will be compatible with the IBM Writing Assistant package.

Junior's Math & Memory Questions

Dear Sir:

I have two questions about the IBM PCjr. My first question has to do with mathematical accuracy. After reading Microcomputer Accuracy (Vol. 4, No. 1) I decided to try a few of the tests mentioned in the article. All but one yielded the right results. I typed in:

```
10 A# = 4*ATN(1)
20 PRINT A#
```

The answer I got was 3.141592979431152, nowhere near pi after the seventh digit. Amazed at how far off it was, I tried:

```
10 A# = 4*ATN(1.000000000)
20 PRINT A#
```

This yielded the correct answer of 3.141592653589793. Why does it do this? Does this mean if I want accuracy past seven digits I must put zeros at the end of each operation?

My last question is about additional memory. In your review of Tecmar's jr Captain you state that "Unfortunately, the additional memory doesn't mean a PC application that was memory-bound on the Junior will now work just fine." This is a seeming contradiction to what is said in "Home Computer Product News," which says, "The IBM PCjr 128KB Memory Expansion Attachment adds 131,072 characters of user memory to a PCjr, and can be used to run thousands of IBM PC programs." Which is correct?

Thank you for your time.

Sean Hillyard
Oakhurst, CA 93644

Continued

There's a paradoxical situation concerning memory expansion for the Junior: The Tecmar Junior Captain came with software that did not allow the use of the memory expansion with larger PC packages that needed contiguous memory space (it may have been corrected in a later version of Tecmar's software, but we're not aware of it). The IBM PCjr 128K byte memory expansion comes with software to configure the Junior so that it looks like an IBM PC with contiguous memory, thus allowing most PC programs to operate correctly.

Your discovery about accuracy, Sean, boils down to the election of "double precision." Unless you specifically declare double precision numbers—which you did in the second case—your Junior will default to single precision. An easier way to declare double precision in your formula would be:

A# = 4*ATN(1D0)

HCM Helps Make Buy Decision

Dear Sir:

I have enjoyed your magazine immensely since it was *99'er Home Computer*. It was a great magazine then and is an even better magazine now. I have been craving more computing power and better software lately. I think that (from your informative coverage of the IBM, Commodore, Apple, and TI machines) a Commodore 64 would fill the slot nicely. It is a good machine at a good price. I was wondering if another Commodore, the Plus/4, is compatible with the C-64. And thanks again for the best computer magazine on the market.

Ashley Jacobs
Caracas, Venezuela

At the present, Ashley, the Commodore Plus/4 is not software compatible with the C-64. There were rumors circulating, however, that Commodore would soon introduce an add-on to make the two machines compatible. In light of the just-announced C-128 machine that is C-64 compatible (see On Screen in this issue), it doesn't appear to be likely. We're glad that you find Home Computer Magazine to be beneficial in aiding you to make a wise purchasing decision.

TI Game Cartridge Trick

Dear Sir:

I found a trick for most of the games made by Texas Instruments for the 99/4A. This trick allows one to change the starting level. To do this, simply select the game (after inserting the cartridge) and when the title screen of the game appears, quickly type (holding down the shift key) 838. I know this works on the games Moonmine, Alpiner, Munchman, Hopper, and Munchmobile, but those are only the ones I know of, so try and see what other games this works for.

David L. Whitlock
Houston, TX

You're right, David. This trick seems to work on most of the TI game cartridges (the ones actually produced by Texas Instruments). This trick can be used to change the level of difficulty and some other conditions of a game at the very beginning. This is ideal for advanced players who wish to skip over the simple lower levels of an arcade game. It is also useful to those less skilled who have never been able to reach the upper levels of these arcade games, and want to satisfy their curiosity.

Texas Decisions Italian-Style

Dear Sir:

About a year ago (when I bought my TI-99/4A) I was greatly surprised at seeing how many peripherals and accessories this computer can have. But just out of production, my hopes to have these peripherals soon vanished, and I haven't made up my mind yet whether to buy another computer or not. That is the reason why I ask you if in America someone is still selling the Extended BASIC module, the Editor/Assembler module, etc. at possibly a reasonable price, in case you have an address.

Principally, I'm not asking you where to buy my accessories, but if it is worthwhile. I would not spend my money on a computer which has no possibility to evolve.

Add the probability that I can't have my computer repaired if it breaks because there are no TI assistance centers in my city. Draw your conclusions and write them (and the addresses) to me.

I wait hopefully for your answer.

Pietro Parrotta
Palermo, Italy

Pietro, it is difficult to say what you should do because you are located in Italy. Peripherals and software such as Extended BASIC, etc. are available in the United States as is support from the TI service centers. There are (or were) two TI facilities in Italy—at Aversa and Reiti. TI-99/4A computers bound for Europe used to go to Reiti for final assembly, so personnel at that facility may be able to guide you in your repair concerns. If you haven't already made a large investment in TI peripherals and software, we suggest you seriously consider purchasing one of the other computer systems we cover in this magazine.

Printer Switched to IBM Confirmation

Dear Sir:

I have been reading *99'er* a long time and was pleased in its change of format to *HCM* after I purchased an IBM PC portable. I would like to know if you could help me in properly using my Axiom GP-100 TI II (which you reviewed in *HCM* Vol. 4, No. 4) with my IBM. I had no problems using it with my TI. I purchased a parallel interface cable for my IBM and got the printer to work, but when using Wordstar I cannot set line spacing, even with the [Ctrl] KS command, although the dot commands work in setting page length. Also, is there any software that will allow my printer to perform graphic screen dumps (the [PrtSc] key doesn't work), or is there any IBM software that will give this printer any new fonts. I would greatly appreciate your assistance.

Sam Ousta
St. Petersburg, FL 33715

Your information concerning the compatibility and incompatibility of the Axiom and IBM machines is interesting, Sam. That particular Axiom version was produced for the TI-99/4A only, and the compatibility you've noticed stems from similar methods of handling the text of both IBM and TI printer routines. However, many differences exist—line spacing and dot-addressable graphics are just two of these. The Axiom's line spacing cannot be modified through software as it can with many IBM-compatible printers, which explains your Wordstar problem. The [PrtSc] function, which is accessible through the GRAPHICS command in DOS (see this issue's IBM "Home Computer Tech Note" for details), is not compatible with the unique graphic methods used by the Axiom. You can access these unique graphic commands (as detailed on pages 24-26 of the Axiom User Guide) from Microsoft BASIC, however, if you use the LPRINT command in place of the OPEN #1:"Axiom" and PRINT #1 in TI BASIC. Here's a one-line IBM program that does the same thing as the demo program on page 25 of the Axiom manual:

```
100 LPRINT CHR$(8):FOR I=1 TO 4:
    LPRINT CHR$(129); CHR$(137);
    CHR$(145); CHR$(161); CHR$(255):
    NEXT I: LPRINT CHR$(15)
```

We hope this gives you some insight into how you might make the most of using the Axiom printer with your IBM PC portable.

A Worm in Apple 3-D Graphics?

Dear Sir:

I know the frustration that B. Matthies spoke of in her letter to you (Vol. 4, No. 4) concerning the 3-D graphics program. I experienced some of the same problems that she outlined. However, I solved the load problem by first making a DOS 3.3 copy of all related 3-D programs and then loading the HELLO program onto the disk. My major frustration was in not being able to reload a saved, newly-created 3-D object. After reviewing the Applesoft 3-D program several times and experimenting with changes, I finally changed "size" in lines 1140-1160 to a value of 100. Also, I deleted the "%" symbol in lines 1760, 1770, and 1920. The program now runs smoothly and it is quite enjoyable to manipulate the axial rotation of various objects.

The latest format of *HCM* (Vol. 4, No. 4) arrived today and it appears that you may have finally found the magazine style and format that you have been searching for. *HCM* readers should be pleased with the quality of the magazine.

Paul E. Pennebaker
Slidell, LA 70461

You've done the right thing, Paul, moving the 3-D Graphics system to its own disk—it will give you plenty of room on the disk to save objects. The problem you describe is one that only readers who typed in the program (rather than loaded it from ON DISK) might have experienced. Chances are you didn't notice that not only do you add the listing on the right side of page 41 from Home Computer Magazine, Vol. 4, No. 2, to the listing from Vol. 4, No. 1, but you also must change line 770 of that listing. The line should read:

770 ON OBJ GOSUB 1190,1320,1480,1900:
REINITIALIZE ARRAY

When this change is made, using integer arrays (the %) in lines 1760, 1770, and 1920, these lines are all correct as printed. Because the SIZE variable is set to 100 in line 1130, your change of that variable in other parts of the program appears to be unnecessary.

IBM Library Software Needed

Dear Sir:

I currently administer a small law library for the Federal Energy Regulatory Commission in Washington, DC. Several of the library's manual filing and tracking systems appear to be candidates for automation.

The library will soon acquire an IBM Personal Computer. My hope is that I can locate software that will permit me to store and track data relevant to interlibrary loans, congressional legislation, and depository materials.

In my opinion, an inventory control program of some type would probably provide the capabilities needed, particularly one designed for library applications.

If you have any suggestions regarding where I might obtain software suitable for the applications I described above, I would appreciate receiving them.

Barbara Giersch
Washington, DC 20426

We are aware of several software packages for the PC, Barbara, that specifically reference (no pun intended) maintaining library files. These include: MDBSIII (Micro Data Base, Inc.), The Data Factory (Micro Lab), Fasfile (Gryphon Systems), and Citation (Eagle Enterprises). For an overview of these (and other) packages, we suggest that your consult two directories: IBM Personal Computer & XT Software Guide (from MicroInformation Publishing, 15420 Eagle Creek, Prior Lake, MN 55372), and PC Clearinghouse Software Directory (from PC Clearinghouse, 11781 Lee Jackson Hwy, Fairfax, VA 22033).

PCjr Serial Port—Not For Printers

Dear Sir:

This letter was written on a PCjr using PC-Write and printed on a TI-99/4 Impact Printer. The printer is hooked up to the extended model PCjr serial (S) jack using the 1 through 8 and 20 leads. Pins 2 and 3 are reversed between the two units. This only works for OPEN #n and PRINT #n output. PRN from DOS, LLIST from BASIC and LPT1: just hang the system, as does an attempt to use PR from PC-Write.

How about a "how to do it" article on this? It should include enough data to set up all functions to work correctly. Perhaps a new ROM would be needed in the printer.

Ed Quenzer
Eastham, MA 02642

One of the problems, Ed, with using the serial port on the IBM systems is that the printer is expected—by both the operating system and Microsoft BASIC—to be connected at the parallel port which, of course, happens to be an extra-cost item. We are looking into this problem and should have your request for a how-to-do-it article filled in a forthcoming issue. One way to solve the problem for now is to spend the money for the PCjr parallel interface and parallel interface cable, remove the serial interface card from the TI-99/4 impact printer, and connect everything together.

Making C-64 Listings Easier to Read

Dear Sir:

Typewritten letters are not the only things which are easier to read when double-spaced. Program listings are easier to read too if they are double-spaced. I thought that your readers might like to know one way that this can be done.

I have a Commodore 64 and Mannesmann Talley MT160L printer with a CARD/PRINT interface. The interface provides several alternative "modes" for printing out a listing, depending on the "secondary address" given in the OPEN statement. For instance, using a secondary address of 2 in OPEN 4,4,2:CMD4:LIST puts the printer and interface in a mode which prints out a program in upper case only and provides a "line feed." A line feed makes the printer advance the paper by one line with each carriage return (each time a new line is started). The statement OPEN 4,4,6:CMD4:LIST will result in the program being listed in upper and lower case, with a line feed. If the printer has also been set up to provide a line feed with each carriage return, then either of these listing modes will give double-spaced listings. A double-spaced listing is easier to read and provides space for inserting notes when editing a program.

For my combination of printer and interface, and perhaps for others, there is an additional advantage. The CARD/PRINT interface translates any control characters to readable abbreviations in the printer version of the listing. This can result in some program lines ending up with over 80 characters. For some reason, with single-spaced listings, this interferes with the line-feed, and overprinting of the preceding line results. Double-spacing prevents this overprinting.

Although I am not familiar with other interfaces, I would guess that they must provide various modes much like the CARD/PRINT interface. Most, if not all, printers enable the user to turn on or off their line-feed function. So, I am reasonably sure that most readers would be able to use this technique to obtain double-spaced listings.

Jack Ryan
El Dorado, AR 71730

Those are some good ideas, Jack, that we're sure quite a few Commodore users will appreciate. Thanks.

Junior Group Likes HCM

Dear Sir:

By the time you are able to publish news about the Las Vegas IBM PCjr Users Group, we will have quite a large group. As a sub-group of a PC club, we were dissatisfied enough to start from scratch, so that all users of the PCjr could be fairly represented.

Since I purchased my first 99/4A in October, 1982, I have depended on your magazine as my major source of computing information. Now that Ziff-Davis and Compute! have stopped publishing PCjr-specific magazines (really stupid), I depend on you for the most complete coverage of the PCjr. You have an objectivity that is missing in other magazines and it allows me to place the PCjr in perspective with the rest of the market. Your ON DISK option is unrivaled, your "Tech Notes" are always used, and your method of identifying each computer to related articles and reviews beats any multi-computer magazine I've read. You have good program listings, and "Industry Watch" holds my interest because of past TI mishaps. I think it is very important for me to tell every member of the PCjr group about *Home Computer Magazine* for their own good.

I would like to ask you to include as much PCjr-related advertising in future issues of *Home Computer Digest* [our sister publication, mailed free to subscribers of this magazine] as you can. With the recent changes in PCjr-specific magazines, I feel that many mail-order suppliers of PCjr products will be looking for a good publication to advertise in.

I would like to tell you that your first issue of *Home Computer Magazine* was a major force of motivation that resulted in my purchase of an IBM PCjr.

Danny Duran
Las Vegas, NV 89115

Danny, it's great to hear from a PCjr users group. Most of the IBM groups that we have heard from have been primarily PC-only in interest. How many other PCjr-specific groups are forming out there? Let us know. Write in to the Group Grapevine editor and tell us about your activities and how we can better serve you.

C-64 Reset Button Problem

Dear Sir:

I have recently installed the reset switch that you featured in your C-64 "Tech Note" (Vol. 4, No. 4). I still have yet to figure out how it can save a program. If I load a program by POKEing (44,64) it will stay in memory, but there is no way to transfer it back to (44,8) to RUN it.

I have a Commodore 64. Could you please give me some advice on this?

Thank you.
S. Schoenfeldt
East Long Meadow, MA 01028

Note: See answer after the following question.

Another C-64 Button Reset

Dear Sir:

Last week I turned bold and decided to install a reset switch in my C-64. I am referring to the article in your magazine (*HCM* Vol. 4, No. 4, page 81).

I have had previous kit-building experience and I used my mini-soldering iron. I was told

Continued

Letters to the Editor CONTINUED

to add (by a friend) a 50-100 ohm 1/4 watt resistor on one leg, which I did. It does reset the machine—back to the logo and all memory is lost! LIST and RUN are both dead. I tried it without the 68 ohm resistor and RAM was also wiped out.

I used the Simon's Basic cartridge then. I hit the reset, then the logo for Simon. I typed OLD and <RETURN> then LISTed it and it came back.

Can this be done without Simon's Basic?

I would like to reset and RUN and/or LIST but . . .

Chet Mentz
Beverly Hills, FL 32665

The purpose of the referred-to "Tech Note" was to totally reset the computer (including all BASIC pointers), thus saving wear and tear on the system. We never intended that this modification be taken as a "super" [RUN-STOP][RESTORE]. We're sorry for the misunderstanding. Further software is necessary to restore any of the pointers. If any readers have a simple method, please write us.

Bermuda Race Lost to IBM World

Dear Sir:

Thank you for sending us a copy of *Home Computer Magazine* containing the review of our Bermuda Race program. There is one correction I wish to make. The IBM version will not be marketed.

We are grateful for the exposure you have given our product. Good luck with your new "no advertising" policy. While I read many magazines specifically for the advertising, I go to the editorials and reviews for "the facts."

Thanks again.

Dave Stitt
Howard W. Sams & Co., Inc.
Indianapolis, IN 46206

Thanks for the update, Dave.

User Functions—No Thanks

Dear Sir:

The article entitled, "Programming: The Name of the Game" that appeared in the Vol. 4, No. 3 issue of *HCM* suggested the DEF program statement to define frequently-used functions in one's game program. However, I think it should be pointed out that functions defined in this manner take almost twice as long to be executed as do the formulae themselves. Consider the following demonstration for the TI-99/4A:

```
100 CALL CLEAR
110 DEF YY=1+2+3+4
120 FOR A=1 TO 32
130 CALL HCHAR(YY,A,42)
140 NEXT A
150 CALL CLEAR
160 FOR A=1 TO 32
170 CALL HCHAR(1+2+3+4,A,42)
180 NEXT A
```

As you can see, since line 130 uses the predefined function, the first loop is executed much more slowly than the loop using just the same formula.

I'm not trying to suggest that the use of predefined functions be abolished, but only that in situations where speed is important should their use be avoided. I hope that this information has been helpful and that it will help to solve some programming headaches.

Richard Solomon
Freehold, NJ 07728

You are absolutely right, Richard. The use of user-defined functions will indeed slow the execution of the TI BASIC programs. User-defined functions should not be used in time-critical loops or often-used subroutines that may affect the "playability" of the game.

Country Club on Cassette

Dear Sir:

First, let me congratulate you on your new format without outside advertising. As a former advertiser in *Home Computer Magazine*, I was skeptical, but after seeing the first issue in the new format, I am convinced *HCM* is setting new standards in computer journalism.

I was also impressed with the thoroughness of Tom Green's review of Country Club in Vol. 4, No. 4. He obviously spent a great deal of time examining the package. There was, however, one inaccurate point in the otherwise-exceptional review. The review indicated that memory expansion is needed to run Country Club. This is true only in the disk version. Making the game compatible with an unexpanded system was our major goal in development. I would appreciate it if you could mention this in a future issue of *HCM*.

As a TI owner expanding into the Apple and IBM markets, I want to commend you for the unequaled quality and depth of your magazine. I look forward to the next issue.

Michael A. Kidd
User-Happy Simulations
Barrington, IL 60010

Thank you for your comments, Michael. And for readers who wish to purchase Country Club on cassette for the 99/4A, we understand that the suggested retail price is $19.95.

Back Issues Appreciated

Dear Sir:

I have just finished reading all the back issues of *99'er Home Computer Magazine* I recently bought. Even though I had purchased the *Best of 99'er* several months earlier, I found that the additional back issues were packed with still more valuable information, and recommend to owners of the TI-99/4 and TI-99/4A to buy the back issues if they haven't already done so.

A problem I have come up against is in Microsoft's Multiplan. I have not found a way to handle dates in the formulas using MM/DD/YY format. The only reference to this that I could find was the VALUE(T) function, but I haven't had any success subtracting a date in one month from another to get a value to use in other formulas, such as in cost/day calculations.

Also, it would be very handy if at the end of the year you would publish a pull-out index of past articles and programs categorized by subject matter.

Donald P. Mefford
Peebles, OH 45660

Yes, Donald, we're working on the index. We appreciate your comments about the older 99'er back issues, and have taken it one step farther. Readers are now able to purchase not only the back issues of the magazine, but also the software contained in each issue recorded on either a cassette tape or a floppy disk.

Unfortunately, Don, there is no simple way in Multiplan to handle the date calculations you mentioned. In fact, manipulating dates to attain cost/day statistics is tricky in any computer language. The VALUE(T) function only works on strictly numeric strings, and the slashes in MM/DD/YY format will cause problems. If you remove the slashes and rearrange the date in a YYMMDD format, you might be able to use the MODULA function to create functions to get the information you desire, but it won't be simple in any case. If any of our readers have any algorithms for these sorts of manipulations (whether in Multiplan or BASIC) please let us know.

Decided to Buy Apple Clone

Dear Sir:

First of all, after complaining about delays of receiving *HCM* on a not-so-regular basis, I'm pleased to report all seems well now.

I'm extremely pleased with the magazine as it's the only one I know of that deals in all formats.

I should mention that I started off buying *99'er* when I got my TI machine about a year-and-a-half ago. Just recently, I upgraded my computer system when I became concerned about getting accessories. A friend of mine recently bought an IBM PC and so had no further need for his Apple clone—namely a Zeus 2001 system.

My system now includes two ATI drives, the Zeus computer, CP/M, Z-80, and an 80-column card, as well as a Gemini 10X printer and Arcomp amber monitor (far superior to green and more legible).

I am currently in the process of keying in programs from *HCM* since you changed from *99'er HCM* to *HCM*. By the time I'm finished, I should have a good backlog of programs.

I hope your standards continue, as to date they are excellent. As well, I'd be interested in hearing from other Zeus owners and in particular, those who have Apples and Apple clones overseas, on how they use their systems or run their clubs.

Keep up the good work and delivery, and you'll have an *HCM* subscriber forever.

Philip Elliott
Kincardine, Ontario, Canada

If your machine is a true clone, Philip, it should be able to read the Apple DOS 3.3 and Apple ProDOS disk formats that all Apple II Family programs in HCM run under. Here at "Programming Central" we're very much interested in hearing about how our programs are

being used on both Apple and IBM clones and we'd appreciate it if our readers would keep us closely informed.

Stuck in Pyramids of Doom

Dear Sir:

My family and I are being driven crazy by Pyramids of Doom. This an Adventure program and we are stuck in the middle of it. If any of your readers have completed this game, can they give us some assistance?

I am very much impressed with your magazine—keep it up!

Peter J. Stephens
Port Jefferson, NY 11777

We don't think anybody has ever returned from the Pyramids of Doom. Wait—one fellow did come back, but he never stopped babbling that "dead mummies don't wear plaid!" We think that may be an important clue . . .

TI Disk Manager II Sought

Dear Sir:

After reading the article "2 for TI" in Vol. 4, No. 4, I have been trying to locate a Disk Manager II—with no luck. Can you suggest a purchasing source?

James A. Canter
Englewood, OH 45322

James, we suggest you check all the mail-order sources in our companion publication, Home Computer Digest [mailed free to our subscribers]. You might also check with an active TI user group in your area, such as: Cin-Day Users Group, Box 519, West Chester, OH 45069

Stadium Jumping With the *IIc*

Dear Sir:

A while ago I purchased a copy of the Vol. 4, No. 4 issue of *Home Computer Magazine*. It was the first that I had seen. It seemed to be what I have been looking for, so I sent in a subscription order.

I typed in a number of the short program listings in this issue for the Apple *II* family and found that they ran beautifully on my *IIc*. That gave me the incentive for trying the "Stadium Jumping" program, but when I tried to run it on the *IIc*, it wouldn't run. I got a "Syntax error in 220." Although I have checked my listing with your published listing many times, I can find no error.

I had a program on disk that would not run properly on my *IIc* but ran with no difficulty on a friend's *IIe*.

In the future, would it be possible to provide listings for ProDOS or at least the changes required for ProDOS operation? I understand that most of the newer Apple computers now use ProDOS, and I'm sure that you would make a lot of us very happy.

This also prompts me to wonder if the programs that you have ON DISK are fully compatible for the *IIc*. If so, I am sure that I will purchase some of them, but I hesitate until I know.

One final note: I believe that there is a typo in your "Stadium Jumping" listing for the Apple *II* in line 740. I believe that a comma is missing in the entry shown as 2626, the next to the last data entry for that line. I may be mistaken because my attempts at running the program have not gotten that far.

John H. Lincoln
Seattle, WA 98117

John, all of our software is presently set up for ProDOS for the Apple machines and the ON DISK media is definitely compatible with the Apple IIc. We began that with Vol. 4, No. 3, and as of Vol. 4, No. 4 all ON DISK media comes with an Apple-licensed ProDOS package—including startup and formatting capabilities. You will find that all ON DISK products work on the Apple IIc. The listing of "Stadium Jumping" is correct as published, and the problem you refer to is either a typing error in line 220 or in one of the DATA statements containing the machine language routines referenced by that line (lines 250-330). The DATA statement in line 740 is correct as published.

TI Schematics Desired

Dear Sir:

I would like to know where I can get the electrical diagrams for the TI-99/4A. If any of your readers know this, I would like to hear from them.

Thank you for a fine magazine.

Reynaldo Rivera
San Juan, Puerto Rico 00926

The easiest way to get the technical information that you are looking for is from a local TI user group. Currently, however, we know of none that are still active in Puerto Rico. The closest group listed is the Miami 99/4A Users Group, ATTN: G. Guibor, P. O. Box 650955, Dept. MUG, Miami, Florida 33265-0955, (305) 522-1984. If any readers know of an active users group in Puerto Rico, please let us know. If the "group grapevine" can't get you the diagrams you need, we suggest that you write us again with a more specific schematic-related question(s) that we will try to answer for you.

Answer to *LOGO* Word List Question

Dear Sir:

In the Vol. 4, No. 3 issue of *HCM*, Gene Thomas asked for a LOGO procedure to insert a word at a specified point in a list. A short routine to do this follows:

```
TO PUTWORD :WORD :LIST :POS
TEST :POS > 0
IFF STOP
MAKE "POS :POS - 1
MAKE "F [ ]
REPEAT :POS [MAKE "F SENTENCE :F
FIRST :LIST MAKE "LIST BUTFIRST
:LIST ]
MAKE "F SENTENCE :F :WORD
MAKE "F SENTENCE :F :LIST
MAKE "LIST :F
OUTPUT :LIST
END
```

In this routine, "WORD is the word you wish inserted, :LIST is the list you want the word inserted into, and :POS is the position you want the word to take in the list. If a value of zero is entered for the position, the routine will not do anything at all. Also, if a position is greater than that of the last word in the list specified, the new word becomes the last one in the list (e.g., PUTWORD "A [Z X C] 9 yields a computer response of [Z X C A]).

Anyway, thanks for a terrific magazine. You can count on my subscription as long as you're in business.

Loring Rose
Pantego, NC 27860

Thanks, Loring. Now we know how to put a word in edgewise, or at the front, or at the back, or . . .

A Vote of Confidence

Dear Sir:

I just renewed my subscription to *Home Computer Magazine* for two years. I was going to mail the renewal and enclose this letter, but instead telephoned the renewal to your toll free number.

At first when I heard the magazine would no longer be for the TI-99/4A exclusively, I was disturbed and disappointed.

The reason I'm writing is to tell you how pleased I am with *Home Computer Magazine* in its new format and especially the separate advertising supplement [*Home Computer Digest*]. I'd be willing to bet other magazines will follow your leadership. Very innovative, practical, and courageous.

Best wishes for continued success and growing circulation numbers.

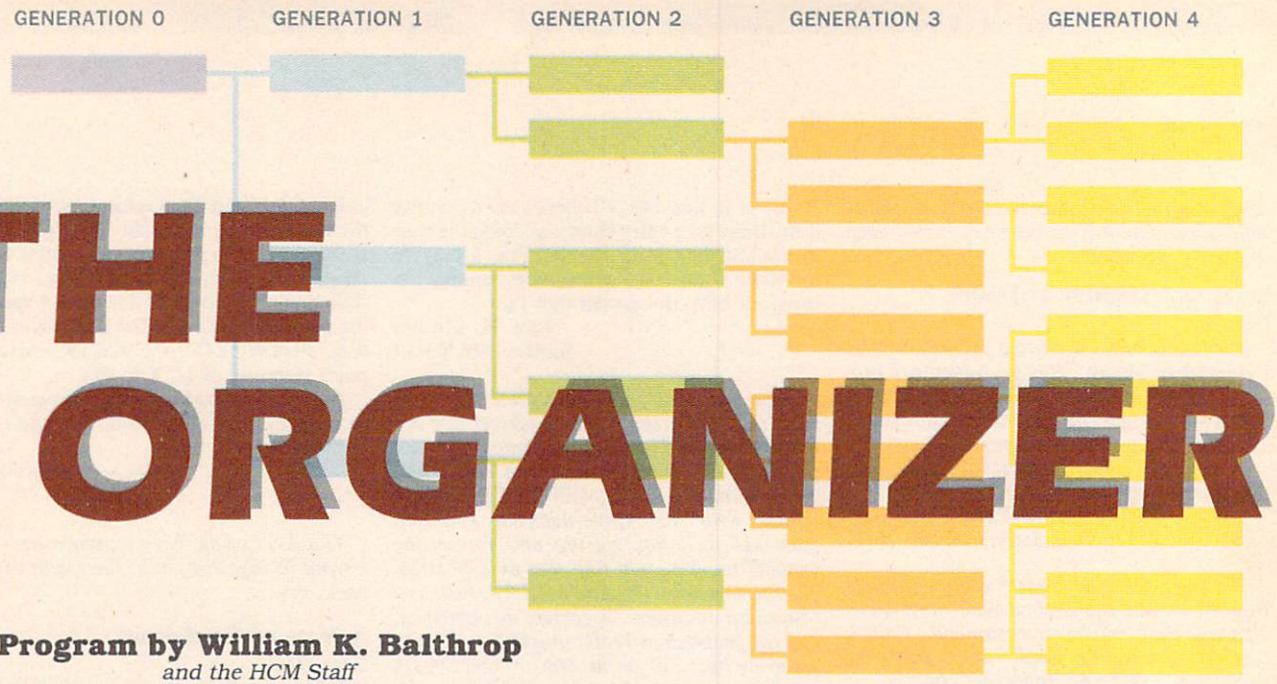
Robert H. Howry
Los Angeles, CA 90025

Thank you for your comments, Robert. We're able to make this new format "work" because of the many tens of thousands of understanding subscribers like you.

HCM

Special Announcement:

Home Computer Magazine is looking for "One-Liners." If you have written a 1-line program in any language that is available on the computers we cover, send it in addressed to *Letters to the Editor*. It may win a $50 prize! By press-time, we had not yet received enough worthy candidates for publication in *this* issue, so keep those entries coming. On page 34 of our last issue, Vol. 4, No. 5, we published the top 4 entries—one for each brand of machine—and awarded $50 to the best one-liner of the four. It can happen to you, too!



THE ORGANIZER

Program by William K. Balthrop
and the HCM Staff
with text by Wayne Koberstein
HCM Staff

Too many thoughts to keep in mind? This program will not only store your thoughts for you—but will keep them organized as well!

I've just got to get organized! How many times have you said that to yourself? Each time you take on a new project or tackle a new assignment, it's a whole new ballgame, with a new set of rules and contrary elements that seem to defy organization. You know that if you just stopped and planned it all out, the actual job would be so much easier—but at the start, it seems less trouble to jump right in and wing it than to develop a real plan; an outline to guide you through this complicated activity. What if you could just sit down at your computer and freely type in your thoughts—having a ready-made framework to hang your ideas *on*? Then you could jump around on this structure, filling it in as you go, until all the elements of your plan sit in their proper places, and your scheme takes its final shape. This sounds like a job for *The Organizer*!

Quite simply, *The Organizer* is a program that allows you to formulate thoughts and place them into an outline-like structure. You may also use the program's word processing functions to edit text or expand in detail any item in the outline—and then print the finished product on paper.

Organize It Yourself

The best way to learn about *The Organizer* program is to *use* it. When you first start the program *Organize*, you will see the title screen, and the main menu:

```
Main Menu

1.  Outline Editor
2.  Reports
3.  File Manager
4.  Quit
```

To begin, select 3, File Manager, for the next menu:

```
1.  List file names on disk
2.  Create a new Organizer file
3.  Delete an Organizer file
4.  Expand an Organizer file
5.  Return to Main Menu
```

First select number 2. A new screen will prompt you to enter a new file name. In Figure 1, we show a short example of an outline covering a kitchen remodeling project. For this example, enter REMODEL as the file name. You will also be asked to specify how many records you expect to use, up to a limit determined by your machine. A record corresponds to one item in your outline; about 200 is a good place to start. (The process of creating this file can take your computer from a few seconds to a few minutes depending on the file size and the machine brand you are using.) After this entry, you are taken back to the File Manager menu. Now select option 5 to return to the Main Menu, and select option 1, Outline Editor. After entering the name of the file you wish to work with, the file and the first screen will be loaded. When this new screen appears, you may begin your outline.

All thoughts are *not* created equal. Every idea potentially has subordinate ideas to support it. In an outline structure, an idea is represented by a "heading," and each supporting idea as a "subheading." In *The Organizer*, a "generation" is simply one heading (which we will call a "parent") and its sub-headings (which we will call "children"). Usually, as you move from one generation to the next, you go from the general to the specific. *The Organizer* causes you to look at only one generation—one parent and its children—at a time. But, at the same time, it keeps each generation in its proper place as part of a larger structure.

On the Outline Editor screen, you will see several numbers displayed, showing the: generation (Gen); number of records used; number of records free; and the row and column of the pointer. The second line displays the name of the current parent, or the current child (if using the text editor). On the third line, you will see the *line pointer*, which is a greater-than (>) sign.

Generation Gap

Understanding the role of the line pointer is crucial to using *The Organizer*. This pointer is the key to moving between generations in the outline, and to performing all the editing functions, including adding new children. Equally important is to understand that, as you work on your outline, what you see on screen is *only one small part* of the entire document—a generation of children under one parent.

> ***"What if you could just sit down at your computer and freely type in your thoughts—having a ready-made framework to hang your ideas on?"***

On this Outline Editor screen, you are usually in one of two modes: Edit Line or Outline Entry. You are in Outline Entry mode when your cursor sits on the line pointer. When you move off the line pointer onto a new line, you are automatically transferred to Edit Line mode—as you do on this first new line of the Outline Editor screen. Here you enter the first child of the primary parent. Each outline record (child) is limited to one screen-line. (With the Text Editor, you will be able to add any number of text lines to a child, as explained later.) Starting with the two most general categories, "Owner supplies" and "Contractor supplies," we would enter these as children to the main parent, "Remodel." Here we are at Generation 0.

First type in the name of the first child, "Contractor Supplies" and press [RETURN]. You will automatically jump back to the line pointer. Now move the pointer down to the next line and enter the next child, "Owner supplies."

As mentioned, the line pointer is depicted by the greater-than (>) sign. This sign serves as more than just a *pointer*, however. Greater-than (>) and less-than (<) command keys are also used to *change generations*. When you return to the line pointer, press (>). A new, blank screen will appear with the Generation number increased from 0 to 1, and the name "Contractor supplies"—now the current parent—on the second line. The line pointer is again on the third line, which is blank—awaiting a new entry. Here you may add new children under this parent or return to Generation 0 by pressing the less-than (<) sign. It doesn't matter which line you are on when you press <, it will still take you back to the preceding parent with its list of children. Remember: > takes you *forward* a generation and < takes you *back* a generation.

Now try entering the 4 items under "Contractor supplies" as its children. Move the line pointer back to the "Contractor supplies" and press >. On this new screen, with "Contractor supplies" named as the current parent, type in one child, go to the next line and type in the second child, and so on. In Figure 3, we show an expanded version of Figure 1. How did we expand it? Simply by adding generations to each item as we thought about them, one at a time, until—by the fourth generation in some cases—we had "thought it all through."

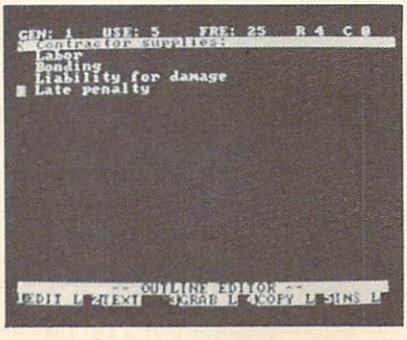
One of 3 other characters may appear to the left of a child's name, each one telling something about that child. The pound sign (#) means that the child has children of its own. The ampersand (&) means that there is text associated with that child, and the asterisk (*) means that *both* text and children accompany the child. Notice that the less-than sign (<) also appears on the screen's second line, to the left of the current parent name. This simply means that—unless you are at Generation 0—you may return to the previous generation to see this parent displayed under *its* parent by pressing <.

An outline can take many forms and can include as many generations—branching off into as much detail—as you need to cover the subject. Figure 2 shows the general structure of the outline in Figure 1, and demonstrates how items branch off of other items to create new generations. And as you can see in Figure 3, the example in Figure 1 can expand into great detail just by branching off each item—eventually creating a complete document containing all relevant information on the main heading.

You can use *The Organizer* to build the same sort of outline—creating your own headings and adding to

REMODEL
- Contractor Supplies
  - Labor
  - Bonding
  - Liability for Damage
  - Late penalty
- Owner Supplies
  - Labor Fee
  - Cost of Materials
  - Blueprints
  - Materials List

Figure 1



A representative screen photo of The Organizer on the IBM PCjr showing the Outline Entry screen.

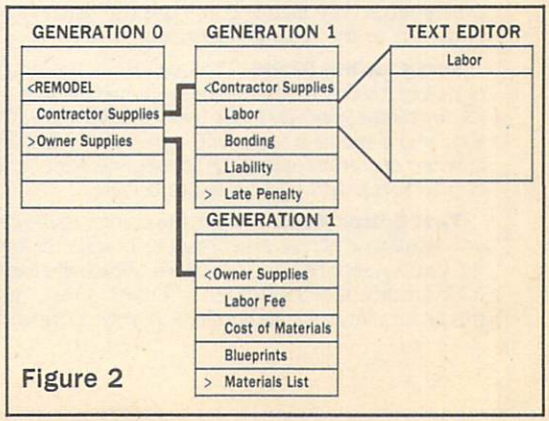


Figure 2

them as you wish. You can also write text to attach to any heading as an explanation, or to expand a heading without creating more subheadings. By the time you have built a detailed outline for, say, a term paper, the paper will be nearly written—simply put the text for each heading together with all the others and fill in as needed.

For the purpose of *The Organizer* program, we will say that the outline is like a system branching from a few, general items to an expanding number of more specific items—each item spawning a new generation of "children." In a given generation, any text you write for a particular child will also be of that same generation—because the text is an *extension* of the child for which it is written.

The Organizer is capable of creating any number of generations. It's limited only by memory or disk storage space. Almost all of the headings and text are maintained on disk, with only the current parent (the one you're working on) and its own children in memory at one time. This offers protection from major data-loss due to a power failure, or other program interruptions. It also means that you should *not* remove the disk while running the program.

"All thoughts are <u>not</u> created equal. Every idea potentially has subordinate ideas to support it."

Outline Entry Mode

As you type-in an entry, you are in Edit Line mode. Every time you finish typing an entry or "child" and press [RETURN], you will jump back to a spot one space to the left of your entered line, where the line pointer sits. Here you automatically return to Outline Entry mode. In this mode, you can make use of a large number of functions, as explained in Table 1. These functions correspond to the proper keys designated in your machine's Control Capsule. Experiment with each of these functions using the sample outline in Figure 2, or one of your own.

Entering Text

From the Outline Entry Mode, you can select Text Editor mode to add text under any item in the outline. (See Table 1.) This is the *word-processing* aspect of *The Organizer*. When you make this selection, a new screen will appear with the cursor on the third line. The first line contains the same information as the first line of the Outline Entry screen. The second line contains the name of the child to which you are adding text. Our sample in Figure 1 contains text (color-keyed) with which you can experiment using all of the text-editing functions—or you may just wish to dive into your own outline right away. These editing functions include: Moving the cursor up, down, left, or right; inserting characters; deleting characters; backspacing; and formatting.

Automatic word-wrap is always in effect while typing. If a line becomes too long either from simply typing characters, or inserting them, the last word in the line will be removed and placed on a new line below the one you are working on. When you've completed your entry, press the [FORMAT] key and the program will go back and check the end of each line for spaces, then check the first word of the next line to see if it will fit on the line above. If it does fit, the word gets moved up. This packs the document so that there are not a lot of empty lines. Usually a formatted document requires

TABLE 1

Move Pointer—The pointer indicates which child of the current parent is to be worked on. The pointer cannot move above the third line on the Outline Entry screen; but if the number of children more than fill the screen, you can scroll up or down by attempting to move the pointer off the top or bottom of the screen.

Change Generation—At the line pointer, press < to go back a generation, and > to go forward a generation.

Edit—To modify an existing child, move the pointer to that line and press the Edit key. Moving the pointer to a *new* line will *automatically* put you in Edit mode. The insert function will also put you into Edit mode. Several editing functions are available in this mode: (1) type over, (2) insert character, (3) delete character, (4) erase line, and (5) backspace. Erasing the entire line erases all characters on the line, but will leave the blank line as a child. While editing a child, the other Outline Entry functions are not available—you must press [RETURN] or [ENTER] to return to the line pointer before you can exercise these functions.

Insert—Move the line pointer to where you want to insert a line and press your Insert key. The lines on and below the pointer will move down, opening a blank space for the new line.

Grab—Use this function to move one or more children around in the outline. Grabbing removes the selected children from the current parent and places them in the hold buffer, where they remain until pasted elsewhere, or until other children are placed in the buffer. When a child is grabbed, *its* children are grabbed with it. Move the pointer to the first line to be grabbed and press the Grab key. The line pointer will be replaced with the (@) symbol. If grabbing one line, press [RETURN]. For more than one line, move the the pointer down by pressing the [down arrow] key so that a second (@) symbol appears beside the last child to be grabbed, and press [RETURN]. Items in the hold buffer are kept on disk, and will not be lost at the end of your current work session with the computer.

Copy—Use this function to make a copy of one or more children. Copies are stored in the hold buffer until pasted or replaced. Copying a child also copies *its* children. Move the pointer to the first line that you wish copied and press the Copy key. Copy multiple children in the same way you do with the Grab-Line function.

Paste—Use this function to insert the contents of the hold buffer into the outline. Move the line pointer to the point of paste, and press the Paste key. If the hold buffer contains a copy and not a grab, the items will be physically copied at this time. If you make any changes in the original items *before* pasting their copy, these changes will show up in the copy. Once a grab is pasted, it is removed from the hold buffer. A copy remains in the hold buffer to be used over and over until it is replaced.

Hold Buffer Display—You may view the contents of the buffer at any time, to prevent deleting the contents before grabbing or copying material. You cannot alter any records in the buffer, or view their children. Simply press the Hold Buffer Display key.

Delete Record—Move the pointer to a child and press the Delete Record key. Deleting a record also deletes its descendents and adds to the records-free count.

Sort Outline Mode—You can have the program sort all or part of the records in the outline numerically or alphabetically (or strictly speaking, by ASCII code). Press the Sort Outline Mode key, and a menu screen will appear with 3 options: (1) Sort children of current parent, (2) Sort a generation (enter a generation to sort:), and (3) Sort entire outline.

Text Editor Mode—Move the pointer to the child to which you want to add text, and press the Text Editor key. This mode has its own set of editing functions, which are explained in the main article under the subhead, "Entering Text," and for which the proper keys are designated in your Control Capsule.

fewer lines. The unused lines dropped by the formatter are returned to the free stack of available records. The Format command nicely formats the entire text under the current child.

To mark paragraphs, type *P at the beginning of a blank line above the new paragraph. When you print your outline with the Reports option from the Main Menu, it will indent 3 spaces at each paragraph—not printing the blank line or the *P marking. For an intentional blank line—one that you want printed—type *B at the beginning of your blank line. Using *P and *B will prevent the formatter from wrapping these lines on the screen display. Instead, they will remain as you entered them.

Text Editor can also use Grab, Copy, Paste, Display Hold Buffer, Delete Line, and Insert Line. Hold Buffer is for text only when you're in Text Editor mode. There are actually 2 buffers—one for outline, one for text. (These functions are also detailed in Table 1.)

Reports Mode

From the Main Menu, you can select the Reports option to print a hard copy of your outline. Here a new menu will ask you if you want it indented. If you answer yes, the printer will print your outline with indents for each generation, as formatted in Figure 1. The second option will print the outline showing no indents, but with the children in order under their respective parents. Under both options, a minus sign in front of each child will differentiate children from text. Text will appear under the child with which it is associated, and—with indentation—will be indented to the same generation.

Additional Options

The File Manager menu contains 3 more options which we have not yet discussed. From this menu, you can: list all the files on disk (Option 1); delete a specified file (Option 3); or increase the size of any file by redefining the number of records (Option 4). From the Main Menu, Option 4, Quit, will halt the program, returning you to BASIC. Appropriate screen prompts will guide you through each of these options.

Use It!

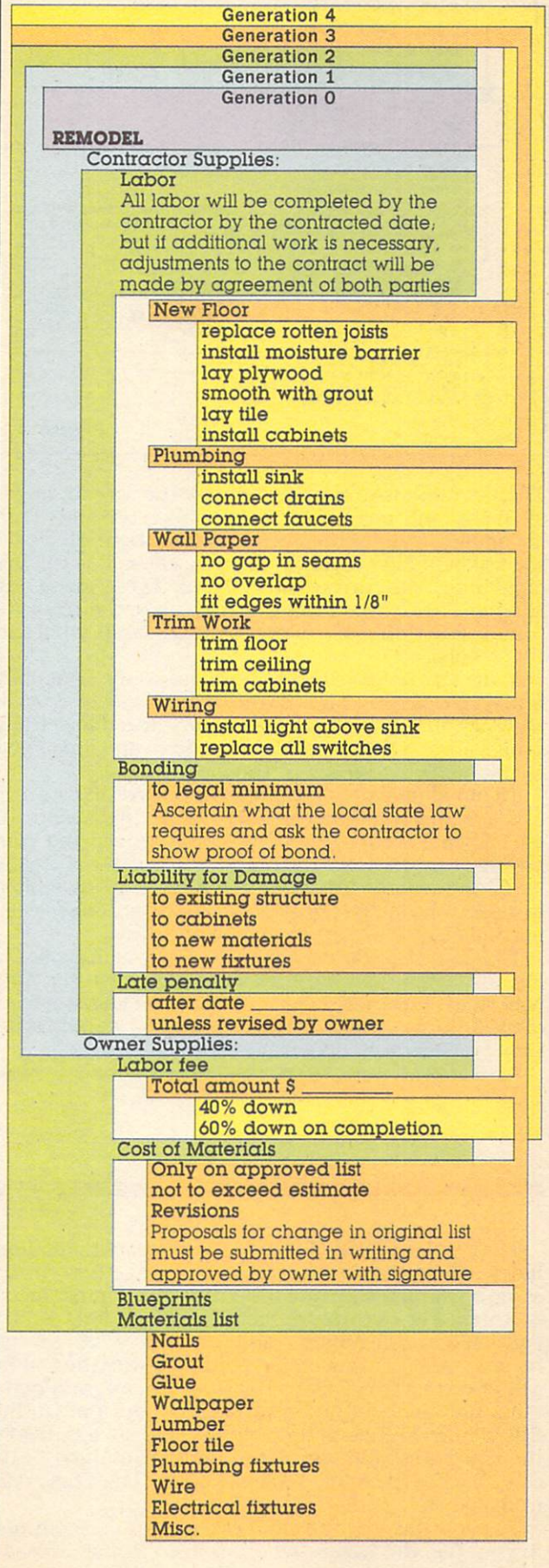
Now that we have given you the skeleton, you may dress him as you please. You will find that this program is a very handy rack to hang your thoughts on. So when important projects loom ahead—and you get that queasy, unorganized feeling in the pit of your stomach—don't panic! Just turn to *The Organizer*.

The Program

The Organizer uses random-access type files to link its records together *two-dimensionally*. Most data file programs link records *sequentially*—in one direction. When these programs search for or add records, they go from one end of the file to the other until they find what they're looking for. A *two-dimensional* file links its records not only top-to-bottom but side-to-side. Figure 4 charts the file structure of *The Organizer* and shows how this two-dimensional linking system works. This structure reflects the user-interface of the program itself. It may be useful to compare Figure 2 with Figure 4 as you follow this explanation.

Figure 4 shows how one record links to others. On the left side are 2 boxes which represent the Outline Linkage Blocks. The upper box contains the links for the screen's current parent. The second box contains the links for the first child. Each box is divided into 5 "compartments," numbered 0 through 4, each representing a link to another record. The first link (starting at the top) of the upper box goes back to the parents.

Figure 3



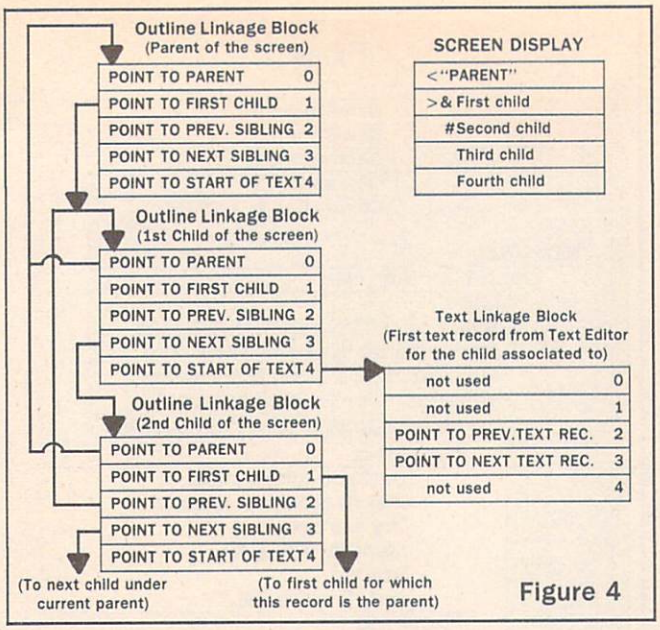


Figure 4

The next link is to the first child of the screen. Link 2 of the second box normally goes to a previous child (a "sibling") under the current parent here. However, the link will have a value of zero because it is the first child under the current parent. Link 3 goes to the *next* sibling. This leaves Link 4, which links the record to the box (the third box), which is the text associated with this record.

In the text record, the program uses only 2 out of the 5 possible links. Links 2 and 3 are used to connect together all the text records in the entire file serially. Link 2 goes to the previous text record; and link 4 goes to the next text record in the file.

If a one-dimensional file links its records like a string of beads, this two-dimensional file structure is more like an intricate weave—with the final pattern too complicated to represent here. In fact, this "pattern" created by all the records linked in this fashion is invisible to the user, who in *The Organizer* looks at only one selected portion (a generation) of the file at a time.

This same two-dimensional linking system is used by all the machine-specific versions of *The Organizer*. Each computer, however, has its own way of keeping track of the linking algorithm. Disk access and loading times vary considerably between machines.

For *your* key-in listing see HCM PROGRAM LISTINGS Contents.

The Apple version of the program stores the linkage block—the values that link records together—in the same physical record that the outline text for that record is stored. For example, Record No. 37 of "File X" will contain both the linkage block and the "text" for that record—whether that "text" is entered with the Outline Editor or the Text Editor. Because the program maintains the identical link structure in both the Outline Editor and the Text Editor, the records are identical to the computer, and can therefore be contained in the same file. On the Apple version, you can Paste only once from the Hold Buffer—*even from a Copy*.

To allow the use of all characters (including commas) in your file, the GET statement is used in disk access—so it will be moderately slow. There will also be a time delay as the Outline Editor loads.

In the Commodore version of *The Organizer*, both the linkage block and the contents of each record are kept in the same file on disk. However, the DOS "P" (Position) command allows the program to keep the linkage block as a single string in each record, and this string is converted into part of a separate integer array after it is brought into memory. This link string includes a value that corresponds to the record number in the file, so that the program can identify a linkage block with the contents of the proper record.

On the C-64, any loading procedure or operations involving disk access can be quite slow; however, the program functions proceed relatively quickly.

In the IBM version, the linkage block for a record is stored in a file separate from the contents of that record, although the two records will have the same *number* in each file. For example, Record No. 37 in the "link file" will be directly associated to Record No. 37 in the "text file" (whether that "text" was entered with either the Text Editor or the Outline Editor). The links file actually requires the use of ASCII characters, rather than integers, as identifiers. The program must convert integer values for the record's linkage block into ASCII string values and back again as it handles each record. Each linkage block is stored in a 10-byte string. These 10 bytes contain 5 numeric values that are converted by the MKI$ function into ASCII string values which the computer can then handle as "random access" fields. Later, these ASCII values are reconverted into integers by the CVI function. In the other "text" file, the program stores up to 38 characters per record. Record No. 1 always contains the "environment" of the outline: total size of file; the pointers for the Outline Editor and Text Editor hold buffers; the empty list pointers; and the pointer to the first record used.

On the IBM machines, disk access and functions involve almost no noticeable delay. Expect a very minor delay when loading.

Like the IBM version, the TI program maintains two separate files—one for linkage blocks and one for the text of the outline. These two files are associated by the record number. Each linkage block record is a string of 10 characters which comprise 5 integer values. Each integer value requires a 2-byte segment of the string. Two functions are used to handle these values: the first function converts an integer value to a 2-byte string and the second function reconverts the 2-byte string to an integer value. Two subroutines—PL and GL—are used to manipulate the 10-byte string used to contain the links. PL places an integer value into a specified position in the 10-byte string. GL retrieves an integer value from the 10-byte link string. This method allows the compact storage of a large amount of information in a small space. For example, an integer number which requires 9 bytes of memory could be stored as a character string—taking up only 2 bytes of memory.

Even though this program disables all sprites (see this issue's TI "Tech Note"), TI Extended BASIC is fairly slow, decreasing the speed of operation somewhat. There will be considerable delays in loading and disk access operations.

HCM

CONTROL CAPSULE
The Organizer

Line Pointer Edit Keys

KEY	FUNCTION
<	Back a Generation — make the parent a child.
>	Forward a Generation — make the child a parent.
Ctrl E	Edit — use [RETURN] to go back to line pointer.
Ctrl T	Text Editor — use [Esc] to go back to Outline Editor.
Ctrl L	Insert — insert a record in the outline.
Ctrl G	Grab — place line(s) in Hold Buffer; remove from the screen.
Ctrl X	Copy — place line(s) in Hold Buffer; don't change original line(s).
Ctrl P	Paste — insert contents of the Hold Buffer.
Ctrl H	Display Hold Buffer — display outline's Hold Buffer.
Ctrl D	Delete Record — purge record from file; place in empty list.
Esc	Quit — return to the main menu.
Ctrl R	Sort Outline Mode — clear screen and prompt user for sort option.
[Return]	Verify — accept current selection.
Ctrl Q	Change Scrolling Speed.
↑	Cursor Up.*
↓	Cursor Down.*

Line Edit and Text Editor Keys

KEY	FUNCTION
Ctrl A	Insert Character — insert a character on the line.
Ctrl Z	Delete Character — delete a character on the line.
Ctrl B	Erase Line — clear line from screen; don't remove from file.
→	Cursor Right — move edit cursor right one character.*
←	Cursor Left — move edit cursor left one character.*
[Return]	Return to Line Pointer — only when in Edit Mode.

Additional Text Editor Keys

KEY	FUNCTION
Ctrl F	Format Text — format entire document currently being worked on.
[Return]	—next line, first position
Esc	Return to Outline Edit Mode.
↑	Cursor Up — same as above.*
↓	Cursor Down — same as above.*
Ctrl L	Insert — same as above.
Ctrl G	Grab — same as above.
Ctrl X	Copy — same as above.
Ctrl P	Paste — same as above.
Ctrl H	Display Hold Buffer — same as above.
Ctrl D	Delete Line — same as Delete Record, above.

*SPECIAL FOR II+ USERS:	
Ctrl K	Cursor Up
Ctrl J	Cursor Down
Ctrl U	Cursor Right
Ctrl H	Cursor Left

CONTROL CAPSULE
The Organizer

Line Pointer Edit Keys

KEY	FUNCTION
<	Back a Generation — make the parent a child.
>	Forward a Generation — make the child a parent.
F1	Edit — use [RETURN] to go back to line pointer.
F2	Text Editor — use [Esc] to go back to Outline Editor.
F3	Grab — place line(s) in Hold Buffer; remove from the screen.
F4	Copy — place line(s) in Hold Buffer; don't change original line(s).
F5	Insert — insert a record in the outline.
F6	Paste — insert contents of the Hold Buffer.
F7	Display Hold Buffer — display outline's Hold Buffer.
Ctrl F8	Delete Record — purge record from file; place it in empty list.
F9	Sort Outline Mode — clear screen and prompt user for sort option.
Esc	Quit — return to the main menu.
↑	Cursor Up.
↓	Cursor Down.

Line Edit *and* Text Editor Keys

KEY	FUNCTION
Ins	Insert Character — insert a character on the line.
Del	Delete Character — delete a character on the line.
Shift F8	Erase Line — clear line of text; don't remove from file.
←	Cursor Left — move edit cursor left one character.
→	Cursor Right — move edit cursor right one character.
Esc	Return to Line Pointer — only when in Line Edit Mode.

Additional Text Editor Keys

KEY	FUNCTION
F10	Format Text — format entire document currently being worked on.
[Enter]	— next line, first position.
Esc	Return to Outline Edit Mode.
↑	Cursor Up — move edit cursor up a line.
↓	Cursor Down — move edit cursor down a line.
F3	Grab — same as above.
F4	Copy — same as above.
F5	Insert — same as above.
F6	Paste — same as above.
F7	Display Hold Buffer — same as above.
Ctrl F8	Delete Line — same as Delete Record, above.

CONTROL CAPSULE
The Organizer

Line Pointer Edit Keys

KEY	FUNCTION
<	Back a Generation — make the parent a child.
>	Forward a Generation — make the child a parent.
F1	Edit — use [RETURN] to go back to line pointer.
F2	Text Editor — use [Left arrow] to go back to Outline Editor.
F3	Grab — place line(s) in Hold Buffer; remove from the screen.
F4	Copy — place line(s) in Hold Buffer; don't change original line(s).
F5	Insert — insert a record in the outline.
F6	Paste — insert contents of the Hold Buffer.
F7	Display Hold Buffer — display outline's Hold Buffer.
F8	Delete Record — purge record from file; place it in empty list.
Cmdr S	Sort Outline Mode — clear screen and prompt user for sort option.
–	Quit — return to the main menu.
Crsr Up	Cursor Up
Crsr Dn	Cursor Down

Line Edit and Text Editor Keys

KEY	FUNCTION
INST	Insert Character — insert a character on the line.
DEL	Delete Character — delete a character on the line.
Cmdr E	Erase Line — clear line from screen; don't remove from file.
Crsr Right	Cursor Right — move edit cursor right one character.
Crsr Left	Cursor Left — move edit cursor left one character.
[RETURN]	Return to Line Pointer — only when in Line Edit Mode.

Additional Text Editor Keys

KEY	FUNCTION
Cmdr F	Format Text — format entire document currently being worked on.
–	Return to Outline Edit Mode.
[RETURN]	— next line, first position.
Crsr Up	Cursor Up
Crsr Dn	Cursor Down
F3	Grab — same as above.
F4	Copy — same as above.
F5	Insert — same as above.
F6	Paste — same as above.
F7	Display Hold Buffer — same as above.
F8	Delete Line — same as Delete Record, above.

CONTROL CAPSULE
The Organizer

Line Pointer Edit Keys

KEY	FUNCTION
<	Back a Generation — make the parent a child.
>	Forward a Generation — make the child a parent.
Fctn 8	Edit — use [ENTER] to go back to line pointer.
Ctrl 8	Text Editor — use [FCTN 9]to go back to Outline Editor.
Fctn 6	Grab — place line(s) in Hold Buffer; remove from the screen.
Ctrl 6	Copy — place line(s) in Hold Buffer; don't change original line(s).
Ctrl 2	Insert — insert a record in the outline.
Ctrl 7	Paste — insert contents of the Hold Buffer.
Fctn 7	Display Hold Buffer — display outline's Hold Buffer.
Ctrl 1	Delete Record — purge record from file; place it in empty list.
Fctn 5	Sort Outline — clear screen and prompt user for sort option.
Fctn 9	Quit — return to the main menu.
Fctn E	Cursor Up.
Fctn X	Cursor Down.

Line Edit *and* Text Editor Keys

KEY	FUNCTION
Fctn 2	Insert Character — insert a character on the line.
Fctn 1	Delete Character — delete a character on the line.
Fctn 3	Erase Line — clear line of text; don't remove from file.
Fctn S	Cursor Left — move edit cursor left one character.
Fctn D	Cursor Right — move edit cursor right one character.
[ENTER]	Return to Line Pointer — only when in Line Edit Mode.

Additional Text Editor Keys

KEY	FUNCTION
Ctrl 5	Format Text — format entire document currently being worked on.
[ENTER]	— next line, first position.
Fctn 9	Return to Outline Edit Mode.
Fctn E	Cursor Up — move edit cursor up a line.
Fctn X	Cursor Down — move edit cursor down a line.
Fctn 6	Grab — same as above.
Ctrl 6	Copy — same as above.
Ctrl 2	Insert — same as above.
Ctrl 7	Paste — same as above.
Fctn 7	Display Hold Buffer — same as above.
Ctrl 1	Delete Line — same as Delete Record, above.

The Organizer (Apple *II* Family)

Main Menu
Explanation of the Program.

Line Nos.	
100-200	Program header.
210-230	Set ProDOS flag and initialize.
240-300	Main program loop.
310-440	Initialization of variables.
450-580	Main menu screen.
590-700	Get Choice.
710-820	Branch to next program.

File Manager
Explanation of the Program.

Line Nos.	
100-200	Program header.
210	Set ProDOS flag and initialize.
220-300	Main program loop.
310-520	Initialization of variables.
530-590	Display main menu.
600-710	Get number choice.
720-880	Catalog disk.
890-1120	Create-file routine.
1130-1220	Delete-file routine.
1230-1470	Expand-file routine.
1480-1630	Number entry.
1640-1760	Headers and messages.
1770-2090	File-name-entry routine.
2100-2250	File-management routines.
2260-2500	Encode link parameters.
2510-2600	Link to main menu
2610-3010	Error-handling routines.

Outline Editor
Explanation of the Program.

Line Nos.	
100-200	Program header.
210	Set ProDOS flag and initialize.
220-410	Main program loop.
420-770	Initialize program variables.
780-1270	Steering routines.
1280-1470	Entry routines for different generations.
1480-1690	Outline-entry options.
1700-2480	Sort outline.
2490-2800	Cursor-down routine.
2810-3080	Line edit.
3090-3460	Insert line.
3470-4850	Grab and Copy line routines.
4860-4940	Display Hold Buffer.
4950-5380	Delete line.
5390-6040	Format text.
6050-6210	Scroll-lines routine.
6220-6410	Forced-format routine.
6420-7200	Miscellaneous-display routines.
7210-7540	File-name entry.
7550-7740	Disk-management routines.
7750-8620	Memory-manipulation-of-files routines.
8630-9080	Text-file-display routines.
9090-9140	Get-character routine.
9150-9240	Return to main menu program.
9250-9500	Error-handling.
9510-9610	Get ProDOS prefix.

Reports
Explanation of the Program.

Line Nos.	
100-230	Program header.
240	Start error-trapping.
250-390	Load file links.
400-440	Option to indent report.
450-670	Print report.
680-840	Error-trapping routine.

Due to space limitations, the key-in listings for the Reports Routine will be published in HCM Vol. 5, No. 2.

The Organizer (Commodore 64)

Main Menu
Explanation of the Program.

Line Nos.	
100-210	Program header.
220-290	Menu display and input.
300-320	Verify and chain to program.
330-380	Error-handling.
390-440	Program variable initialization.
450-460	Move-cursor routine.

File Manager
Explanation of the Program.

Line Nos.	
100-230	Program header.
240-290	Main program loop.
300-420	Get disk directory.
430-460	Chain to main menu program.
470-480	Data for menu.
490-770	Create-file routine.
780-860	Delete-file routine.
870-1000	Expand-file routine.
1010-1020	Move-cursor routine.
1030-1070	Initialize program variables.
1080-1170	Error-handling routines.
1180-1260	Input routine.

Outline Editor
Explanation of the Program.

Line Nos.	
100-210	Program header.
220-410	Main program loop.
420-770	Initialize program variables.
780-1270	Steering routines.
1280-1470	Entry routines for different generations.
1480-1690	Outline-entry options.
1700-2480	Sort outline.
2490-2800	Cursor-down routine.
2810-3080	Line edit.
3090-3460	Insert line.
3470-4850	Grab and Copy line routines.
4860-4940	Display Hold Buffer.
4950-5380	Delete line.
5390-6040	Format text.
6050-6210	Scroll-lines routine.
6220-6410	Forced-format routine.
6420-7200	Miscellaneous-display routines.
7210-7540	File-name entry.
7550-7740	Disk-management routines.
7750-8620	Memory-manipulation-of-files routines.
8630-9080	Text-file-display routines.
9090-9140	Get-character routine.
9150-9240	Return to main menu program.
9250-9500	Error-handling.
9510-9610	Get ProDOS prefix.

Reports
Explanation of the Program.

Line Nos.	
100-230	Program header.
240	Start error-trapping.
250-390	Load file links.
400-440	Option to indent report.
450-670	Print report.
680-840	Error-trapping routine.

Due to space limitations, the key-in listings for the Reports Routine will be published in HCM Vol. 5, No. 2.

The Organizer (IBM PC and PCjr)

Main Menu
Explanation of the Program.

Line Nos.	
100-240	Program header.
250	Start error-trapping.
260-330	Display main menu.
340-430	Error routine.
440-510	Routine to display program options.
520-580	Program option data.

File Manager
Explanation of the Program.

Line Nos.	
100-230	Program header.
240	Start error-trapping.
250-290	Display main menu.
300-330	Display a catalog of the disk.
340-430	Create a file.
440-480	Delete a file.
490-520	Exit back to the main menu.
530-550	Display title.
560	Option data.
570-640	Error-trapping routine.
650-830	Expand-the-file-size routine.

Outline Editor
Explanation of the Program.

Line Nos.	
100-260	Program header.
270-280	Start error-trapping—initialize program.
290-370	Load file links.
380-460	Main control loop for Outline Editor.
470-500	Edit line control routine.
510-550	Text Editor control routine.
560-630	Text Editor screen loader.
640-690	Text Editor screen saver.
700-1110	Grab-and-Copy routine.
1120-1220	Insert-line routine.
1230-1630	Paste-line routine.
1640-1720	Display the Hold Buffer.
1730-1910	Delete a line from the file.
1920-2180	Sort the outline.
2190-2470	Format text in the Text Editor.
2480-2590	Get next record from the empty list.
2600-2750	Text Editor cursor routines.
2760-2790	Set up screen.
2800-2880	File-handling routines.
2890-3010	Load and display Outline screen.
3020-3540	Line Entry routines.
3550-3620	Change level routines.
3630-3680	File field routines.
3690-3900	Error trapping routine.
3910-3940	Mode display subroutines.

Reports
Explanation of the Program.

Line Nos.	
100-240	Program header.
250	Start error-trapping.
260-370	Load file links.
380-410	Option to indent report.
420-570	Print report.
580-680	Error-trapping routine.

Due to space limitations, the key-in listings for the Reports Routine will be published in HCM Vol. 5, No. 2.

The Organizer (TI-99/4A)

XB All *Organizer* programs require TI Extended BASIC.

Main Menu
Explanation of the Program.

Line Nos.	
100-240	Program header.
250-270	Display main menu.
280-340	Run program from main menu

File Manager
Explanation of the Program.

Line Nos.	
100-250	Program header.
260-290	Main menu.
300-370	Display catalog of the disk.
380-450	Create a new file.
460-490	Delete a file.
500-590	Expand a file.
600	Exit to main menu.
610	Routine to display title header.

Outline Editor
Explanation of the Program.

Line Nos.	
100-250	Program header.
260-310	Initialize program.
320-390	Load file links.
400-510	Main control routine for the Outline Editor.
520-810	Delete-line routine.
820-980	Insert-line routine.
990-1530	Paste-line routine.
1540-1940	Sort outline.
1950-2310	Format text for the Text Editor.
2320-2780	Grab and Copy line routine.
2790-2870	Display Hold Buffer.
2880-2910	Edit line control routine.
2920-2970	Text Editor control routine.
2980	Exit back to the main menu.
2990-3070	Line pointer control.
3080-3160	Text Editor screen loader.
3170-3220	Text Editor screen saver.
3230-3900	Line-entry routine.
3910-3960	Miscellaneous routines.
3970-4070	Disk-I/O routines.
4080-4240	Load outline screen.
4250-4330	Set up screen.
4340-4430	Get next outline record.
4440-4520	Get next text record.
4530-4560	Program data.
4570-4590	GL subroutine to get link value.
4600-4660	PL subroutine to place link value.

Reports
Explanation of the Program.

Line Nos.	
100-250	Program header.
260-370	Load file links.
380-410	Option to indent report.
420-590	Print report.

Due to space limitations, the key-in listings for the Reports Routine will be published in HCM Vol. 5, No. 2.

Orbital Defender

by Scott Williams
and the HCM Staff

As you seem to hang in the heavens, Earth hangs its hopes on you. With only a split-second's warning, will you recognize the enemy?

Friend or Foe? As a lonely guard stands ready, this is always the bottom line. True of sentries everywhere, at all times, this question now presses on the entire planet—and you are all that stands between Mother Earth and the invading hordes from outer space. *Orbital Defender* is a game based on reflexes and hand-eye coordination—placing you in command of a patrol ship orbiting Earth. Your mission is to recognize and destroy any hostile ship approaching the planet, but to let friendly ships pass unharmed. Your enemies are the Sandian hordes, and the space pirates from Alpha II. Traffic is heavy, and you have to make instant decisions; or, you may unwittingly miss a target, destroy an ally, or possibly be shot down yourself!

Think Fast

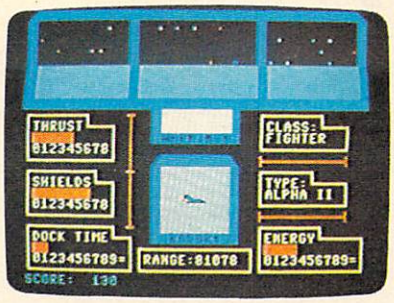
As Planet Earth slides slowly beneath, and the stars appear to rise over a curved horizon, you fly a lonely and dangerous vigil. Your ship is armed with a short-range particle-beam discharge cannon. A radar scope in your control panel homes in on any passing spacecraft and displays its shape. It is your decision to fire or not—and it must be a split-second judgement: friendly or hostile? Before you see the craft on the screen, a warning light and alarm brings you to attention. If you recognize the ship as an enemy, press the [SPACE BAR] to fire. Your ship's computer locks in the target, but you must decide who's who. Earth law declares that only a human being—not a computer—will decide whether life is to be taken in defense of the planet. You'll be awarded points for every enemy you shoot, but docked for each Earth ship that you mistakenly destroy.

Your ship's instrument panel is an active one, displaying several factors important to your survival and success. Three sliding scales indicate thrust, your shield strength (if an attack catches you with your shields down, you're dead), and dock time—the time left before your ship reaches an Earth base satellite. Three information displays show the class, type, and firing range of any ship on the radar screen, and a fourth sliding scale shows the level of energy remaining in your ship.

Energy and Skill

Your thruster, shields, and cannon all consume energy—running out of energy makes you a perfect target for the Sandians or Alpha II pirates. The only way to get more energy is to dock with an Earth base, which is easy—as long as you don't blow *it* out of the universe. If your itchy trigger finger destroys the base before you get there, you'll have to conserve energy until the next dock. You can, for example, turn off your shields to conserve, but you will have to shoot perfectly to survive until you can reach the next base. If you shoot more than 6 *Earth* ships though, you will not be allowed to dock with an Earth base for the rest of the game.

From the first menu, you can choose 10 different skill levels, ranging from Rookie Cadet to Commander. Skill levels vary according to how much time you have to react before firing or taking a blow. The program keeps track



A representative photo of Orbital Defender taken from the Commodore 64 screen.

"Earth law declares that only a human being—not a computer—will decide whether life is to be taken . . ."

of how many ships of what type are destroyed, and it displays this along with your score at the end of the game.

Now sit down at the controls and do your best to defend our Mother Planet. Just be sure to stay on your toes, and—before you fire—remember to ask, *Friend or Foe?*

For *your* key-in listing see HCM PROGRAM LISTINGS Contents.

CONTROL CAPSULE
Orbital Defender

KEY	FUNCTION
Space Bar	Fire
A	Increase Thrust
Z	Decrease Thrust
S	Raise Shields
X	Lower Shields

The Apple version of this program uses four assembly language routines. These routines control sound effects, character graphics, and two other very interesting functions: shape expansion and page switching. In the shape-expansion routine, the program takes a small shape from the shape table and enlarges it on the screen. The page-switching routine rapidly switches between two screens stored separately on high-resolution graphics Pages 1 and 2. (For more on these graphics pages, see the Apple "Tech Note" in this issue, on page 60.) The list below shows the line number where the data statements for each routine are located:

Routine	Data Line Nos.
1. Char. Graphics	1230-1240
2. Sound	1240
3. Page Move	1240-1250
4. Shape Expansion	1250-1280

The call addresses for each routine are all listed in lines 250 and 260 of the program:

```
250  PAUSE = 30:NS = 25:P1 = 32:P2 =
     64:BASE = 2048:DIM XS(NS),
     YS(NS):FX(1) = 39:FX(2)
     =139:FX(3) = 239:C(1) = 2:C(2) =
     5:C(3) = 6:C(4) = 0
260  HCHAR = BASE:SOUND = BASE + 55:SPKR
     = BASE + 67:MOVE = BASE + 75:OUT =
     BASE + 115:TABLE = BASE
     +300:SHPTB =768
```

Line 250 sets the base address for all the routines in memory, and line 260 defines each address relative to the base. When one of these routines is called by the program, a parameter is passed to the machine language routine, causing the routine to run in a certain way (depending on the value of the parameter). For example, line 790 calls the sound routine with a parameter value of 255, creating a sound for approximately 1/4 second, starting with a high pitch and decreasing. If you change this value to 63, it will create several fast oscillations from high to low pitch. Try it. If you wish, you may try other values in this parameter, or find new lines in which other routines are called, and change their parameters—just to see what happens. Or try this: After running the program, the sound routine remains in memory. Halt the program and enter the following line:

```
FOR Z = 0 TO 255:HOME:PRINT Z:CALL SOUND, Z:
FOR TD = 1 TO 1500:NEXT:NEXT
```

Now as you change values in the sound parameter, you will hear the immediate results.

CONTROL CAPSULE
Orbital Defender

KEY	FUNCTION
Space Bar	Fire
T	Increase Thrust
Shift T	Decrease Thrust
S	Raise Shields
Shift S	Lower Shields

Although the Commodore 64 provides 8 different sprites, all of which can be used simultaneously, this program uses only one sprite—which it redefines to get all 9 different shapes of the hostile and friendly crafts displayed on the radar screen. Because it displays only one shape at a time, the program avoids having to redefine more than one sprite. Line 1430 sets an offset that determines which shape will be POKEd to the sprite. Line 1440 POKEs the shape data to the sprite from a shape table stored in memory. Data for the shapes are in separate data statements at the end of the program.

Continued on next page.

CONTROL CAPSULE
Orbital Defender

KEY	FUNCTION
Space Bar	Fire
Cursor Up	Increase Thrust
Cursor Down	Decrease Thrust
Cursor Left	Lower Shields
Cursor Right	Raise Shields

This version uses IBM's GET and PUT commands to generate changes in the graphics screen. The GET command allows the program to grab a rectangular section of the screen and save it to an array. (Two parameters define the rectangle by the position of its upper left- and lower right-hand corners.) The PUT command allows you to place graphics stored with the GET command back on the screen.

The use of GET and PUT allows simple animation. Several options can be used with the PUT command to determine how the image will be placed on the screen. In this program, we use the PSET when displaying ships on the radar, which will place the entire image back just as it was when we grabbed it with GET. Then we use the LINE command to erase the radar image. The BF (Box Fill) option is used with a black background color—a very fast and effective way to erase controlled areas of the screen.

CONTROL CAPSULE
Orbital Defender

KEY	FUNCTION
Space Bar	Fire
E	Increase Thrust
X	Decrease Thrust
D	Raise Shields
S	Lower Shields

[Note: This program has been converted from the original in Extended BASIC to a BASIC version. To conserve space, we are publishing only the BASIC version in this issue of the magazine. However, the Extended BASIC program will be included with the BASIC program on our disk and cassette media service (ON DISK and ON TAPE) for Vol. 5, No. 1.]

The most significant aspect of the TI version of *Orbital Defender* is its use of the CALL COLOR command to enhance graphics. With CALL COLOR, the program can "hide" parts of the graphics screen until they are needed to create certain effects. This is very helpful in creating the illusion that extensive graphics are being rapidly and simultaneously written to the screen when, in fact, they were there all the time.

HCM

Orbital Defender (Apple *II* Family) Explanation of the Program.

Line Nos.	
100-200	Program header.
210-270	Initialize program.
280-290	Display title screen.
300-350	Read program data for shapes.
360-370	Start sound routine.
380-390	Get skill level.
400-530	Display playing screen.
540-700	Main control loop.
710-870	Radar picks up a ship.
880-940	End of game routine.
950-1010	Dock with Earth base.
1020-1050	Radar display routine.
1060-1290	Program data.
1300	Error routine.

Orbital Defender (C-64) Explanation of the Program.

Line Nos.	
100-180	Program header.
190-260	Display title screen.
270-520	Initialize program.
530-630	Display level option menu.
640-880	Display the playing screen.
890-1050	Main control loop.
1060-1120	Scroll stars.
1130-1160	Scroll the Earth.
1170-1570	Radar picked up a ship.
1580-1640	Read keyboard for controls.
1650-1770	End-of-game routine.
1780-1930	Dock with Earth base.
1940-2930	Program data.
2940-2980	Sound data.

Orbital Defender (IBM PC and IBM PCjr) Explanation of the Program.

Line Nos.	
100-220	Program header.
230-350	Initialize program and graphics.
360-440	Main control loop.
450-630	Display title screen.
640-710	Display playing screen.
720-760	Flash warning light.
770-970	Display controls.
980-1160	Rotate the Earth.
1170-1190	Input skill level.
1200-1250	Update controls.
1260-1460	Main control loop.
1470-1700	Sound and graphics routines for lasers and blasts.
1710-1790	End of game. Option to play again.

Orbital Defender (TI-99/4A) Explanation of the Program.

Line Nos.	
100-180	Program header.
190-230	Initialize program.
240-250	Branch to display level option.
260-340	Display playing-screen control panel.
350-420	Main control loop.
430-480	Routine to scroll the earth.
490-890	Radar picked up a ship.
900-990	Read keyboard for controls.
1000-1040	End of game—option to play again.
1050-1160	Dock with Earth base.
1170	Display screen-graphics routine.
1180-1630	Program data.
1640-1680	End of game data for display.

QUIZ-PRINT

A Finishing Tool for your Quiz Construction Set*

by William K. Balthrop
and the HCM Staff

Calling all teachers, students, trivia and non-trivia buffs—now you can create a custom-organized printout of all your self-made quizzes with this do-it-all program!

*In the last issue of *Home Computer Magazine* (Volume 4, Number 5) we presented "Quiz Contruction Set," an entertaining and educational set of programs designed to enhance your personal library of computer tools. We now offer this further addition that allows you to control the "shape-of-things-to-come."

Mary is studying for her history exam of next week, and as she reads her textbooks she devises questions and answers on her computer and later prints them out so that she may quiz herself and her study group, as well as highlight information presented. Jim is getting ready for a dinner party, and after shoving a roast in the oven, he sits down at his computer to create a set of trivia questions and answers for a little fun after dinner. And Karen types up and prints out a multiple choice test (and an answer sheet) that she will distribute for her second grade class' spelling test tomorrow.

All of these people are using *Quiz-Print*, a helpful, easy-to-use program designed to be a supplement to the *Quiz Construction Set* programs, *Quiz-Make* and *Quiz-Take*, published in the previous issue, Vol. 4, No. 5. *Quiz-Print* is a reporting utility which is capable of producing hard-copy quizzes from a quiz data base created with *Quiz-Make*.

A number of options make this more than just a reporting program. You can reorganize the quiz, use multiple-choice formats, or blind-answer questions.

When you RUN the program, you will be presented with a menu screen first:

```
MAIN MENU SCREEN

1)  SETUP
2)  PRINT
3)  SAVE
4)  LOAD
5)  EXIT
```

Before using *Quiz-Print*, you must create a quiz using the *Quiz-Make* program. To start, select the Setup routine and insert a disk or cassette containing a *Quiz-Make* file into the drive or recorder. After selecting 1) SETUP, you will be asked to enter the quiz file name that was created using *Quiz-Make*. That file will then be loaded into memory. Do *not* use LOAD to load this file—it is used *only* to load your quiz report once you have set it up and saved it.

Option 1, Setup, is used to define the parameters of the hard-copy quiz that you wish to produce. This option first helps you select a format for your answers, and then lets you choose the questions to be asked if you do not want the computer to randomly choose them.

Format The Answers

The next question you will be asked is whether you would like to print multiple-choice style questions:

```
PRINT MULTIPLE CHOICE (Y/N)?
```

If you press N, then the program will skip down to the menu described under "Choosing Questions." When you choose not to have your questions arranged in a multiple-choice format, the printout simply includes the question alone, and a blank line for the answer.

If you press Y, you will be prompted for the following information in reference to a multiple-choice quiz. First enter the number of answers you would like to have appear below each question:

```
HOW MANY CHOICES (2 to 10)?
```

Next, you can either have the computer randomly select answers from within the quiz to fill in the incorrect multiple choices, or you can enter your own wrong answers to each question:

```
SELECT ONE:
  1) USE RANDOM ANSWERS FROM THE QUIZ
  2) ENTER YOUR OWN ANSWERS
```

Select the first option if you want the computer to select multiple-choice answers for you after you have finished choosing your questions. It will display its selection for the first question and ask if it's OK. If you press N for no, then it will select a different set of random answers for your inspection. No answer will ever be repeated twice within one multiple-choice question. For this reason, you should have at least as many questions in the file as there are number of choices per question, or else choose to enter the answers yourself for each question.

If you elect to enter your own answers for your questions (option 2), you will need to enter one answer for each choice, determined by the number of choices per question that you indicated in an earlier option. The number of answers that you will enter for a question is actually one less than its total number of possible answers, because one of them will be the correct answer—which you cannot change.

Pick The Proper Place

In the next option, you can either let the computer place the correct answers within the proper fields of multiple answers randomly, or select the position that the answers will appear in for each question yourself:

```
SELECT ONE:
  1) RANDOM ANSWER POSITION
  2) SELECT ANSWER POSITION
```

If the *computer* randomly picks the position of the correct answer within each multiple choice selection, it will do so and then ask you if it's OK. If it is not, it will keep selecting different positions until one is satisfactory. If *you* want to enter the position of the correct answer, the computer will display all of the wrong answers for a question, each next to a letter. The last letter is on a blank line, and the question's correct answer will be displayed below it. Press the keyboard letter corresponding to where you would like to place the correct answer. The answer on the chosen letter—if there is one—will be bumped down, and the correct answer will be inserted.

Choose The Questions

This next menu will always appear next when you set up a quiz, regardless of whether you selected the multiple choice option:

```
SELECT ONE:
  1) SHOW QUESTIONS IN SAME ORDER AS FILE
  2) RANDOM QUESTION ORDER
  3) PICK AND CHOOSE QUESTION ORDER
```

In the above menu you are given three choices. In choice 1, the questions in the quiz will be printed in the same order in which they appear in the original *Quiz-Make* file. If you are working on a multiple choice quiz, you are then shown the first question, some answer choices, and the correct answer and asked:

```
IS THIS OK (Y/N)?
```

If you press N, then the program will continue to show you answer choices until you press Y. If you press Y, then the program will tell you how many records (questions) you have selected, and inquire whether you want to add another. If you are not working with multiple choice questions, then you are simply shown each question and asked whether you want to add it to the file. In both cases, you are taken back to the main menu when you decide to stop adding questions.

In the second choice, the computer will mix up the order in which the questions will be printed. You will be shown each question that the computer has selected and asked to accept or reject it. From here on, choice 2 operates just like choice 1 above.

"You can reorganize the quiz, use multiple choice formats or blind-answer questions."

In the third option, you will be able to scan through the quiz file and pick the questions that you would like to include in the quiz; they will be printed in the order in which you select them. Each question can be selected only once. Once you select a question, it will be removed from the list of questions from which selections can be made. Use the arrow keys to scroll through the selections. Enter the number of the question you would like and press [ENTER] or [RETURN]. If you already have a list of the questions printed out or written down, you can simply enter the numbers of the questions desired in the order that you want them, without scrolling the screen. The question does not need to be on the screen to be selected. If you are required to enter more information for a question after making a selection (such as your own wrong answers for the multiple choice option), you will be asked for that information before making your next selection.

Once you have selected your questions in any of these options, you will not be able to view them again until you obtain a printout of your report file.

Now For The Cheatsheet!

The final prompt you will need to answer to set up your quiz will give you the option of printing an answer sheet at the end of the quiz to be used in grading papers.

```
PRINT ANSWERS AT END OF REPORT (Y/N)?
```

The answer sheet includes the report title, a second heading of your choice, and the date, in addition to the number of each question and its correct answer.

The other 4 options on the main menu are fairly self-explanatory: Option 2, Print, will neatly generate the quiz that you set up in option 1. Like the answer sheet, it includes the report title, second header, and date. Option 3, Save, will save the quiz report parameters to disk or tape. By doing this you will need to design your quiz only once, loading the quiz from disk or tape to get another copy. Option 4, Load, will allow you to load the quiz report parameters previously saved with option 3. Option 5, Exit, allows you to gracefully exit the program. Each option except 5) EXIT automatically returns you to the main menu when you are done working in it.

We think you'll find that with the flexibility in print formatting that *Quiz-Print* offers, the spectrum of applications for the *Quiz* programs is extremely broad. Now take a look at *Quiz-Print Tutorial* on the next two pages for some technical insight into print formatting.

HCM

For *your* key-in listing see HCM PROGRAM LISTINGS Contents.

Quiz-Print (Apple *II* Family) Explanation of the Program

Line Nos.	
100-200	Program header.
200-290	Initialize program.
300-360	Main menu selection screen.
370-890	Control loop to set up initial report.
370-500	Load Quiz-Make file.
510-520	Prompt for multiple choice.
530-540	Number of choices.
550-560	Location of answers.
570-590	Answer position.
600-610	Order of questions.
620-650	Select answer sheet at end of report.
660-800	Control routine for building a quiz report.
810-890	Is question OK? Add another?
900-1130	Pick and choose questions.
1140-1200	Select question at random.
1210-1260	Select random answers for multiple choice.
1270-1320	Enter multiple choice answers.
1330-1380	Random position for answers.
1390-1440	Choose position of right answer.
1450-1610	Save quiz report.
1620-1750	Load a quiz report.
1760-1920	Print quiz report.
1930-2050	Print answer sheet.
2060-2330	Key-scan routine.
2340-2490	Error routine.
2500-2570	Exit program routine.

Quiz-Print (IBM PC and PCjr) Explanation of the Program

Line Nos.	
100-180	Program header.
190-240	Initialization.
250-290	Main menu. Input user's selection.
300-490	Get user's report options.
300-360	Load Quiz-Make file.
370	Select multiple choice.
380	Number of choices.
390-400	Location of answers.
410-440	Select answer position.
450-460	Order of questions.
470-490	Select answer sheet at end of report.
500-610	Control the construction of the quiz report.
620-780	Pick and choose questions.
790-840	Select questions at random.
850-900	Random multiple-choice answers.
910-960	User multiple-choice answers.
970-1020	Select random position for multiple-choice answers.
1030-1060	User selects position for multiple-choice answers.
1070-1110	Save quiz report.
1120-1160	Load quiz peport.
1170-1270	Print quiz report.
1280-1360	Print answer sheet.
1370-1420	Key-scan subroutines.
1430-1500	Error-trapping routine.
1510-1540	Exit the program.

Quiz-Print (C-64) Explanation of the Program

Line Nos.	
100-190	Program header.
200-270	Initialize program variables.
280-380	Main-menu selection screen.
390-990	Get user's options.
390-670	Read Quiz-Make file.
680-720	Select multiple choice.
730-780	Enter number of answers for multiple choice.
790-830	Location of answers.
840-880	Enter answer position.
890-940	Select question order.
950-990	Select answer sheet.
1000-1210	Control routione to construct the quiz report.
1220-1510	Pick and choose questions.
1520-1560	Select questions at random.
1570-1610	Select random answers.
1620-1680	Enter multiple-choice answers.
1690-1730	Select random posion for right answer.
1740-1780	Choose position of right answer.
1790-2020	Save quiz report.
2030-2250	Load a quiz report.
2260-2590	Print quiz report.
2600-2710	Print answer sheet.
2720-2730	Locate-cursor routine.
2740-2860	Input routine.
2870-2890	Single-key input routine.
2900-2950	Illegal-entry routine.
2960-2980	Exit routine.

Quiz-Print (TI-99/4A) Explanation of the Program

Line Nos.	
100-180	Program header.
190-290	Initialization.
300-450	Main menu. Input user's selection.
460-880	Get user's report options.
460-530	Load Quiz-Make file.
540-580	Select multiple choice.
590-640	Number of choices.
650-700	Location of answers.
710-760	Select answer position.
770-820	Order of questions.
830-880	Select answer sheet at end of report.
890-1050	Control the construction of the quiz report.
1060-1550	Pick and choose questions.
1560-1650	Select questions at random.
1660-1760	Random multiple-choice answers.
1770-1850	User multiple-choice answers.
1860-2000	Select random position for multiple-choice answers.
2010-2080	User selects position for multiple-choice answers.
2090-2200	Save quiz report.
2210-2340	Load a quiz peport.
2350-2760	Print quiz report.
2770-2950	Print answer sheet.
2960-3100	Key-scan subroutines.
3110-3200	Exit the program.

Quiz-Print Tutorial

by the HCM Staff

Accessing the Printer

There are primarily two types of DOS (Disk Operating System) currently being used on the Apple: DOS 3.3, and ProDOS. DOS is responsible for interfacing the computer to the outside world. ProDOS has fixed several bugs that existed in the earlier DOS 3.3. One such bug affects the way we initiate communications with a printer.

Both DOS manuals tell you to use the following command to start communications with a printer:

```
PRINT D$;"PR#";SL
```

But many DOS 3.3 programmers prefer to use a shortcut around DOS with the following command:

```
PR#1
```

This command will not work at all with ProDOS. If you attempt to use it, your program will not function properly. Use the following command to redirect information back to your screen:

```
PRINT D$;"PR#0"
```

Output Algorithm

The basic algorithm for figuring out the number of questions that will appear on a page is identical to the one used on all the other machines. The formula is located in lines 1860. See the IBM explanation for formulas and details.

Formatting

The *Quiz-Print* program is written to output to the 40-column Apple screen, but printers can normally also output in 80-column format. In order to have the printouts take advantage of this 80-column format without requiring an 80-column card in your Apple II+ or IIe, line 1840 contains a command that turns off the screen display while printing, and sets the width to 80 columns:

```
1840 PRINT D$;"PR#";SL: PRINT CHR$ (9);"80N"
```

The commands VTAB, HTAB, INVERSE and NORMAL have no affect on output to the printer. If you wish to format your document across the paper, you will need to use the TAB(col) command, where col is the column you want to TAB up to.

HCM

Accessing the Printer

Quiz-Print on the Commodore 64 uses the standard printer which attaches to the special serial port (the same port which connects the disk drive). The command format to turn on the printer is OPEN *lfn,dn*. The *lfn* is the logical file number—and we use 4. The *lfn* is then used in all the PRINT# statements in the program to output to the printer. The *dn* is the device number. We again use 4 because this is the default value on Commodore printers. Selecting a logical file number that is the same as the device number helps avoid confusion in reading the program's code.

Each time something is sent to the printer, we use the PRINT#4,*var* command—where *var* is the name of the variable or string to be printed. Always place the # symbol immediately after PRINT—including a space causes a syntax error. To terminate output to the printer, the buffer must first be cleared with a PRINT#4 command and is not followed by a variable. The channel is closed with a CLOSE4 command.

Output Algorithm

The basic algorithm for figuring out the number of questions that will appear on a page is identical to the one used on the other machine brands. Its formula is located in lines 2440. See the IBM explanation for formulas and details.

Formatting

The major formatting consideration here stems from a lack of a form feed on Commodore printers. To approximate this function, the program keeps track of the number of lines that have been printed on any page by using the NL variable. This value is then subtracted from 66 to determine how many blank lines are to be printed to reach the top of the next page.

Another important formatting consideration is the answer sheet. The TAB function does not work the same way on a printer as it does when printing to the screen. Thus, when multiple-choice questions are selected, a special function is used in lines 2630 through 2650 to space the answers with five answers in each row. By taking the two rightmost characters from the STR$ of the number of the answer, a one-digit number lines up in the same column as a two-digit number.

HCM

One of the main arguments in favor of purchasing a computer is that it will *reduce* the amount of paperwork a task involves. Occasionally though, we need computers to *produce* paperwork for us. The computer is capable of taking a lump sum of information, mixing it around, and spitting it out in a formatted, easy-to-read report. *Quiz-Print* is just such a program.

This article serves a dual purpose: It will further enhance your understanding of the *Quiz Constuction Set* of programs, especially *Quiz-Print*; and it will also increase your general understanding of how to get a printer to do what you intend—by translating your specific format requirements into simple BASIC commands. With this new knowledge, you should be off to a great start at developing your own custom programs to generate reports.

We have placed all the explanations for each machine brand directly beside each other so that they will be easy to compare—both in their differences, and in their similarities.

Accessing the Printer

The IBM computer makes communication with the printer a breeze—with such easy-to-use commands as LPRINT. You don't need to worry about opening a printer port. The system automatically outputs to the default printer device. If you wish to use a device other than the normal (default) printer channel, you will need to use the OPEN "COM command (which is beyond the scope of this article).

Output Algorithm

Quite often a report needs to span more than one page. One formatting rule which we have adhered to in *Quiz-Print* is that no question is cut in half by a page break. Some printers have the ability to automatically skip the page breaks, but this, in itself, is not satisfactory for our application.

Each question in the printed quiz uses several lines on the printer. If the number of lines used by a question were always the same, it would be a simple matter of limiting each page to a fixed number of questions. However, the number of lines used by each question varies depending on whether the questions are multiple choice, and on the number of choices supplied with each question.

By dividing the number of lines per page by the number of lines used for each question, we arrive at the number of questions per page. To find the number of lines used for each question, you add the number of multiple choices, if any, to a fixed overhead number of lines. In this program, the minimum number of lines needed is seven. Line 1220 does all of this:

```
1220 NPP=INT(60/(7+NC))
```

NC = the number of multiple choices selected

Formatting

For the program to properly format your quizzes, you must have a printer capable of performing a form feed that is implemented by the program. After printing the specified number of questions, a form feed (ASCII 12) is sent to the printer. This is done in lines 1210, 1230, 1250 and 1340.

HCM

Accessing the Printer

In TI BASIC, the INPUT command allows you to enter any printing parameters required for a specific printer. In any printing operation, including *Quiz-Print*, you will be prompted for this information before you begin. The kinds and number of parameters are determined by the type of printer and interface your system includes.

Most printers for the TI-99/4A require a serial interface card, which plugs into the peripheral expansion box. This card actually allows *either* serial (RS232) *or* parallel communications to the printer. If the printer is set up for parallel communications, you will only need to specify *PIO* to satisfy all parameters. If you are using the RS232 serial interface, the parameters are more complex. In this case, you must specify: (1) baud rate (bits per second); (2) whether each ASCII character sent will consist of either 7 or 8 bits; and (3) even, odd, or no parity.

If you don't know what to specify for your printer, consult your printer's user handbook, which should contain the proper information. Most TI-compatible printers come with a default setting that matches the 99/4A's RS232 serial interface. This would require that you specify a baud rate of 300, 7 bits, and odd parity.

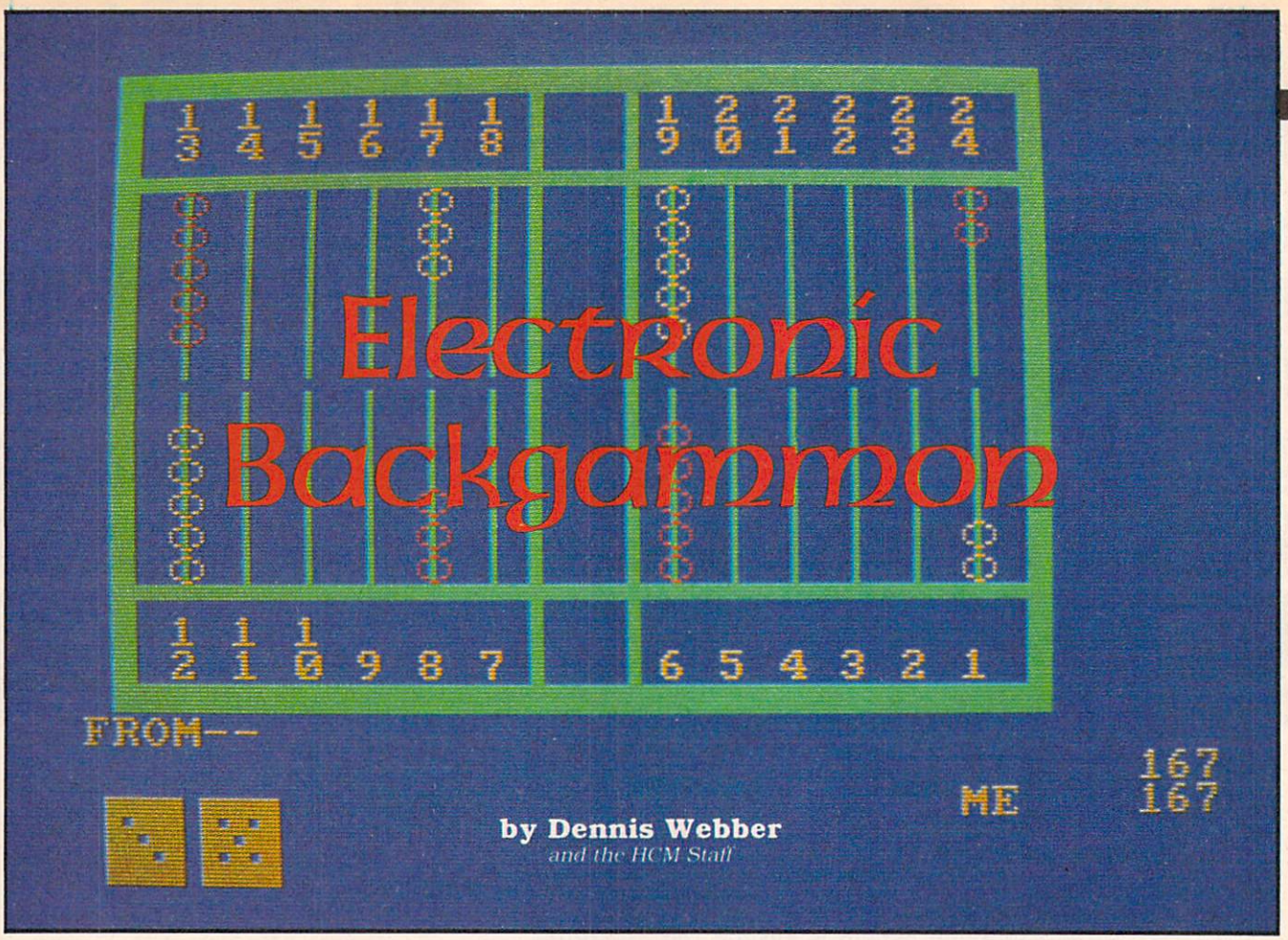
Output Algorithm

The basic algorithm for determining how many questions in *Quiz-Print* will appear on a page is identical to that used for the other machines. See the IBM explanation under this heading for formulas and details.

Formatting

For nice, neat pages in *Quiz-Print*, you want your printer to print a specified number of questions per page, and then skip to the top of the next page before continuing. Thus, the program sends ASCII code 12 to the printer after sending each page's questions, which tells the printer to do a form feed. Most printers are capable of recognizing this code—but if your printer cannot, it will simply ignore the code and continue printing *without* a form feed.

HCM



When it comes to making a logical move in Backgammon, the Ur of Chaldes didn't have anything on your home computer.

Precursors of the classic game Backgammon have been found to exist in several ancient civilizations, including those of the early Greeks and Romans. But the earliest version of the game is believed to have been created in 3000 B.C. by the Ur of Chaldes. (A land called Chaldea was a region of South Babylonia along the Euphrates and the Persian Gulf.)

Perhaps you are already an old hand at this old game, or maybe you're one of those who has always wondered "What's this strange design on the back of my checkerboard?" Anyone familiar with Backgammon, however, knows that it can challenge Checkers or even Chess in its ability to involve and interest players from all around the world. In this version of the game, the computer is your opponent—and a darn good one at that.

Know The Rules

The Backgammon board contains 24 positions—12 on each side. Players move their checkers (commonly called pips) around the board on these locations through rolls of 2 dice. (See title picture.)

The computer plays black and moves clockwise, while the user plays white and moves counterclockwise. The game board is set up in the following manner:

POSITION	PLAYER	# OF PIPS
1	computer	2
6	user	5
8	user	3
12	computer	5
13	user	5
17	computer	3
19	computer	5
24	user	2

The computer's "inner table" is positions 19 through 24, and the user's inner table is positions 1 through 6. The object of the game is to first move all of your pips to your own inner table, and then off the board. The computer rolls the dice for you; on the first roll, the one with the highest number will begin the game first. Your computer opponent will tell you to press any key to roll the dice. After the roll is determined, you may move one or two of your pips, using both dice (one at a time) to move one pip, or moving two pips—one with each die.

Once the computer has rolled the dice, it will ask you to enter a move. Numbers less than 10 must be entered with a leading zero: 01, 04, 07, etc. The prompt FROM— indicates that the computer expects you to enter the current location of the pip you would like to move. After entering a legal location, it will then add to the prompt. For example, if your first entry was 13, the display would look like this: FROM—13 TO— You would now need to enter the location where you would like to move the pip. If the move is legal, the graphics display of the board is updated to show the pip at its new location. With each move, all versions except the IBM actually redraw the entire board.

You can land only on an empty position, on a position occupied by your own pips, or on top of a *single* pip of your opponent. When this happens, the computer's piece is placed on the Bar, which is in the center of the board. If one of your pips is relegated to the Bar, you must place it back on the board before you can move any other pips. To get back on the board, you must roll a value that can move the pip onto the board without landing on a position occupied by 2 or more opponent's pips. You move onto the board at position 24. You could place your pip on position 24 if you rolled a 1, on 23

if you rolled a 2, on 22 if you rolled a 3, and so on, depending on how many pips occupied those positions. The computer must do the same if you knock its pips onto the Bar. The computer re-enters the board at position 1.

Once all of your pips are on your inner table, you can start moving them off the board. The first player to get all of his or her pips off the board is the winner. Imagine now that there is a position 0; you must move your pips to position 0 to get off the board. You can do this by using an exact roll of the dice, or you can use exact values of the dice to move your pips closer to the board's exit. Using a value larger than what is needed to get your innermost pips off the board is also legal. For example, if you have only two pips left on the board at locations 3 and 5, and you roll a 2 and a 6, you would do the following:

```
FROM—05 TO 00
FROM—03 TO 01
```

You would then have 1 pip left on the board. Notice that the exact value needed to move off the board was entered, even though the value on one die was greater than this value. This is important because the computer keeps score of the remaining number of moves each player will have to make to get all pips off of the board. When one player moves all of his or her pips off of the board, that player's score is 0. The other player loses by the number of moves remaining in his or her score.

"As an opponent, the computer is very aggressive—placing a higher priority on offense than on defense."

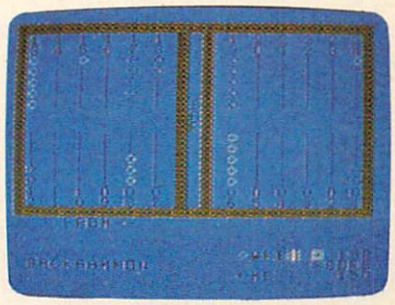
The Program's Logic

As an opponent, the computer is very aggressive—placing a higher priority on *offense* than on *defense*. Ideally, the computer would weigh offensive *against* defensive moves and then choose the best move to suit an overall winning strategy. However, memory considerations dictate that we use a purely offensive strategy, one based on moving pips off the board as rapidly as possible. So, the computer often will pass up chances to bump your pip to the center Bar in favor of moving one of its pips forward. This is literally a measure of its intelligence—the *artificial* intelligence created by the interaction of the computer and the program.

Electronic Backgammon uses the same basic algorithm for all of the machine versions. When a move is indicated, the computer goes through a series of logic steps, searching for the optimum place to move. After rolling its dice, the computer first checks every pip on the board; if it determines that there are no legal moves, it displays a move-blocked message and the opponent gets to move. The computer begins this move sequence by first locating all of the pips that are the farthest from its inner circle. If there are no pips in its home territory, then the program checks each quadrant of the board, looking for the best possible move it can make while protecting its other pips as well as possible. If there is a legal move, the computer first determines whether it can move into the inner circle. It then checks to see whether there is a pip that is open (unprotected) and whether it can move to protect the pip. Next, the program locates the enemy pips on the board and determines whether there is an open pip among them. If there is, it checks whether it can bump the pip from the board to the Bar. If there isn't an open pip, it makes its first move, then rechecks the board and determines whether the pip it just moved is open. If the pip is open and the computer can protect the pip, it will do so on its next move. If all of the pips are in the home territory, the computer goes through another sequence of loops to determine which pip it can remove from the board.

The program checks for all legal moves. It will not let you move from a position you do not occupy. It will not let you move to a position other than forward, toward your inner table; you could not move from 13 to 18 because this would be going away from your inner table. The program will also prevent you from moving off the board if you do not have all of your pips in the inner table. It will also stop you from placing your pip on top of a position that contains two or more computer pips. Likewise, the computer will always make a legal move.

Because of limited memory space, the program cannot have a large set of different moves to choose from, so it chooses the first move that satisfies the conditions that it is working under. The program always does this checking in a fixed pattern, not randomly.



This is a representative screen photo taken from the TI-99/4A version.

For *your* key-in listing see HCM PROGRAM LISTINGS Contents.

The Apple version of *Electronic Backgammon* uses the DRAW command with shape tables to create the pips, and the numbers which indicate the positions of the pips. After a move is registered, the XDRAW command removes the number and redraws the line. With each move, all 24 positions on the board are redrawn with an updated count on each position.

In the Apple version, the computer's pips are open circles, while the user's pips are solid white circles.

The C-64 version of *Electronic Backgammon* uses standard character graphics to create the backgammon board design. When the computer moves one pip, it erases and redraws all 24 pips in their proper positions.

Electronic Backgammon (Apple *II* Family) Explanation of the Program.

Line Nos.	
100-190	Program header.
200-220	Protect hi-res page.
230-290	Initialization and title screen.
300-370	Determine who is first.
380-440	Roll dice and branch.
450-750	Player's move.
760-780	Computer blocked?
790-840	Must computer move from Bar?
850-1970	Computer move logic.
1980-2140	Update computer's pip location.
2150-2210	End-of-game options.
2220-2230	Blocked- and invalid-move messages.
2240-2390	Get player's input.
2400-2910	POKE shape table and set hi-res.
2920-3260	Draw board and move pips.
3270-3290	Display dice roll.

Electronic Backgammon (IBM PC and IBM PCjr) Explanation of the Program.

Line Nos.	
100-230	Program header.
240-260	Initialization and title screen.
270-310	Display playing screen.
320-380	Determine first player.
390-590	Player's turn.
600-670	Is computer blocked?
680-870	Complete computer's choice.
880-890	Update computer's pip location.
900-920	End-of-game routine.
930-1020	Routine to display dice on the board.
1030	Time-delay routine.
1040-1050	Display routines.
1060-1140	Calculate pip positions.
1150-1310	Display pips routine.
1320	Scan keyboard.
1330	Program data.

Electronic Backgammon (C-64) Explanation of the Program.

Line Nos.	
100-190	Program header.
200-300	Initialization and title screen.
310-560	Draw board routine.
570-670	Determine who is first.
680-750	Get player's input.
760-1530	Check movement from Bar.
1540-2770	Main logic to determine computer's moves.
2780-2940	Update computer's pip variables and move.
2950-2980	Determine dice roll.
2990-3120	POKE inputs to screen.
3130-3190	Keyboard-input routines.
3200-3440	Move pieces.
3450-3480	Invalid-move message.
3490-3650	End of game options.
3660-3760	Computer's-move control loop.
3770-3810	Can't-move routine.

Electronic Backgammon (TI-99/4A) Explanation of the Program.

Line Nos.	
100-180	Program header.
190-290	Initialization and title screen.
300-340	Determine who is first.
350-390	Roll dice and branch.
400-640	Draw pieces on board.
650-870	Player's turn.
880-900	Is computer blocked?
910-940	Must computer move from Bar?
950-1660	Complete computer's choice.
1670-1740	Update computer's pip location.
1750-1770	Make move.
1780-1840	End-of-game routine.
1850	Invalid-move subprogram.
1860	Time-delay subprogram.
1870	Can't-move subprogram.
1880	Inner-table-blocked subprogram.

XB *Electronic Backgammon* requires TI Extended BASIC.

The computer's pips in this version are black, open circles while the user's pips are solid white circles. When stacked, the black open pips become blue open circles, and the user's white pips become solid blue.

PC jr

In the IBM version of *Electronic Backgammon*, the algorithm for moving the pips is comparatively simple—and thus very efficient. Instead of redrawing all 24 pips each time one is moved, the program simply erases the pip being moved and redraws it at its new location on the board.

In the IBM version, the computer's pips are red open circles, and the user's pips are brown. When stacked, the computer's and the user's pips become solid shaded. The IBM computers create the pips using the CIRCLE command. When they stack a pip, they draw the shape of the pip with the CIRCLE command, and then fill the circle with the PAINT command.

The TI version of *Electronic Backgammon* is written in Extended BASIC only. It uses only character graphics to create the backgammon board. Every time a pip is moved, the computer redraws all 24 positions on the board.

In this version, the computer's pips are black, and the user's pips are blue. It is possible to stack pips on a line, because the computer just changes the colors of the stacked pips—the user's pips will be a light grey color, and the computer's pips will be a dark blue.

HCM

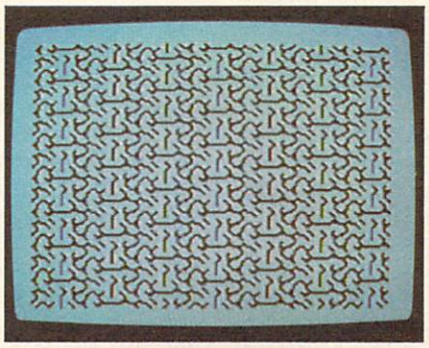
Worm Wood

by William K. Balthrop
HCM Staff

Worm your way into character graphics with this hypnotizing—but easy—BASIC routine.

The design team at Texas Instruments went to great pains to build a computer which is versatile, yet easy to use. The BASIC graphics commands are a case in point. They allow you to build graphics characters, and then place those characters anywhere on the screen. Each character can be placed repeatedly on the screen as many times as you wish. And when you change the graphics pattern for that character, its shape is automatically changed on the screen at each location where the character appears.

By placing characters in a fixed pattern throughout the screen, it is possible to play with their individual graphics patterns to create seemingly random, yet symmetrical, screen patterns. In this program, we have designed several characters with squiggly lines. When placed on the screen, they appear to resemble the inside of an old log which has been half-eaten by worms. Hence the title of our program: *Worm Wood*.



Painting With Characters

You could paint the screen with graphics characters in a number of ways. Perhaps the simplest is just to print them on the screen. This presents a problem, however. In *Worm Wood*, we need to print each line of characters slightly offset from the previous line. The print statement doesn't really do the trick, for even if we place a semicolon after the last item printed, any item longer than the rest of the line will automatically print onto the next line. To avoid this, it is necessary to print one character at a time. We could have placed the characters on the screen with the CALL HCHAR statement, but this would have required an additional loop. Instead, we use just one loop to print 672 characters.

By changing the patterns of the characters on the screen, we can now make the entire screen change. This can often have a hypnotizing effect, as you can see with *Worm Wood*. Let's take a look at the letter A, which appears on the screen quite a few times during this program, though you may not recognize it. This is because the shape of the character you know as A has been changed. Whenever A is placed on the screen, the new shape is used. Now for the best part: If you change the shape of A again, you will see every character A on the screen change to the new shape at the same time. Because of this effect, we are able of to change large parts of the screen in a very short time.

After you have watched the screen wriggle around for awhile, press [ENTER] to advance to the next screen—which is just like the previous one except that the characters are placed at random, with no predetermined order. After each screen begins to change, you can press [ENTER] at any time to restart a new screen.

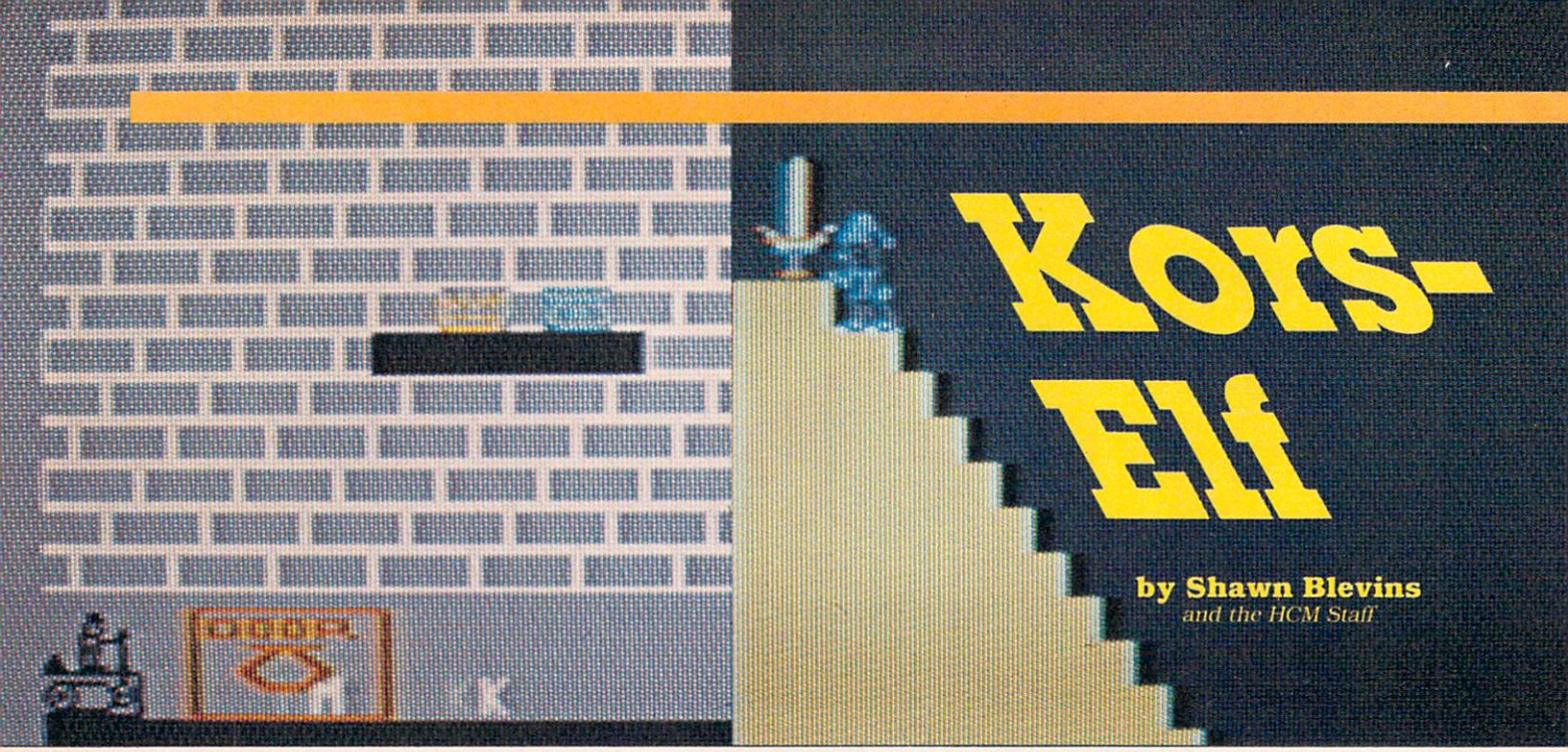
The TI-99/4A home computer is capable of taking you into a graphics wonderland. With only a few basic principles and a little imagination, you'll be able to create your own dazzling displays. Experiment with the program below by changing the patterns used for the graphics characters. Discover the magic that is literally at your fingertips, and have fun.

HCM

For *your* key-in listing see HCM PROGRAM LISTINGS Contents.

Worm Wood (TI-99/4A BASIC) Explanation of the Program

Line Nos.	
100-190	Program header.
200-240	Set up the WORM$() array which contains the graphics shapes.
250-270	Assign graphics patterns to characters.
280-340	Paint the first screen of graphics—symmetrical pattern.
350-400	Constantly change the shapes of the characters—scan keyboard to exit.
410-450	Paint second screen of characters—random patterns.
460-470	Graphics shape data.



Kors-Elf

by Shawn Blevins
and the HCM Staff

Work a little magic with the keyboard as you try to free the Kors-Elfs from the evil Overlord by waking up the letters.

Hail to you, mighty hero of the Kors-Elfs! Only you can save us—and our fallen friends, the Letters—from the spell of the evil Overlord! Our fate is placed in your nimble fingers as you attempt to end the Overlord's domination of our people.

Hordes of letters with glazed alphabet eyes suddenly advance single-file upon you. They've been drained of their will to spell because the evil Overlord has broken their syntactic order. As defender of the elfen race, you must act quickly, or else the wicked prince of letter darkness will ex-spell all words—and eventually elfs—from the land of Kor.

You are the new leader of the Kor-Elfs (not Kors-*Elves*, this is before the modern plural of the word *elf*), a race of creatures in a fantasy land whose survival is threatened by the cruel Overlord of Kor. He rules this land from his castle with the aid of a mystic sword. With this sword, the Overlord has hypnotized and enslaved the long-time friends of the Kors-Elfs, the Letters, and now uses them as guards in his castle. His dispatchers stand ready to release the Letters to attack anyone who invades the castle.

Break the Spell

You have only one weapon—a spell you learned from another race of small, blue creatures (to remain unnamed) many years ago. This spell will awaken the Letters and send them to another world from which you can bring them back once your mission is accomplished. As you might have guessed by now, that mission entails entering the castle, fighting the Letters all the way, in order to steal the mystic sword. Once the sword is stolen, you can use its powers to defeat the Overlord, thereby freeing the Letters and saving your people.

You can stop this character assault if you can type, even if you're a "hunt-n-peck" keyboard whiz. When you correctly type the randomly-displayed Letters that march two-by-two across the screen toward you, you can accomplish your worthy task—awaken the letters from their trance and gain entrance to the inner chamber where the mystic sword is hidden. You, the Kors-Elf, advance from the right side of the screen; and the letters (coming in pairs) advance from the left side. As you get closer to the entrance of the castle, the letters have less distance to travel—which makes it more difficult to type them before they make contact with you. But beware: if you are not fast enough at typing the right letter before it touches you, it will zap you, and you will have to start all over again.

An Eight-Fingered Success Story

Kors-Elf is not just an adventure game, it is a true test of your keyboard mastery—an exciting typing tutor for young children and on the upper levels, a challenging keyboard drill for the experienced typist. There are three levels of difficulty: beginner, intermediate, and expert. On the beginner level, you have five chances to get past the letter guards and find the sword. The two upper levels allow you only three tries, and each one requires considerably more typing speed: the letters are cast upon the walkway more quickly, and typing them to squelch them is more difficult. Once you successfully get past the letter guards and steal the sword, you are given the option to continue on at the same level, or begin again and change the level of difficulty.

For *your* key-in listing see HCM PROGRAM LISTINGS Contents.

The Apple version uses a shape table to define the letters with the DRAW and XDRAW commands. The shape table numbers correspond to all the letters in the alphabet—A is 1, B is 2, etc. The ASCII of the keyboard code is translated into the number of the shape of the letter you enter by subtracting 64 from it. The XDRAW command is used to move the letters and the elf. To determine whether the elf and the letters are touching, the values of the variables (which determine the position of the elf and the letters) are compared. If they are the same, the elf is sent back; if not, he advances.

"Kors-Elf is not just an adventure game, it is a true test of your keyboard mastery— an exciting typing tutor for young children and on the upper levels, a challenging keyboard drill for the experienced typist."

In the Commodore version of *Kors-Elf*, the letters are placed on the screen with character graphics, and the elf is a sprite. The C-64 uses coincidence checking to determine whether the elf is touched by one of the advancing letters. This is done by checking the sprite-to-foreground collision register at memory location 53279 in line 370. If a blank is the only character in contact with the elf, the coincidence flag is not set and the elf continues advancing on the castle. But if the elf comes in contact with one of the letters, he is sent back to the beginning and the number of tries is incremented by 1. When the number of tries completed equals the number of tries for that level, then the game is over.

The IBM version of *Kors-Elf* uses three variables to keep track of the position of the elf and the letters. The variables refer to the screen location of the elf and the two letters that are attacking it. The program compares the location of the elf with the location of the letters to determine whether they are touching. It also checks the letter that you key-in to free the "zombied" letter. A 16-key buffer is cleared after each key press.

The TI version is written in BASIC and character graphics are used for the elf and the advancing letters. Each character used for the letters and the elf is a variable which is checked to determine whether the letter is touching the elf (both variables are the same). If they are, the elf is sent back; if they are not the same, the elf continues on his quest. Because of this, the motion of the letters and the elf will be a little jerky, but the action is still fun. Because there is no keyboard buffer, you have to hold down the key for the character that you want to wake up until you hear a beep. This will tend to make the program respond more slowly.

HCM

Kors-Elf (Apple *II* Family) Explanation of the Program.

Line Nos.	
100-210	Program header.
220-240	Protect hi-res.
250-260	Jump to initialization.
270-560	Main program loop.
570-660	Award sword screen.
670-760	End game; replay.
770-800	Grapic subroutines.
810-870	Music routines.
880-1130	Inititialization subroutines.
1140-1360	Music and graphic data.
1370-1590	Setup subroutines.

Kors-Elf (C-64) Explanation of the Program.

Line Nos.	
100-190	Program header.
200	Jump title screen and initialize.
210-430	Main control loop.
440-470	End of a round.
480-710	Player reached the sword.
720	Player is out of lives.
730-1170	Music routines.
1180-1690	Title screen and initialization.
1700-2080	Program data.
2090-2380	Screen-setup routine.
2390-2670	Program setup.
2680-2730	Set-and-clear-bits routine.

Kors-Elf (IBM PC, IBM PCjr) Explanation of the Program.

Line Nos.	
100-220	Program header.
230-290	Program sequence control.
300-420	Define graphics.
430-960	Display title screen. Get level.
970-1370	Display playing screen.
1380-1700	Routines for Wizard & Elf .
1710-1770	Update score.
1780-1800	Collision.
1810-1920	Initialization.
1930-2140	Release-letter routine.
2150-2270	Graphics subroutines.
2280-2740	End of game. Play again?
2750-2990	Sound-effects routines.

Kors-Elf (TI-99/4A) Explanation of the Program.

Line Nos.	
100-190	Program header.
200-240	Display title screen and initialize.
250-420	Main control loop.
430-490	End of a round.
500-610	Player reached the sword.
620	Player is out of lives.
630-740	Music routines.
750-770	Screen-setup routine.
780-930	Program data.

XB *Kors-Elf* requires TI Extended BASIC.

Personal Loan Calculator

in BASIC for the TI-99/4A

by H. W. Button
and the HCM Staff

Personal Loan Calculator originally appeared in Vol. 4, No. 5 for Apple, Commodore 64, IBM PC and PCjr, and TI-99/4A (Extended BASIC) systems. Here we present a BASIC version for 99/4A computer users with "bare bones" or minimal systems. This is a handy companion program to the *Savings* program published for the TI-99/4A in Vol. 2, No. 6. The two programs together form a comprehensive software package for everyday personal financial decisions. For additional information, please refer to these previous issues.

To the uninitiated, borrowing money can be an intimidating experience. But if you know how much you can afford to borrow by computing the amount and frequency of your payments—toward both the principal and interest—you can feel more confident when taking the loan plunge. And you can compute the relevant figures with our *Loan Calculator* program.

The main menu of the *Loan Calculator* contains 5 options, which may be selected in any order. They are:

1. Payment amount
2. Number of payments
3. Loan amount
   (How much you can afford to borrow.)
4. Amortization Schedule
5. Exit the program

Two questions are initially asked of you in options 1, 2, or 3: the first inquires whether your payments will be made monthly or annually, and the second asks whether the length of the loan period is expressed in months or years. Input from options 1 and 2 will help determine how often interest is compounded. According to the amount of principal still owed on a loan and the way the interest is compounded, the amount of interest due for each payment will change.

Option 1 also asks for your:

Interest rate?
Months (years) of loan?
Amount of Loan?

Option 2 requests your:

Interest rate?
Monthly (annual) payment?
Amount of loan?

Here the program will test to make sure that the payment amount you indicated in option 2 is higher than the interest generated during each payment period. If it is not, you will receive an error message and be returned to the main menu.

Answering any of the questions in option 3 will generate a report consisting of the following information—the data that you have already entered, and the data you wish to know:

Interest rate
Compounded (monthly or annually)
Loan Amount
Payment Amount
Number of payments
Term of the loan
Total interest
Total cost (principal + interest)

Option 4, Amortization Schedule, provides your monthly and final payment amounts and your payment schedule after you enter data for the following:

Loan amount?
Number of monthly payments?
Interest rate?

You will be prompted to enter the starting and ending payments that you want included in the report, which will display:

Payment #
Interest for this payment
Principal for this payment
Loan balance for this payment

When you finish scrolling through the report, the program will return you to the main menu. Pressing (ENTER) when you finish any of the main options will also return you to the main menu.

This *Loan Calculator* program is a handy, flexible tool for anyone who is either considering or already paying off a loan. If you are in one of these positions, this program is "just what the banker ordered." **HCM**

For *your* key-in listings see HCM PROGRAM LISTINGS Contents.

Loan Calculator (TI-99/4A BASIC) Explanation of the Program

Line Nos.	
100-190	Program headers.
200-330	Initialize program and display title screen.
340-440	Input term and expression of loan payments.
450-760	Solve for the amount of the payment.
770-1090	Solve for how much you can borrow.
1100-1500	Solve for number of payments.
1510-1700	Calculate and display report screen.
1710	Halt program.
1720-2160	Routine to calculate and display amortization schedule.

HCM Review Criteria

Each month, *Home Computer Magazine* (*HCM*) reviews products designed for the Apple *II* Family, Commodore 64 and VIC-20, IBM PC and PCjr, and Texas Instruments 99/4A computers. *HCM* reviews take a detailed look at the quality, utility, and value of commercially available packages for these machines. Because our publishing charter forbids accepting outside advertising, we strive to make the scope and content of our review pages shine with a unique blend of humanistic frankness and objectivity.

Not only will you find all relevant information for making a wise purchase decision, but in some special cases we also provide nuggets of compu-prestidigitation.* For example, we frequently include essential documentation not furnished by the manufacturer. Additionally, each issue of *HCM* tries to review at least one outstanding product —a "Diamond in the Rough"—which, because of company size, marketing clout, or for some other reason, has not received the attention it deserves.

At the beginning of each review, a review-at-a-glance box provides the user with an instant assessment of the product. Each item will be evaluated, where relevant, with the criteria below.

HCM Review

Name: Old Art
Program Type: Recycled Graphics
Machine: Apple *II* Family, C-64 & VIC-20, IBM PC & PCjr, TI-99/4A
Distributor: Hit 'n' RUN Software, Inc.
Price: $99.99 (or trade for '72 Pinto)

System Requirements:
Disk Drive, Joystick, Trash Can optional

	Poor	Fair	Good	Excellent
Performance:				
Engrossment:				
Documentation:				

Products may also be evaluated in the following areas:

*** Flexibility—**
Can the product be adapted to the specific needs of the users?

*** Cost/Benefit—**
Is the product worth the user's investment in time and money?

*** Necessity—**
Is the product a solution for which a problem already exists?

*** Originality—**
Is it unique in concept, or simply a "me too" product?

*** Longevity—**
The "Boredom Factor." Does the program sustain interest?

*** Rewards—**
Are the audio-visual rewards motivating and appropriate?

*** Concept Presentation—**
Are the concepts presented clearly, logically, and in depth?

*** Special Effects—**
How does quality of sound and visual effects rate? Do they enhance or detract from the product or learning process?

*** Performance—**
How well the product performs as intended; how well it takes advantage of a specific machine's capabilities; how well it responds to the user's commands; how effectively the graphics, sound effects, music, or speech are integrated with the software.

*** Engrossment—**
Whether the game or activity has that intangible quality that holds players on the edge of their seats while the hours tick by unnoticed.

OR

*** Ease of Use—**
The degree to which a user can interact with the product without outside help; the ease and effectiveness of error-handling features; whether the actual reading level of the activity is appropriate for the suggested audience.

OR

*** Ease of Set-up—**
How well the product design facilitates easy installation.

*** Documentation—**
The quality of the printed matter that comes with the product; whether the instructions are clear and comprehensive; whether the machine configuration requirements are spelled out. Information such as how to load a program, use the keyboard, and restart an activity contributes to the documentation rating, as do tips on performance peculiarities.

Attention Software Authors & Peripheral Inventors:

* WANT TO BE DISCOVERED? *

Home Computer Magazine Wants To Give You A Chance!

We are looking for home computer products that have not received the attention they deserve. Each month, we will be singling out one such package for special review. If you have a unique commercial product of exceptional quality—but your advertising and promotion budget has not allowed you to capture major media attention—we want to see it. We will consider reviewing any product that meets our high standards.

We are an Equal Opportunity Reviewer!

In order to qualify for possible review, your product must:

1. Currently be available for purchase to readers of this magazine.
2. Make a unique and important contribution to the home computer industry.
3. Be of outstanding merit, quality, and value.
4. Be consistent with the type of machines and products we normally cover.

If you feel that your product qualifies, mail it to:

Home Computer Magazine
Attn: Editorial Submissions
1500 Valley River Drive, Suite 250
Eugene, OR. 97401

We reserve the right *not* to reply to each inquiry, so please do *not* contact us except to request return of your product. If you want your product to be returned, please include sufficient return postage.

***Compu-prestidigitation** (kóm´• pū • prĕs´• teh • dĭ • jeh • tā´• shŭn) —*n* 1. The magical quality of unexpected comprehension that results from presenting technical information about computers in a lively, entertaining, visually attractive and easy-to-understand format. 2. The magical tricks that make a computer sing, dance, and do all sorts of wonderfully useful things.

Rocky's Boots and Robot Odyssey I

A dual review by
Steve Nelson
HCM Staff

Explore the world of electricity and machines. Program your own robots, and even design your own microworlds and games with these two trend-setting programs.

Each of us exists in a world that is different from the world of the person standing next to us in line at the grocery store—or wherever we may be at a given moment in time. Yet, at the same time, our individual worlds also conform to a larger, more common world. Once you grasp this concept, then you also realize that there are worlds within worlds within worlds. A drop of water in a pond is a tiny microcosm boiling with frantic life. A yard square section of desert is a unique world separate from its counterpart in the rain forests of New Guinea. Each world has a logical and unyielding structure by which the creatures living in them must follow in order to exist. If you are born into this world, you either arrive preprogrammed by nature to survive, or you have a support system—a family that protects you and teaches you until you can function on your own.

This concept of worlds within worlds applies nicely to the microworlds you enter with The Learning Company's two educational games, *Rocky's Boots* and *Robot Odyssey I*.

Microworlds

Your guide to these magical worlds is the cursor—it has the fantastic ability to become several different things. It is how you travel around inside the tutorials, discover Rocky's machines, check out the Innovation Lab, and finally, enter Robotropolis. With it you can explore the wondrous inner workings of machines and robots while learning about all of their different gadgets and gizmos. The cursor can also become many other things, as you will see in the tutorials.

In *Rocky's Boots*, you must learn the basics of machine design. The game has 6 sections; 4 of these are tutorials explaining the basics of circuitry, how electricity works, and how logic gates function. The other 2 have to do with the games Rocky's Boots and Rocky's Challenge. In this program you must build a machine with the parts supplied on the screen—one that directs a "boot" to kick specific objects as detected by sensors.

Rocky's Boots begins by teaching you how to move in this new world. You are the cursor, a white square which you can move, left, right, up, or down. You travel through a series of rooms, pausing to read instructions and practice the new skills you are learning.

The first tutorial shows you how to construct simple machines using parts supplied on screen. Here you will learn about inputs and outputs, and how to connect them. In both games, the flow of electricity is easily observed because the wires connecting the circuits to the machines glow orange when on, and white when off. Both games require a color monitor.

Logic Gates

In the second tutorial, you are shown the differences between an AND gate, a NOT gate, and an OR gate. These logic gates, which act just like the circuits in your computer, help you control the electricity flow to the machines. A secret room in the logic gates tutorial conceals an alligator. This alligator will attack your cursor as you try to build your machine. Fortunately, you can hook up an alligator sensor to a boxing glove and use it to punch out the alligator—effectively keeping it at bay while you obtain the parts you need.

You are now ready for Rocky's Boots. This part of the package is actually the third tutorial. Here you can put everything you have learned into the actual building of a machine that will complete a specific task. The task you must complete here is to design a machine that uses Rocky's boot (literally a blue boot) that can be connected to "kick" specific objects that pass in front of a sensor. The objective of this tutorial is to teach you how the machine in Rocky's Boots operates, and to give you practice-time building machines. If you successfully kick out all of the objects and score 24 points, Rocky comes out and does a little dance for you.

The final tutorial is on flip-flops, clocks, and delays—all devices that you will need in order to solve some of the more complicated problems in the final section of the game, Rocky's Challenge. Here you put all the new information that you have learned to the test. There are 32 games for you to play, and you even have the option to create your own game based on the above format.

Rocky's Boots was created for ages 9 and up—but I don't think that it would keep a teenager interested very long. It's true that some of the 32 games are

quite difficult and require some concentrated thinking in order to solve them, but they are still just variations on the same game. Any child who knows how to read should be able to use this program. For older children the game needs to be spiced up a little more before it will hold their interest for any length of time.

	Robot Odyssey I
Name:	Robot Odyssey I
Program Type:	Educational
Machines:	Apple *IIe*, *IIc*, *II+*
Distributor:	The Learning Company 545 Middlefield Rd. Ste. 170 Menlo Park, CA 94025
Price:	$49.95
System Requirements:	64K, color monitor recommended
	Poor Fair Good Excellent
Ease of Use:	
Engrossment:	
Documentation:	

	Rocky's Boots
Name:	Rocky's Boots
Program Type:	Educational
Machines:	Apple *II+*, *IIe*, *IIc*, IBM PC & PCjr
Distributor:	The Learning Company 545 Middlefield Rd. Ste. 170 Menlo Park, CA 94025
Price:	$49.95
System Requirements:	64K, color monitor recommended
	Poor Fair Good Excellent
Ease of Use:	
Engrossment:	
Documentation:	

Robot Construction

Robot Odyssey I is an extension of the principles taught in *Rocky's Boots*. The same elements are used (flip-flops, gates, inputs and outputs) along with several new ones, but they are used on much more complicated and interesting machines—robots. Your job is not only to figure out how to make them operate, but you will also have to program them to work in such a manner as to retrieve objects, avoid dangers, and even ride the subway!

Robot Odyssey I teaches you how to design and program robots to perform specific tasks that will let you escape from the underground city of Robotropolis. In doing so, you learn some of the basic fundamentals of engineering, digital logic, and circuit design. You also learn some useful problem-solving skills, such as breaking one large task into smaller individual tasks, or visualizing a solution to a problem and then generating a hypothesis and testing it to see if it works.

Within this world there is really two worlds: Robotropolis, and a place called Innovation Lab—a place where you can design, program, and test your own robots. In addition, there are three tutorials that teach you the basics of how these robots work.

"Your guide to these magical worlds is the cursor—it has the fantastic ability to become several different things."

Basically, each robot consists of a little cube (kind of like a small dirigible) carrying special internal sensors or other tools which, when the sensors are stimulated, will activate tools or thrusters according to the design of the robot's circuits. Sensors are of several types, and can detect objects on contact, in the same room, or in other special situations. A robot's tools include a mechanical arm that can be used to grip objects, an antenna for communicating with other robots, and a periscope-like device that lets you see where the robot is going while you are riding inside of it. You can wire the circuits in any configuration that you like in order to make the robot move in the direction you wish or react appropriately to certain stimuli.

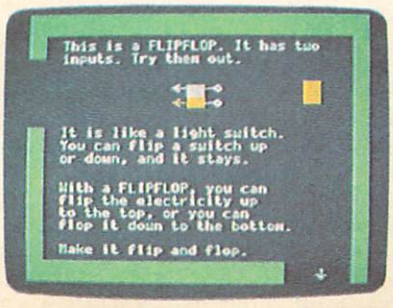
Robotropolis, like the rooms in the lab and the tutorials, is a labyrinth through which you must maneuver using a cursor or a robot. For example, when a robot's contact sensor—which is wired to a thruster—touches a wall, it sends electricity to the thruster, causing the robot to move. The trick is to wire the thrusters to the sensors in such a way as to cause the robot to move in the direction that you want.

Back to School

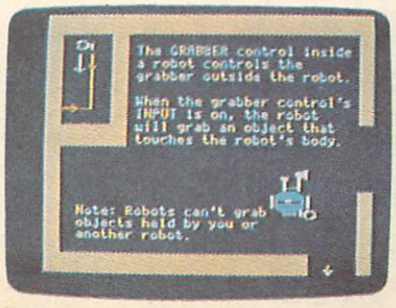
Odyssey's three tutorials take you step-by-step through the inner workings of robots, and they do it in a unique fashion. As the cursor, you are able to lock onto a robot or a part for a robot and carry it around with you. Or, you can take your cursor inside a robot and actually manipulate the parts that make it go. (You can even take robots inside other robots!)

The first tutorial is called Robot Anatomy. Here you are taught how robots behave, what makes them react in certain ways, and how to change their behavior. Outwardly, each robot is essentially the same as the next. But inwardly, you can design the robot to do different things like pick up objects, maneuver through mazes, or even dance with a partner. Through this process of discovering how robots work, you will be laying the foundation for future explorations in Robotropolis. What you learn here in Robot Anatomy, and in other tutorials as well, is crucial to your success in Robotropolis.

The second tutorial is called Toolkit. As I mentioned earlier, the cursor can be changed into the toolkit when needed for quick programming changes or experimentation in the Innovation Lab. In this tutorial, you learn about simple circuits, gates, flip-flops, and



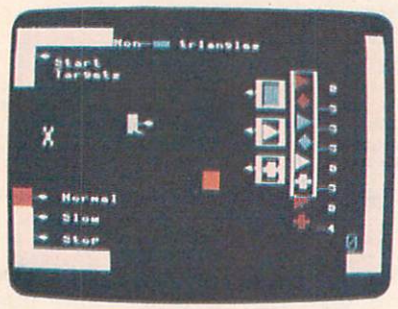
This screen is from the flip-flop tutorial in Rocky's Boots explaining what a flip-flop is.



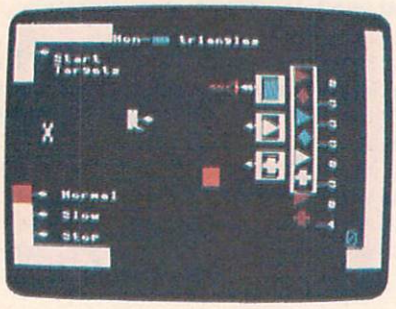
This screen is from the Robot Anatomy tutorial in Robot Odyssey I that explains about the robot's grabber control.

Building a Machine

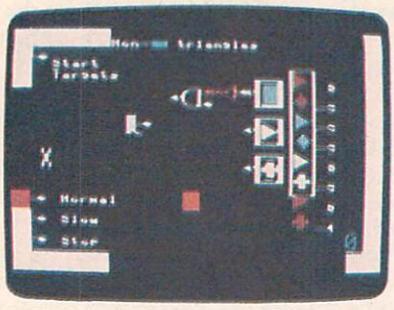
This series of screen photos from Rocky's Boot's shows the step-by-step construction of a machine designed to kick out non-green triangles.



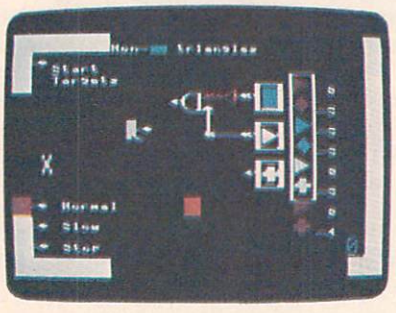
The basic machine in Rocky's Boots before modification.



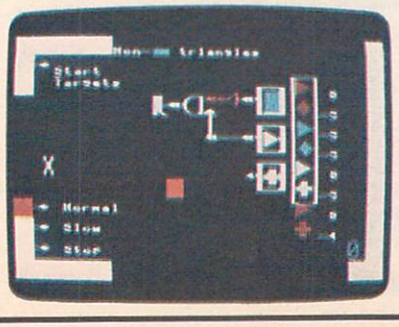
Adding a NOT gate which prevents the sensor from indicating a green triangle.



Adding an AND gate which allows the machine to detect non-green triangles.



Adding two wires completes the circuit connecting the sensors.



Attaching Rocky's Boot finishes the machine. It is now ready to play the game.

nodes, and about how they affect the performance of a robot. The toolkit actually carries around with it a complete set of circuits, logic gates, nodes, flip-flops, and other equipment necessary for robot programming. This makes it possible to do reprogramming in the field.

The final tutorial is called Chip Design. Here you can learn how to wire-up chips to perform different tasks that help you find your way through Robotropolis. It will take a little experimentation before you will figure out this section, but it will be useful to know before you go to town—wiring together all the doodads and gizmos inside the robot can get pretty confusing. You can make things a lot easier if you pre-program a chip to instruct the robot to do the same thing. This tutorial will teach you how to design a chip to make a robot perform a specific task. As chips are much smaller than robots, you will be working with an enlarged chip, actually going inside of one with the cursor, bringing with you the circuits, gates, and other items necessary to complete the job.

"These games emphasize thinking for yourself—reasoning out and understanding a problem before you tackle it."

If you've tried all of the tutorials, you are now ready to investigate Innovation Lab. This is a complete world all its own. Inside it, you can practice designing and programming robots and polish your skills before attempting to find your way out of Robotropolis. A chip facility is also located here and when you get a chip programmed the way you want it, you can take it to the burner room (a special room where your large prototype chip is transferred to a smaller chip) where a copy can be made which you can save to a separate disk and use later in Robotropolis.

City Sewer

Robotropolis is a separate world from the tutorials and the lab. It is a real "microworld" with its own logic and set of rules which—in order to succeed or even survive—you must discover and follow. You begin your journey after dreaming (or were you awake?) that you fell out of bed and landed in Robotropolis' city sewer. From here, you must negotiate 5 levels of the city and avoid ampere 'bots, ride the subway, find your way through the town maze, free the master robot, and ride a giant disk drive (among other things) in order to escape. The only way you can do all this is to use robots to help. Three robots are in the sewer, and you can use them to help you find your way out. Each one is programmed to do different tasks, but they won't be able to help you escape unless you reprogram them as you find new dangers to avoid. Sounds pretty complicated doesn't it? Well it can get to be a real head-scratcher before you finally get out—if you ever do . . .

What is notable about *Rocky's Boots* and *Robot Odyssey I* is the way they use tutorials to build up your skills before you travel into Robotropolis or play Rocky's Challenge. In these worlds, you must find out what the rules are before you can play the game. Fortunately, the tutorials are interesting and grow in difficulty as you progress through them. Both programs require the persons playing them to follow a structured path in order to gain the skills needed to successfully complete the game. In doing this, they learn deductive reasoning, how to plan out a problem beforehand,

and how to debug their solutions. Some of the more complicated games can require extensive wiring, and even the best plans may not work the first time through.

Problem-Solving Skills

These games emphasize thinking for yourself—reasoning out and understanding a problem before you tackle it. Lots of puzzles wait to be solved in Robotropolis and most of them come with no explanation, leaving it up to you to figure out the best way to handle them. The tutorials teach you the skills you need to solve these puzzles, but they don't show you how. This can become quite frustrating until you actually do figure out the solution. This technique of giving you problems to solve with little if any explanation is an important part of *Robot Odyssey I* (I didn't run into this very often while playing *Rocky's Boots*). To add to the mystery, there are several little creatures that you will notice as you travel through Robotropolis that don't seem to have any function at all, except to bounce up and down or flutter about as you travel past them.

Response to either the keyboard or a joystick is excellent in both games. You have total control of cursor movement with no delays or problems. The machines and robots respond particularly well to your programmed input. And, as you get more skilled at designing circuits for the machines and robots, you can make them as complex as possible.

Rocky's Boots and *Robot Odyssey I* are available for the Apple *II* family of computers including the Apple *IIc*. *Rocky's Boots* is also available for the IBM PC and PCjr with no appreciable difference between versions.

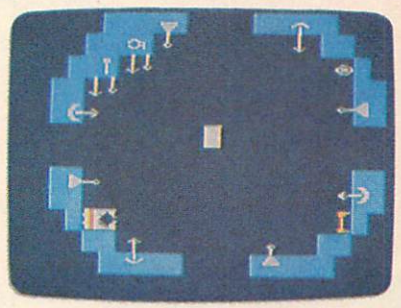
Both programs come with good documentation which gives you a general understanding of how each part of the program works. In both cases, the tutorials take up the bulk of teaching you about the microworlds you will be traveling through and how to handle various problems you will be facing. However, in *Robot Odyssey I*, there are several things that aren't explained in the manual and that I didn't have time to investigate thoroughly. Believe me, it could take months to travel through the entire city of Robotropolis.

In general, my reaction to these packages is very positive. *Rocky's Boots* is a good starter package that may be useful in preparing younger children for the more sophisticated *Robot Odyssey I*, but it is fairly limited in complexity—and if your child is quick to learn, he or she will probably become bored with it relatively soon. *Robot Odyssey I* is simple enough in the beginning for younger children (even smart 9 year olds), but it can become as complicated as you want it to be. The beauty of both of these programs is that you are allowed to progress at your own pace. If you don't understand something, you can always go back and review it. Even while deep inside Robotropolis you can save the game on disk and return to the Innovation Lab or the tutorials.

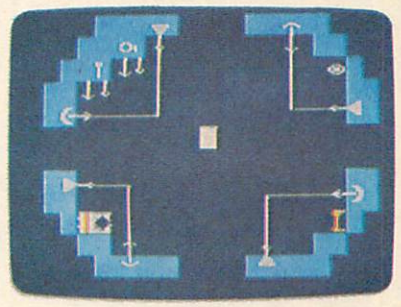
I find it very encouraging to see educational games like these that stress basic learning skills and that are presented in non-violent and positive learning situations. *Rocky's Boots* and *Robot Odyssey I* are both excellent examples of educational games parents can purchase with confidence, knowing that their children will indeed learn useful skills and be entertained at the same time. The worlds they explore here are limited only by a child's imagination and experiences, both of which will grow and benefit from interacting with these games. **HCM**

Programming a Robot

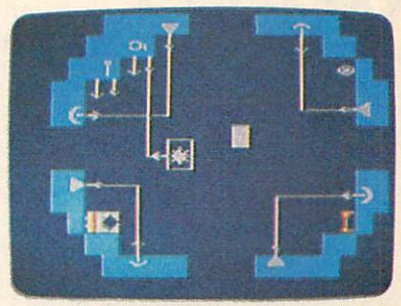
This series of screen photos from Robot Odyssey I shows the construction of a Robot designed to move through a maze and grab objects.



Interior of an unprogrammed robot showing the input and output connections.

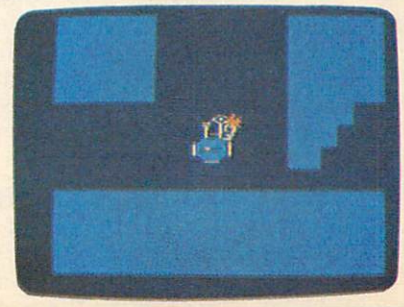


Connecting the thrusters to the bumpers allows the robot to move in any direction in the maze.



Wiring the robot's grabber control to an energy crystal.

Maneuvering the robot through the maze.



This screen shows the robot holding the energy crystal in its grabber arm.

Apple Seedlings

by Anders Nereim
and the HCM Staff

Want to date your ProDOS files, but don't have a calendar/clock card? Enter the Apple Dating Game with this BASIC utility.

When I received ProDOS in a bundle with the Apple *IIc*, I found little information describing its advantages over DOS 3.3. It was quite a while before I discovered some of the most useful features of ProDOS. For instance, the Date function which dates files as they are modified is just what we in the business world are looking for.

As an architect, I had bought the *IIc* mainly because of its trim appearance. I couldn't believe that I really needed clock hardware in order to date a file, and the thought of yet another peripheral besides the power supply offended my desire for a tidy workplace. So I wrote this BASIC utility that sets the ProDOS date. If you have a clock/calendar card, this utility will *not* reset the time; you need to use the software included with your own card. However, if you don't want to buy a clock/calendar card, but still want to be able to set the date for ProDOS, this program is all you need.

It Wasn't Provided—So I Wrote It

When I realized that none of the programs on my UTILITIES disk that came with the Apple *IIc* accessed the Date function, I decided to write my own BASIC utility. The location and configuration of the date is contained in two bytes of the ProDOS global variable page: $BF90 and $BF91. To conserve memory, the Apple engineers divided these two bytes into three groups of 7, 4, and 5 bits, holding the year, month, and day respectively. I could see that setting the date would require some decimal-to-binary conversion. Undaunted, I went ahead and found that I could even make it friendly to an inexperienced user.

From Keyboard Input to POKE Values

The input routine is compact and has enough error checking to keep totally-bogus dates from being entered. After you type in two-digit numbers for each month, day, and year, you're given the opportunity to accept or reject what has been typed by entering either a Y or an N. If you do make an error, just type in any date and press N to try again.

Once the program places the date in the variables MNTH, DAY, and YR, the actual number-crunching begins. Lines 420-460 divide the decimal month by decreasing decimal powers of 2. A FOR-NEXT loop stores the results of each division in the integer array MNTH%( ).

Lines 470-490 begin to manipulate the day and part of the month into a "byte" size piece. Lines 500-530 convert the decimal year to binary in the array YEAR%( ), using the same kind of backstepping FOR-NEXT loop that you used in lines 420-460.

Key in the program as shown in the listing, and save it on a disk by typing SAVE SET.DATE. Type RUN and you will get the date-entry template on the screen. Enter six numerals corresponding to the month, day, and year without any slashes or dashes, because these are supplied by the program.

If you set the date with this program and then open a new file, the system date will appear in both the DATE MODIFIED and DATE CREATED columns when you use the CATALOG command. If you write to an existing file, the DATE MODIFIED column will contain the system date at the time the file was changed.

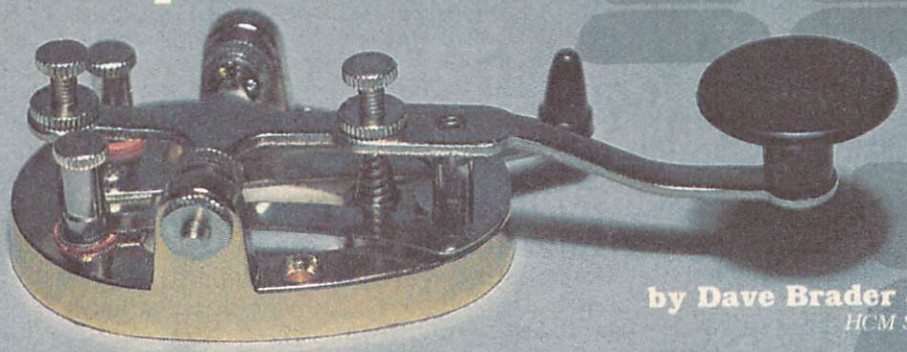
HCM

For *your* key-in listing see HCM PROGRAM LISTINGS Contents.

***Set Date* (Apple *II* Family—ProDOS only)**
Explanation of the Program

Line Nos.	
100-190	Program Header.
200-410	Input routine.
420-570	Convert input and POKE.

Computer Links to Amateur Radio



by Dave Brader & Steve Nelson
HCM Staff

Out with the old and in with the new. Computer/ham radio links offer exciting possibilities in data transfer and communication.

A Dash Of History

Amateur radio operators have been around since Marconi discovered the wireless. As radio grew, hundreds of private citizens began experimenting with sending messages. These early amateur radio operators (affectionately referred to as hams) first started sending messages locally, but as they experimented, they were soon able to send and receive messages at greater distances. At first, operators had to relay messages in order to send them over large distances because the range of the early short-wave radios was limited.

By 1914, when a man named Hiram Maxim founded the American Radio Relay League (ARRL), amateur radio had grown into a popular hobby. The purpose of the ARRL was to unify and organize amateur radio operators and regulate the distribution of the airwaves. But, in 1917, the United States went to war. Most of the amateur radio operaters served in the armed forces, and the ones that didn't were out of luck because the government soon banned amateur radio on the airways.

When World War 1 ended, the ban on amateur radio continued until the ARRL convinced the government to drop it in 1919. After the war, the ranks of the amateurs swelled, and the distances that messages were transmitted increased. By 1924, amateur radio operators were sending and receiving messages across the Atlantic and around the world. Ham radio operators even have their own satellite—OSCAR for "Orbital Satellite Carrying Amateur Radio."

There are now more than 250,000 amateur radio operators in the United States, and many more around the world. These hobbyists perform valuable services in times of disaster or national emergency, working hand in hand with the authorities to keep vital communication lines open.

Although computer hobbyists didn't really increase to significant numbers until about ten years ago, private use of two-way radio communications has been around since before 1914, when the American Radio Relay League was first formed by Hiram Maxim. Radio hobbyists, as some of you may know, are called amateur radio operators (or hams) and must be licensed by the Federal Communications Commission to transmit in this country, and by other government agencies to transmit in foreign countries. To obtain an amateur radio license, each one has had to pass exhaustive technical tests, and demonstrate proficiency in Morse code. Currently, there are just under a million amateur radio operators world-wide.

"With this new equipment connecting the computer to the amateur radio station, the old brass Morse code key is truly obsolete."

The Old and the New

Modern electronic technology in the United States and elsewhere has been boosted over the years by these amateur tinkerers. Ham radio enthusiasts, like computer hackers, are always fiddling with their equipment, trying new things—inventing, innovating and patching new gadgets into their systems.

An interesting combination of these two stable forms of electronic technology is now taking place. This new combination (home computers and amateur radio) is opening the doors to a new frontier—that of instantaneous, world-wide, low-cost computer networking. This union is made possible by a radio transceiver-to-computer terminal which, with appropriate software, allows you to turn your CW station into a completely "CW/RTTY/ASCII" station.

Continued

Morse Code

One of the original ways of communicating using amateur radio was, and still is, via a single tone that is interrupted at different rates to produce Morse code (named after Samuel Morse). This code was originally transmitted using a device called a Morse code key, which allowed the operator to tap out the signal through the transmitter. But with this new equipment connecting the computer to the amateur radio station, the old brass Morse code key is truly obsolete. Now, instead of tapping out the code, the operator merely types the words on the computer keyboard. The computer then generates the Morse code for each of the characters typed and sends it to the transmitter for broadcast via Continuous Wave transmission (CW). Other radio operators can then receive this signal, and either convert it by ear or—if they are equipped with another computer/radio link—use their equipment to translate the Morse code to text on screen. This method of using the computer/radio is probably most interesting to the amateur radio operators themselves. Computer hobbyists may be more interested in just getting information from point A to point B without any errors.

Combining a computer with amateur radio in such a way could mean that hearing-impaired hobbyists will now be able to "get on the air," because they will be able to see the transmitted words on their computer screen. In some cases, this text can be saved to cassette or disk—depending on the configuration—and even printed out on a printer that is connected to the system.

"Imagine, if you will, sitting down at your 'rig,' typing in a letter on your computer, giving a couple of commands, and in a few moments receiving acknowledgement of your letter's receipt from the other side of the world."

Low-Cost Data Transmission

Error-free transmission of computer-generated signals is another unique area in the computer/ham connection. While experimenting with their computers, amateur radio operators have devised various methods of transmitting information during poor broadcasting conditions so that an error-free message is received on the other end. Obviously this is important if you are trying to send someone a computer program. (Have you ever tried to run a program that had a crucial comma out of place?) One of these methods of transmission is entitled AMTOR (Amateur Teleprinting Over Radio), which is a microprocessor-controlled, error-correcting data communications protocol. Another method is "packet" radio, where data is transmitted using ASCII or baudot codes on a point-to-point basis, similar to a telephone modem. The transmission is broken into pieces called packets, which are checked for errors by the computer.

If you want to experiment with ham data transmission, there is one possibilty we have not yet mentioned: phone patching. Those of you who already "network" over the telephone lines probably pay the price in higher phone bills. But a connection can be made from your computer over a modem to a ham transceiver with just a local call. Then, whatever you send to this local ham can be transmitted to a distant ham station and —through another phone patch—relayed to another computer in your network, thus saving you a long distance phone call.

Electronic Mail

Imagine, if you will, sitting down at your "rig," typing in a letter on your computer, giving a couple of commands, and in a few moments receiving acknowledgement of your letter's receipt from the other side of the world. What we are imagining is not yet taking place on a minute-to-minute basis—perhaps at this time only once or twice a week. But messages are flying around in text form in the airwaves over shorter distances—say across the United States—many hundreds of times per day. So, you might ask, what is the application to me? With just one willing ham operator, your club now becomes a node in this potential network of clubs across the country—even the world. With preset operation of your club's computer radio station you will be able to communicate and pass messages, club newsletters, or programs that are in the public domain, all for just the cost of the amateur radio operator's time, the electrical power, and the initial cost of some special hardware and software to link the computer to the amateur radio transmitter/receiver. Just as with "land line" communication networks, your computer can be left on in the receive mode while you are away, and as other computer/radio systems transmit to your station, it will receive and record automatically.

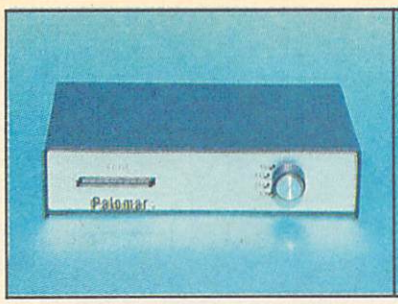
To help get you started in your own club, or if you are an amateur radio operator *and* a computer hobbyist, we have included a chart that shows a few of the hardware "computer link" packages available for all of the computers that we cover in *Home Computer Magazine* —along with a summary of their capabilities and the costs involved. Bear in mind, of course, that the cost of the radio equipment is not included in this chart. Also be aware that before this equipment can be used to transmit information on the air, either you, a member of your club, or a willing friend must be an amateur operator duly licensed by the Federal Communications Commission (in the United States). Please note that a license is not required to *receive* information from amateur radio airwaves. This means anyone can buy a shortwave receiver that covers the proper frequencies, one of the interfaces to the computer, and the supporting software and receive text right out of the air—without a modem, without a telephone.

What kind of text, you ask? How about the Reuter's News Service; or perhaps you'd like to receive the news coming out of Iran in print form. Radio Iran sends out radio teletype signals with their news, as does Great Britain, Russia, and many other foreign countries (wire services in the United States, such as the Associated Press, encode their signals because they charge their customers for receipt; therefore, amateur stations are not able to pick up the text and unscramble it).

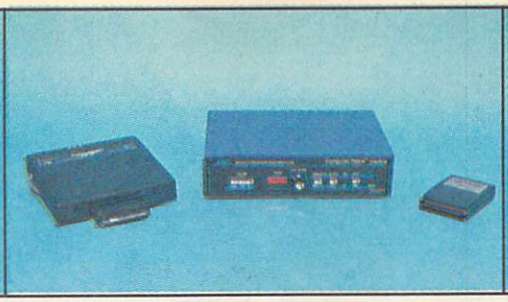
The ARRL sends a news bulletin over the airways 3 times a day, Monday through Friday, reporting on FCC actions, ham-fests, satellite positions, and other information of general interest to amatuer radio operators.

We feel this area of computer/amateur radio has a great deal of potential. Let's hear from those out there who have experience in this field. Send in some letters to the editor that we can share with the rest of our readers—or send us a review of one of these products for possible publication.

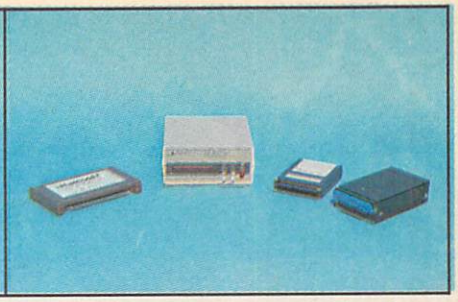
HCM



The CI-103 Interface from Palomar. This unit works with the Kantronics Hamsoft and Hamtext software.



The CP-1 Interface, MAP-64/2 micropatch, and MBA-TOR C-64 from Advanced Electronics Applications.



The Interface, Hamtext C-64, and Hamsoft for the TI-99/4A and the VIC-20 from Kantronics.

A Partial List of the Available Hardware and Software Computer Links for your Amateur Radio

Advanced Electronic Applications
P.O. Box C-2160
Lynnwood, WA. 98036
(206) 775-7373

IBM/Computer Patch Interface

Requires RS232-2 option for CP-I, cost $39.95. Requires RS232 cable to IBM serial port.

Hardware needed: The Advanced Electronics Applications model CP-1 computer interface priced at $239.95.

Software needed: CP-1/IBM-PC by Hamcom, cost $59.95.

Capabilities: Send/receive Morse code from any speed up to 100 words per minute; send/receive radio teletype, 60-100 words per minute; send/receive ASCII, 110-300 or any other baud rate allowed by PC-DOS.

Apple/Computer Patch Interface

Hardware needed: The Advanced Electronics Applications model CP-1 priced at $239.95.

Software needed: CP-1/Apple-1, cost $39.95.

Capabilities: Send/receive Morse code at variable words per minute; send/receive radio teletype at variable words per minute; send/receive ASCII at 110 and 300 baud.

VIC-20/Computer Patch Interface

Hardware needed: The Advanced Electronics Applications model CP-1 priced at $239.95.

Software needed: MBA-TOR/20, cost $119.95.

Capabilities: Send/receive Morse code up to 99 words per minute; send/receive radio teletype from 60-100 words per minute; send/receive AMTOR (Amateur Teleprinting Over Radio).

C-64/Computer Patch Interface

Software needed: MBA-TOR/64, cost $119.95.

Capabilities: Send/receive Morse code up to 99 words per minute; send/receive radio teletype from 60-100 words per minute; send/receive AMTOR (Amateur Teleprinting Over Radio).

VIC-20/Micropatch MAP-20

Software is built in—no added cost.

Capabilities: As it turns out, this option is not available at this time.

C-64/Micropatch MAP-64

Hardware needed: The Advanced Electronics Applications model CP-1 priced at $239.95.

Software needed: none—included with package. Additional cost—none.

Capabilities: Send/receive Morse code up to 99 words per minute; send/receive radio teletype from 60-100 words per minute; send/receive AMTOR (Amateur Teleprinting Over Radio).

Kantronics
1202 E. 23rd st.
Lawrence, Kansas 66044
(913) 842-7745

TI/Kantronics

Hardware needed: The Kantronics interface priced at $269.95.

Software needed: The Hamsoft for the TI-99 from Kantronics priced at $99.95.

Capabilities: Send/receive Morse code, 5-99 words per minute; send/receive radio teletype, 60-100 words per minute; send/receive ASCII, 110-300 baud; parallel printer interface accessible from TI BASIC.

C-64 Kantronics

Hardware needed: The Kantronics interface priced at $269.95.

Software needed: The Hamtext for the C-64 priced at $89.95.

Capabilities: Send/receive Morse code, 5-99 words per minute; send/receive radio teletype, 60-100 words per minute' send/receive ASCII, 110-300 baud; VIC serial printer compatibility.

VIC-20/Kantronics

Hardware needed: The Kantronics interface priced at $269.95.

Software needed: VIC-20 priced at $49.95.

Capabilities: Send/receive Morse code, 5-99 words per minute; send/receive radio teletype, 60-100 words per minut; send/receive ASCII, 110-300 baud; serial or parallel printer support.

Apple/Kantronics

Hardware needed: The Kantronics interface priced at $269.95.

Software package needed: Hamsoft Apple (disk), priced at $19.95.

Capabilities: Send/receive Morse code, 5-99 words per minute; send/receive radio teletype, 60-100 words per minute; send/receive ASCII, 110-300 baud; support for parallel printer.

Palomar
P.O. Box 1924-FW. Mission Rd.
Escondido, CA. 92025
(619) 747-3343

TI/Palomar

Hardware needed: The Palomar CI-103 priced at $139.95.

Software needed: The Hamsoft for the TI-99 from Kantronics priced at $99.95.

Capabilities: Send/receive Morse code, 5-99 words per minute; send/receive radio teletype, 60-100 words per minute; send/receive ASCII, 110-300 baud; parallel printer interface accessible from TI BASIC.

C-64/Palomar

Hardware needed: The Palomar CI-103 priced at $139.95.

Software needed: Hamtext C-64, priced at $89.95.

Capabilities: Send/receive Morse code, 5-99 words per minute; send/receive radio teletype, 60-100 words per minute; send/receive ASCII, 110-300 baud; VIC serial printer compatibility.

VIC-20/Palomar

Hardware needed: The Palomar CI-103 priced at $139.95.

Software needed: Hamsoft VIC-20 priced at $49.95.

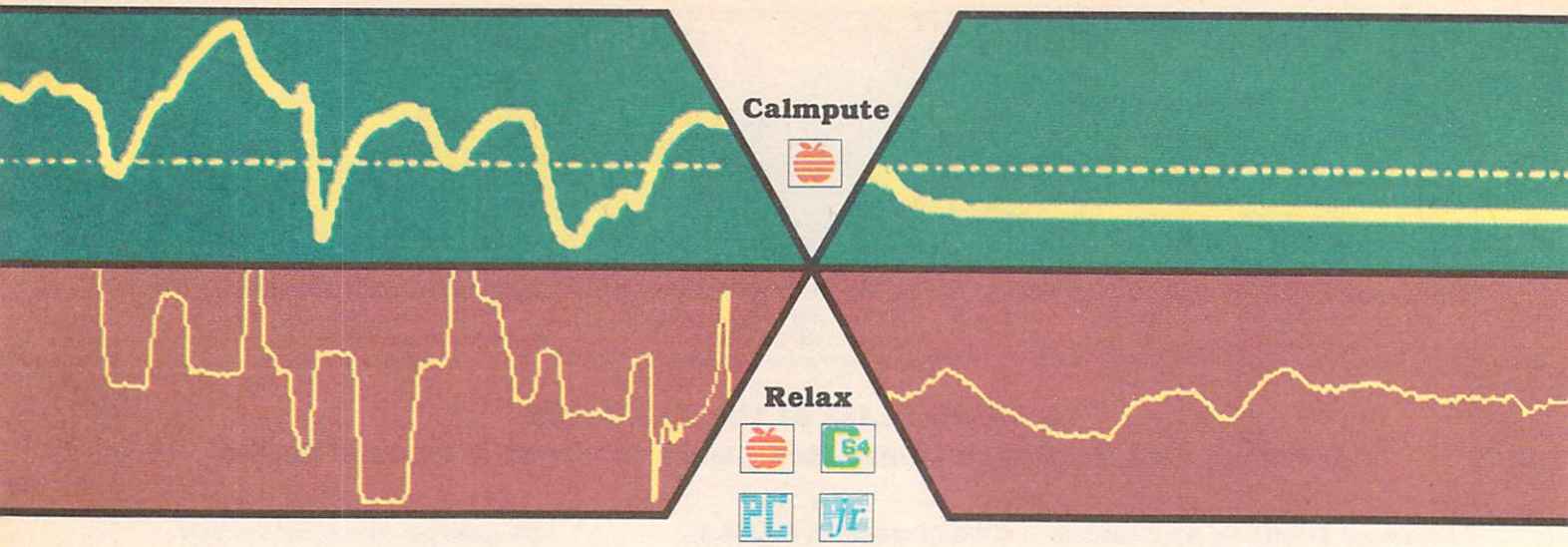
Capabilities: Send/receive Morse code, 5-99 words per minute; send/receive radio teletype, 60-100 words per minute; send/receive ASCII, 110-300 baud; serial or parallel printer support.

Apple/Palomar

Hardware needed: The Palomar CI-103 priced at $139.95.

Software package needed: Hamsoft Apple (disk), priced at $29.95.

Capabilities: Send/receive Morse code, 5-99 words per minute; send/receive radio teletype, 60-100 words per minute; send/receive ASCII, 110-300 baud; support for parallel printer.



The Biofeedback/Stress-Reduction Connection

A Review of Calmpute and Relax

by Dana M. Campbell
HCM Staff

Psychiatrists and masseurs beware—these two stress-reducing programs are moving in on your tension-untightening turf.

Sitting cross-legged with intense concentration, you strive to take control of your body's reactions to the pressures of everyday life. One by one, the muscles in your shoulders, neck, and head relax as you become peaceful and calm. The headache that no amount of aspirin could dispel is now magically gone, thanks to . . .yoga? TM? No, it's *biofeedback*—the self-help aid of the '80's souped-up with home computer technology.

Much of the media attention recently given to the importance *stress factors* play in a person's life—such as job pressures, physical and lifestyle changes, interpersonal relationships, etc.—is now shifting toward *stress management*. At the same time, biofeedback training as a way of coping with stress has undergone a shift from the halls of medicine and academia to the home. According to some, the home computer can now handle the stress measurement and reduction functions that doctors together with large, complicated machines once performed. The *Relax* and *Calmpute* stress-reduction programs demonstrate two very different methods of home-brewed biofeedback. The question is, do they work?

An Electromyograph

Relax employs an electromyograph as its "stress-measuring" device. Three small metal sensors imbedded in a headband press against your forehead and measure the amount of electrical discharge in the skeletal muscle fibers. When facial muscles contract, they produce electrical activity that the sensors pick up and transmit to the EMG unit, which then translates this information for the computer. Like *Calmpute*, *Relax*'s objective is to make you aware of your physiological expressions of stress and teach you how to recognize and control them.

Relax offers three options in its program for achieving this goal, and numerous others in its comprehensive guidebook. The first program option, the Relax Graph, displays a needle-like trace on a graph numbered from 350 to 500 units along the left side. This is the upper portion of a scale that measures relative muscle tension and relaxation. The number of electrical discharge readings taken off the headband per second determine how many interval marks are created along the top and bottom of the graph. Sample rates range from .25 to 60 readings per second—the lower the rate, the longer the period of time your tension level is measured for each sample.

Continued observance of the immediate effects that your tension level has on the needle is supposed to increase your awareness of body tension, which then makes it possible to concentrate on relaxing and lowering the needle on the scale. This scale is the only quantitative way offered here to measure how much you can relax during a session with the program.

The Sensoral Kaleidoscope is a series of symmetrical patterns that slowly change colors, merge, and separate according to your tension level. As you become relaxed, the bright patterns slowly turn more muted, and the patterns generate closer to each other and to the center of the screen, continuing to calm you. The gently pulsating designs are entrancing to the point of being hypnotic, and offer a pleasant alternative to the coldness of the graph.

The Balloon Game is the third option, and it is a test of your newly-acquired relaxation control. You must control the flight of a hot-air balloon solely through the headband and your tension level—no joysticks or keyboards allowed. By moving the balloon up and down across the screen, you can avoid the arrows that will pop your balloon, and catch bubbles for extra points. However, I found that the best way to control the balloon is to wiggle your eyebrows up and down. You can choose varying skill levels, as well as whether you would like the game to be

determined by the number of hits or misses. The novelty of using the headband as a game peripheral in this way clearly adds to the challenge and enjoyment of the game, and the headband concept presents many exciting possibilities for future applications.

An Introductory Guided Relaxation Exercise on an audio cassette tape accompanies the package, and in fact, is one of the best, most effective parts of the package—it brought me to the threshold of sleep. It is meant to accompany the Relax Graph and features a soothing monotone male voice that provides suggestions for relaxing while calming sounds can be heard in the background—such as the rolling waves of the ocean or soft music.

	Calmpute	Relax
Program Type:	Biofeedback	Biofeedback
Machines:	Apple *IIe II+, IIe*	Apple *II* Family, C-64, IBM PC & PCjr
Distributor:	Thought Technology Ltd. 2180 Belgrave Ave. Montreal, Quebec Canada H4A2LB (514) 489-8251	Synapse Software 5221 Central Ave. Richmond, CA 94804 (415) 527-7751
Price:	$89.95	$149.95 for Apple & IBM
System Requirements:	Disk Drive	None
	Poor Fair Good Excellent	Poor Fair Good Excellent
Performance:	▬	▬▬
Ease of setup:	▬▬	▬▬
Documentation:	▬	▬▬▬

Galvanic Skin Response

The *Calmpute* hardware measures Galvanic Skin Resistance (GSR)—the conductivity or electrical resistance of the skin, and the opening and closing of its pores, both of which are controlled by the sympathetic nervous system. The Calmpute Biosensor that you hold in your hand sends a minute, electrical current through the skin. Increases in tension produce increases in current flow, which is then interpreted and displayed by the computer.

The main menu offers 8 options, including Instructions and an Introduction to Biofeedback, which simply repeat information stated in the manual. The Calm-Scope option is the equivalant of the Relax Graph, charting your tension level with a graph and needle that tracks across the screen. Calm-Bar mode is virtually the same as the Calm-Scope except that a bar chart is created, with the bar heights varying with your tension level. It is better when used for measuring over longer time periods. The Relax Graph's time setting reflects the number of readings per second, but the Calm-Scope and Calm-Bar time indicates the number of minutes it takes the computer to fill the screen with data.

"The humor comes in when your supposed response seems quite unreasonable and illogical—or at the least, out of proportion."

The Relaxation/Stress Management option is comparable to *Relax*'s Sensoral Kaleidoscope, but less sophisticated. Different geometric designs appear one at a time and you try to reduce the size of the image and/or the pitch of the feedback tone. It's a pretty dull exercise that fails to hold the attention that *Relax*'s Kaleidoscope captures.

Calm-Prix is a game also controlled by the biofeedback device. In this case you are a race car driver avoiding pot holes and curves on an ever-narrowing track. You control the car with the GSR, and the more you relax, the faster the car moves, increasing the game's difficulty and the stress involved. Obviously, the idea is to keep cool under pressure. I didn't.

The Physical and Psychological Stress Test options were the most enjoyable and the most revealing of the options. The Physical Stress Test has 4 suboptions which alternate in making you stressful and then relaxed. Exercise 1 displays some empty boxes, and when one randomly turns color you are supposed to hit the space bar as fast as possible. Your response change and percent of GSR change is then computed and displayed. In the second exercise you practice tensing your arm and inhaling, and then releasing your arm and exhaling following the peaks and valleys of the waves moving on the screen. The third exercise has you practicing relaxing and deep breathing following screen instructions, and the final exercise again has you reducing the size of a screen image by relaxing.

I found *Calmpute*'s Psychological Stress Test to be both intriguing and humorous. The mental-stressors angle is not part of the *Relax* program itself, although the workbook addresses it. In the first suboption of this test, you complete a list of your most- and least-favorite male, female, place, activity, sport, food, music, and movie or television program. The second suboption is simply a word-association test where 16 words (including a few from your new list) are presented, and your GSR response is measured for 5 seconds per word. You can save your list on the disk if desired. Horizontal bars then chart your stress reaction to each item when displayed. The humor comes in when your supposed response seems quite unreasonable and illogical—or at the least, out of proportion. I had to question the validity of the test when I received a high response to "neutral" words like "wood" and "door" and a low response to a couple of things I strongly dislike. Of course, a psychologist would probably say that there is a reason for all such seemingly incongruous responses buried deep within the psyche, but I say it's either a gliche in the whole concept or in the device.

Not Just A User Manual

Good, complete documentation is provided for the *Relax* program. The user is informed on how to start implementing a total stress-reduction program into his/her life, with and *without* the software. Most importantly, explanations on how to use the program are easy to understand, concise, and logically organized. The one exception is the manual's treatment of the Kaleidoscope option. Adjusting the EMG to the proper setting is only described in vague terms, so you can never be quite sure whether it is set correctly.

Explanations of how to use each of the options of the *Calmpute* program are adequate and complete, but the guide's treatment of the background of stress and the workings of biofeedback is rather cursory. In addition, the text seems to have been oversimplified—so much so that it comes off as being condescending.

However, the *Relax* manual is so much larger than the *Calmpute* guide (200 pages compared to 25), it is necessarily much more thorough in its coverage of stress and its related factors. Where the *Calmpute* manual provides hit-and-run coverage of a few

individual suggestions for stress management, including sample journal sheets, a "living lab" program, meditation, and yogic awareness, the *Relax* book traces the sources and effects of stress, discusses body maintenance and relaxation skills, and covers thought, time, and job-stress management.

In addition, *Relax*'s extensive self-evaluation workbook allows you to compile a personal "stress profile" of physical, mental, and other stressors and stabilizing influences in your life. This helps in identifying areas that are troubling or comforting, and in setting goals for your stress-management program. The only problem with this whole profile is that I wasn't sure what to do with the stress identification lists; sure I could fill them out, but what was I to do with that information? Its significance is never explained.

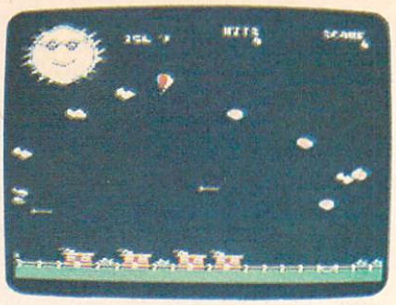
". . . muscle feedback and GSR monitoring are not the same as stress reduction, in spite of what the packages . . .uh, stress."

Is This The Answer?

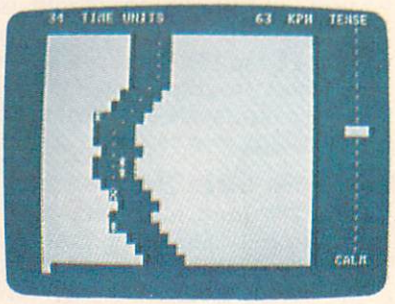
Speaking of problems, it's time to address a few important points. First, muscle feedback and GSR monitoring are not the same as stress reduction, in spite of what the packages . . .uh, stress. Physical manifestations of stress also include increased blood pressure, oxygen consumption, blood sugar production, and perspiration, among other things. Muscle relaxation takes care of only *one component* of the body's stress response, so don't overestimate the power of these packages. Although the *Relax Workbook* discusses this angle and attempts to help compensate with its lifestyle analysis and stress-reducing profile, the *Calmpute* manual glosses over it rather quickly, and its total stress-reducing program is brief and a bit lightweight.

Second, continued muscle tension and/or changes in the skin's moisture level may be a symptom of something besides stress. While these programs do seem to reach their objective of relieving muscle tension (and even headaches in a few), the biofeedback equipment is not up to professional standards, and should not be used as a cure-all to ignore the symptoms of other possible mental and bodily ills.

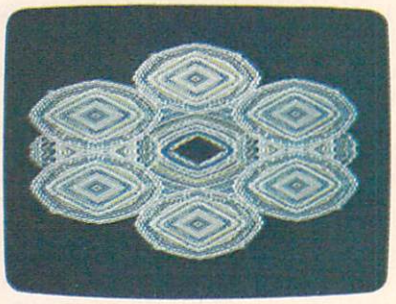
Third, there remains some doubt in my mind that what is being measured is what the manufacturers say it is, and if it is, whether the measurement is an accurate account of what is occuring. The EMG must be sensitive enough to measure the amount of electrical discharge from the skeletal muscle fibers, amplify it, filter out irrelevant signals, and display the results. I found, however, that even when the headband rested on the table, the Relax Graph's needle continued to move on the screen exactly as if I were wearing it. It was apparently picking up signals in the environment, or perhaps even from a nearby elevator—who can say? Shaking the wires attaching the EMG to the headband produced similar erratic, then smooth results on the screen. Perhaps the machine is *too* sensitive, or the filter is inadequate. It also seemed as though the electrodes were more pressure-sensitive than anything else, which would also explain why the slightest touch or movement of the wires triggered a jump of the needle.



The Balloon Game from the Relax program. The hot air balloon is controlled by an electrode-imbedded headband.



The Calm-Prix from the Calmpute program. The car's speed is regulated by the amount of galvanic skin resistance in a user's fingers.



Relax's Sensoral Kaleidoscope. The intensity of the colors changes with one's tension level.

One of the shapes from Calmpute's Relaxation/ Stress Management option. The shape's size decreases as the user relaxes.

As I mentioned earlier, this feedback from the *Calmpute* device appeared inappropriate much of the time. We've all heard of people who can trick lie-detector tests, which is part of the reason why their use is still controversial. There were times when myself and others used these machines and were, in fact, extremely relaxed—but the machines indicated an *increase* in tension or GSR. Thus, these feedback signals and their significance are questionable. My recommendation is *try before you buy*. **HCM**

HCM Review

Name:	Break Street
Program Type:	Dance Simulation
Machine:	C-64
Distributor:	Creative Software 230 East Caribbean Dr. Sunnyvale, CA 94089 (408) 745-1655
Price:	$24.95
System Requirements:	Disk drive, joystick

	Poor	Fair	Good	Excellent
Performance:				
Engrossment:				
Documentation:				

Break Street

A Review
by Dana M. Campbell
HCM Staff

Bop to the beat of the latest sounds 'till you find the streetcorner with the kids showing off all the right moves.

Creative Software seems to have devised a novel solution to an ugly problem—gang warfare. Rather than resort to the violence inherent in the combat, attack-and-destroy genre of video games, their new release, *Break Street*, proposes resolving a confrontation between street gangs by having a dance-off; a contest of style, if you will. Just as break dancing quickly caught on from the inner-city ghettos and spread across the country, this program is also catchy and works wonderfully—I hope this game's creators tackle world peace with their next program.

Of course, *Break Street*'s real premise does not focus on social problems, but on the dance contest itself. The new gang in town, the Stingrays, is challenging your turf, and the only alternative available is to use your best moves and breakdance them back to their old neighborhood.

You can choose from among 4 cool, toe-tapping breakers to execute your moves. Or, 4 players may compete in one game. Each dancer is dressed in the latest street styles, and the game's bright, bold colors and the details of the alleyway hangout make this game especially attractive to watch.

The other captivating element of this game is the actual movements of the dancers—incredibly intricate and precise. A joystick controls all movement, so it is important to have one that is extremely responsive in all 8 directions. There are 2 basic modes that can be selected with the joystick, each with 8 separate options (dance steps). Uprock is the stand-up dance mode, and includes the Wave; the Tut left and right; Poppin; the Moon to left and right; Check!, which stylishly ends your dance; and Footwork, which takes you to the ground-level mode. Footwork enables your dancer to do the 1990 (handspin); a Headspin; the Windmill; a Kneespin; the Scorpion; a Backspin; Freeze!, which also stops the dance; and Uprock, which stands you up again.

The Commodore's fine sound chip must be given some credit for adding another lively aspect to *Break Street*: its upbeat, sassy music. Even those who usually have trouble following rhythm will have no trouble choreographing a bouncy dance in tune to the beat. One drawback though: too much of a good thing can drive ya nuts! There is little variation in the tunes, and over time the constant, endless beat will bore you, if not grind on your nerves.

Flying High

The game begins in Warmup mode, where it serves as a demonstration of the dance steps and as a chance to practice routines. You can also slow down or stop the dancer to study certain moves.

And you really *can* learn some breakdance moves by watching and copying the characters while in Warmup mode, but I wouldn't recommend trying *everything* you see on the screen—this program holds a few surprises. For all of the little touches of reality injected into this program, there are a couple of moves that will make your head spin (pardon the pun). For instance, holding down the fire button and circling the joystick around while in Footwork mode will put you into a Backspin until you spin in helicopter fashion right off the top of the screen!

The Fatigue Factor

In Competition Play, a fatigue factor is introduced. While the Uprock moves are less demanding and increase your energy level, the Footwork moves get progressively more difficult as you tire. However, they are worth more points and allow you to generate bonus time. A color bar indicates the amount of energy you have left. Unfortunately, you cannot save favorite dance routines on the disk.

It is also unfortunate that the documentation is unclear in places. An outright error occurs in the Summary of Function Keys, where it says the (RUN/STOP) key restarts the game. The proper key for this is actually the (RESTORE) key. In addition, "bonus time" is mentioned, as is "extra time," and "bonus points," and "bonus," but none of them are adequately defined, so it is easy to confuse them. The problem may be just a matter of inconsistency.

A Form of Expression

This program will probably hold more appeal for adolescents than for adults, who may grow tired of the repetition of the game and its music after one or two sessions with it.

Overall though, I enjoyed *Break Street* and appreciated the opportunity it offers to participate in a new kind of performance video game. Like ballet, where the art form is simply the body itself, the only action occuring on screen is that which you choreograph the little figures to perform. It is an exciting new form of expression that breaks away from old shoot-em-up arcade tradition—possibly fostering a new, upbeat video trend. **HCM**

INDUSTRY WATCH

GAMES THAT BOTH ENTERTAIN AND EDUCATE—GUARANTEED

Officials at Springboard Software, Inc., a Minneapolis-based educational software company, are so concerned that many buyers of learning games are disappointed (because many games' educational value is questionable), that they are offering parents a unique, money-back guarantee. A refund of the full purchase price of one of its learning games will be given to any customer who feels that the software is not helping to improve the designated skills of a child using the product. The designated skills vary according to the individual game programs. To date, no consumers have returned a Springboard product under the guarantee.

LAP-SIZE IS NICE, BUT DOES IT HAVE A MARKET?

Texas Instruments has introduced a lap computer with a full 24-line flat screen that may provide some stiff competition for the Data General One, the first full-size flat-screen lap computer. TI's Pro-Lite is a 10-1/2 pound computer that fits in a briefcase, and comes with a single 3-1/2 inch floppy disk drive and 256K RAM, expandable to 720K. Selling for a cool $2,995, the Pro-Lite supposedly does not suffer as badly from a dim, low-contrast screen display, which is characteristic of the 128K Data General One selling for $2,895. The Pro-Lite can also exchange files with other TI Professional Computers and IBM PC products when an interface cable is connected. However, the success of either machine is anybody's guess. According to a Yankee Group survey, portability is at the bottom of the list of criteria of general computer buyers for judging personal computers.

EXPERT SYSTEMS ARE FINDING THEIR NICHES

Expert-system software is carving its way into more and more daily office applications, and may be quite the mass-market product for the home by the 1990's, predicts DM Data Inc., an Arizona research firm. Whereas conventional programs contain large data bases and allow the computer to retrieve whatever information the user wants to see, expert systems do the same and then draw conclusions from the information according to the same analytic methods specialists use. Expert systems are widely used in the medical field to analyze patient symptoms and then suggest possible diagnoses and treatments, and by bankers to assess risks in loans and policies. Home users may soon be able to get disks that, for example, advise them on how to fix household or car problems.

THE HOUSE KEY, THE CAR KEYS, AND THE COMPUTER KEY?

Software protection devices are taking a turn in two directions: booby traps, and hardware-based devices. Vault Corp. and Defendisk have reportedly offered (and later retracted—under pressure) software publishers the option to protect their products from being illicitly copied by putting booby-traps in their programs that could plant a "worm" in the unauthorized copier's disk operating system, or cause other havoc. Already the legalities of such a strategy are being decried and questioned, while other manufacturers in the industry say the traps could cause more problems than they're worth. Meanwhile, Lotus Development Corp. is taking another tack. They have developed an inexpensive box that plugs into some of the lines in a system's RS-232 port, accessable only with a key containing a protection code. This alternative does not "lock out" those who need to make legitimate backup copies.

IS ANOTHER TI HOME COMPUTER ON THE HORIZON?

Rumor has it that an East Coast third-party company is presently working on a new Texas Instruments-compatible computer that will be released sometime in the spring of 1985. The machine (supposedly a clone of the never-released TI-99/8 computer), is said to contain 64K RAM and 16K video RAM, and will sport the newer, 16-bit TMS9995 microprocessor. It is expected to be upwardly-compatible with existing TI software. The success of such a computer may depend on its price, its support, and the number of existing TI users that would opt for this higher- performance machine. The machine is expected to sell primarily to a residual user base already locked into a high investment in TI software and peripherals.

LISA LOSES TOP BILLING AS POWER MACHINE; NAME MIGHT CHANGE

No, the Lisa is not the most powerful computer Apple makes, according to Big Red's latest pitch to national accounts. This claim to fame is said to belong to Apple's new laser printer, which has 1.5 megabytes of RAM, 512K of ROM, and a 12-megahertz clock speed. It produces 7 pages per minute. Speaking of the Lisa, an Apple insider reports that it is not the "machine" that Apple plans to drop, but its "name." Lisa's third upgrade will supposedly unify Apple's 32-bit family, and rather than name this prima-donna Mac something like Lisa III, Apple will probably drop the "Lisa" name altogether.

IBM STRIKES OUT TO INCREASE BOTH DIRECT AND RETAIL SALES.

IBM retail dealers are receiving mixed signals from Big Blue. The November mailing of IBM's 36-page, personal-computer sales catalog to hundreds of thousands of IBM customers and others in an effort to increase direct sales is seen by many dealers to be "a direct assault on the retailer." However, prices in the catalog of hardware, software, and supplies were reportedly not aggressive enough to seriously damage retail sales. At the same time, IBM launched a multimillion dollar software inventory financing program that will provide each participating dealer with a kiosk of IBM-labled software worth up to $15,800. Most companies require dealers to finance their own software inventory, but IBM's dealers will not pay for the program's packages until they are sold. While helping dealers get their cash flow under control, the move may spell trouble for smaller software distributors vying for shelf space, who cannot afford to follow IBM's lead.

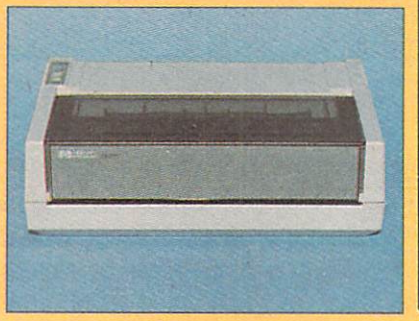
APPLE & IBM SWITCH TACTICS TO GAIN EACH OTHER'S MARKET

Not satisfied with the market segments they do own, the leaders of the business and home computer markets are now attempting to overtake each other's turf—and are sparing no expense to do it. Apple is using the 512K Macintosh to finally forge its way into the corporate market, and IBM has done exactly what it said it would always avoid—"mass merchandised" its PCjr. Apple is counting on the Fat Mac's easy-to-use environment and on Lotus Development Corp.'s Jazz integrated program to spur sales in the Fortune 2000 market. A low-cost networking product to be introduced by mid-1985 also figures into their plans. The company has reportedly allocated nearly $200 million for its advertising efforts. IBM, on the other hand, is seeking to break Apple's stranglehold on the home and school market with its PCjr. IBM has spent $40 million to advertise the PCjr during 1984, and has arranged a promotional coupon tie-in with marketing giant Procter & Gamble. This, coupled with its extensive price reductions and direct mail campaign for PCjr system/equipment packages, has sent the retail takeaway of the once-floundering PCjr soaring.

The HP Thinkjet Printer

A review by Steve Nelson
HCM Staff

HCM Review



Name:	Thinkjet
Product Type:	Printer
Machines:	TI-99/4A, IBM PC, PCjr, C-64, Apple II Family
Distributor:	Hewlett Packard 19420 Homestead Rd. Cupertino, CA 95014
Price:	$495
System Requirements:	Centronics-compatible parallel interface

	Poor	Fair	Good	Excellent
Ease of Use:				
Ease of setup:				
Performance:				
Documentation:				

If you think the noise of a printer will drive your household crazy, the Thinkjet may be able to help—it's a peaceful dream come true.

One of the most annoying sounds to me is the noise of a line printer screeching across the page as it generates text. In the offices where I work, there are often 3 or 4 of these printers going at the same time. The din from all this can make even the toughest veteran of office work chuck it all and join the granola crowd living in a tepee in Northern California. If you feel the same way I do about noisy printers, you will undoubtably appreciate Hewlett Packard's new offering—the *Thinkjet* printer.

The *Thinkjet* printer utilizes a disposable drop-in cartridge that incorporates both the print head and the ink supply. It is extremely easy to install and lasts for about 500 pages. Like a dot-matrix printer, the *Thinkjet* printer uses tiny dots to print a character. The difference between them is in the method used to transfer the ink to the paper. The *Thinkjet* printer actually squirts tiny drops of ink onto the paper from the print head. Since the head doesn't strike the paper, the only noise you hear is the sound of the print head sliding on the carriage. It's very quiet.

The print head is also much lighter than a conventional impact head, which makes the *Thinkjet* operate faster—it prints about 150 words per minute.

Special Features

The HP *Thinkjet* printer has 4 different print pitches: Normal (80 characters per line), Expanded (40 characters per line), Compressed (142 character per line), and Expanded-Compressed (71 characters per line), allowing you to have some flexibility in printing. You also have two other special functions available: Underline, and Bold print. Both of these features can be used to affect a single word or entire lines with no loss of speed. You can even mix all of the modes together, printing an expanded, underlined, bold word for instance. Changing the print pitch or adding bold or underlines is a simple process.

The print head is set to print in both directions, but you can specify one direction if you wish—it gives you slightly better alignment between lines. You can also adjust the line spacing for either 8 lines per inch or 6 lines per inch.

"The graphics modes are fully programmable and work with most popular software packages."

Graphics

The *Thinkjet* printer uses a form of graphics called "dot image," where the paper consists of a grid of dot positions. Graphics data specifies which dot positions are printed and which are left blank. This printer has two graphics modes: Default (96 by 96 dots per inch) and High Density (192 by 96 dots per inch). The graphics modes are fully programmable and work with most popular software packages.

What impressed me the most about this printer—besides its quiet operation—is its friendly nature. It is very easy to set up, and print-head changes are fast—you don't get your fingers or the paper all covered with ink. The mode-select switches are conveniently located on the back of the machine.

Loading the paper into the printer is easy, but setting the paper where you want it is a nuisance as there is no platen knob. You must use the line-feed button to position the print head at the top of the page.

Although the printer works best when using the special paper designed for it, it also works with any other computer paper, but the print quality is poor, and the print head clogs up—needing cleaning every few days. With the special paper, the print is crystal clear—almost letter quality. Unfortunately, this paper is much more expensive ($50 for a box of 2500 sheets) than regular paper (averaging $22 per box of 2500).

After using the *Thinkjet* printer for a few days, I really got to appreciate its small "footprint" on my typically cluttered desktop, and its ultra-quiet printing. The only real drawback I can see is its price: $495. For that much money, there are several printers that will give you letter-quality output on regular paper and that provide more features than the *Thinkjet*. You'll have to decide whether the added expense of both paper and printer is worth the silent, compact advantage.

Personally, I don't see too many home users opting for this little trend-setting peripheral at its current price/performance ratio. Hewlett Packard will soon be shipping a new Unix-based portable computer that has the *Thinkjet* printer built-in. Being incorporated into this kind of "luggable" computer may ultimately prove to be *Thinkjet*'s best application. **HCM**

IBM PRESSIONS

by William K. Balthrop
HCM Staff

Ready for a taste of easy graphics the Big Blue way?
Try this handy recipe for a beautiful pie chart . . .

Join us as we explore the world of the IBM PC and PCjr. Unleash the power behind these work horses while you unlock the secrets you thought accessible only to professional programmers.

Within their price range, the IBM PC (equipped with color board) and PCjr are without a doubt two of the most versatile color graphics computers you can buy today. This is evident by the ease with which business graphics can be created—as the simple pie chart program provided here will illustrate.

We're starting off this new mini-column with a pie chart because it ably demonstrates the powerful concept of using easy-to-implement graphics for a purpose other than "just playing games." Pie charts can take complex lists of hard-to-understand numeric values and convert them into simple, colorful representations—forms which have great meaning at a glance.

How To Create The Pie

The CIRCLE command is the first item in our pie chart recipe. This command is used only once in the construction of the pie chart. Only the radius and the aspect ratio are specified—all other parameters are left to their defaults. The radius is set to 90 pixels with an aspect ratio of 1.

The LINE command draws a line from the center of the circle to a point on the perimeter of the circle. The B( ) array contains a decimal value between 0 and 1 which indicates the position of the line on the circle where it meets the perimeter.

The PAINT Command

We use this command to fill in the areas of the pie with color. We made 10 different colors or combinations of color to fill different sections of the pie chart. The tiling characters are contained in the C$( ) array, which is used in place of the attribute parameter. The hexadecimal characters assigned to the array determine which color or pattern of colors will appear when used with the PAINT command. (See Figure 1.)

The program provided here can be greatly expanded in several directions. You could connect a data base to it for information storage; add a report feature which dumps the pie chart to the printer; or incorporate the chart into a spreadsheet or analysis program. The possibilities are unlimited. Experiment with what we have provided here, and see where it takes you.

HCM

For *your* key-in listing see HCM PROGRAM LISTINGS Contents.

Hex	Binary Code	First Pixel	Second Pixel
0	0000	BG	BG
1	0001	BG	GREEN
2	0010	BG	RED
3	0011	BG	BROWN
4	0100	GREEN	BG
5	0101	GREEN	GREEN
6	0110	GREEN	RED
7	0111	GREEN	BROWN
8	1000	RED	BG
9	1001	RED	GREEN
A	1010	RED	RED
B	1011	RED	BROWN
C	1100	BROWN	BG
D	1101	BROWN	GREEN
E	1110	BROWN	RED
F	1111	BROWN	BROWN

Figure 1: *This chart illustrates the 16 combinations of color possible with hexadecimal digits. Two such digits are used to specify each character in the tiling string. Thus, each character defines four pixels. BG refers to the current background color.*

Pie Chart (IBM PC and PCjr) Explanation of the Program

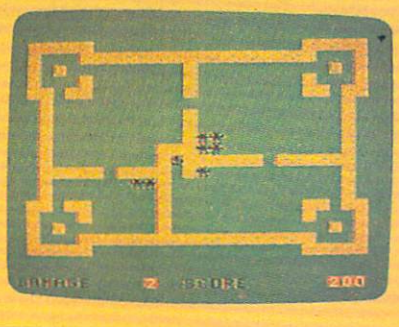
Line Nos.	
100-220	Program header.
230-280	Initialize program variables.
290-310	Display the title screen.
320-380	Enter values for the pie chart.
390-440	Create the pie chart.
450-460	Option to do another chart.

King of the Castle

A review by
Dana M. Campbell
HCM Staff

HCM Review

✧ Diamond-In-The-Rough



Name:	King of the Castle
Program Type:	Arcade game
Machine:	TI-99/4A
Distributor:	Cydex Software P.O. Box 14471 Portland, OR 97214
Price:	$17.95

System Requirements: joystick; cassette recorder and Mini Memory; or disk drive, 32K memory, and Editor/Assembler or Extended BASIC cartridge.

	Poor	Fair	Good	Excellent
Performance:				
Engrossment:				
Documentation:				

Vikings are rapidly taking over the castle . . . but all is not yet lost! The King has a few tricks up his sleeve—tricks borrowed from a star-studded Oriental cast.

Startled awake from the sound of horns blown by Viking invaders as they storm your castle, you, a Norman king, rush downstairs to defend what is left of your home. Unfortunately, your servants and guards confirm your long-standing suspicion that they are all spineless cowards who flee at the first sign of trouble. This leaves you there all alone to battle the army of intruders.

But you are no ordinary king of medieval times—facing the onslaught with only sword in hand. No, you wield some deadly Ninja throwing stars and land mines, weapons recently acquired in your travels through the Orient. What a lucky break . . .

Okay. So the premise of *King of the Castle* is a bit hokey. But that doesn't prevent it from being a good, entertaining, fast-moving game for the TI-99/4A. In fact, the first thing one notices about this game is its speed. The program is written in TMS9900 assembly language, which accounts for its speed of execution and keeps the action at a high level.

That action occurs on two floors of the castle—the second floor, where the king sleeps, and the main floor, where the Vikings first come in. Stairs in each of the 4 corners of the castle zap you from floor to floor, and a secret passageway takes you down and outside (or back upstairs if you're already outside).

The floor plans hardly, if ever, change between the two floors or between the 8 difficulty levels—which is a shame. A little more variety and challenge might be added with the doorways and stairwells appearing in different places. While I'm at it, let me add that in general, the game's graphics are strictly geometric in shape, and the screen as a whole is pretty simple. It could use some dressing-up.

Shooting Stars

Your job is to run around setting land mines for the Vikings and felling them with stars, while avoiding running into them. Contact with Vikings (which look like little spiders) is not only unsavory, it will result in damage points being counted against the king, and the destruction of the Viking contacted. With each higher difficulty level, the attackers get quicker and smarter, using more direct routes to track down the king.

King of the Castle offers nothing for those with lazy, passive strategies; you've got to work to earn any points, or even to make the game proceed. You must actively seek out the Viking warriors on the lower floor and chase them down. Hanging back on the first floor waiting for them to storm the upper floor is a waste of time, for only a couple will come up if you remain stationary. Ditto on the semi-crowded lower floor if you hide out in a corner and wait for them to approach you. These tricky Vikings prefer to cool their heels behind a nearby wall and wait for *you* to approach *them*.

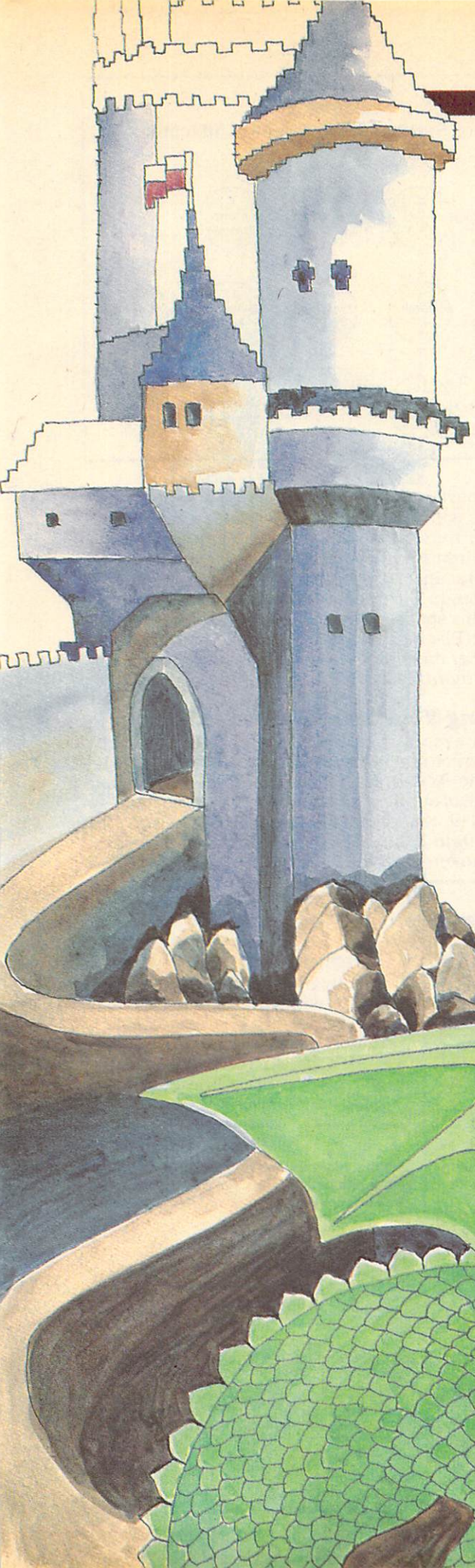
"These tricky Vikings prefer to cool their heels behind a nearby wall and wait for you to approach them."

Pursuing them is not always easy either. Often it is difficult to bring the king to a complete stop and move him through the doors or onto the stairs. This can be quite frustrating, especially when you try to get the king *facing* the right direction to fling a star, and he starts *moving* in that direction, walking right into his target!

The documentation for this game is quite brief, but complete. It just happens that everything you need in order to understand and set up the game—whether you're using cassette tape or a disk drive—can be stated on one sheet of paper.

The only suggestion I would make for this Diamond-in-the-Rough is to have it display the current difficulty level throughout the course of each game. With 8 levels and little variation between them, it's easy to forget which one you picked when you began your game. And if you play extra well or poorly, you'd want to know the game's level for later reference.

So, if you're a TI-99/4A user tired of the slowness of BASIC programs, pick up *King of the Castle*. It's a fast-playing game that puts up quite a challenge and will keep you engrossed for hours. HCM



Build a LOGO Adventure

Part 2 of 4
by Andrew Keith
and the HCM Staff

In this tutorial, we turn keyboard commands into computer action as we map out our freshly created adventure world.

In Part 1 of this tutorial, we began to construct an Adventure Game using LOGO. We started off by writing a procedure, GETCOMMAND, to get a typed-in command (like N to go north) and pass it to another procedure, PARSER, which analyzes the words in that command. These procedures all appear in *Home Computer Magazine* Vol. 4, No. 5. But the essence of an Adventure is to *go* somewhere. Even though we wanted to go north in the previous segment, we hadn't yet given it a map to use. We need to create that map now and translate it into LOGO, so that the computer will be able to respond to our moves. We must first decide where our Adventure is to take place. It can be set anywhere—from an African jungle to a space station in the year 2199 A.D. For our game, we chose a medieval-style fantasy land populated by witches, dragons, and the like. You'll find that once you have established a setting, your adventure will take on a life of its own. Our next step is to think up what the locations (or "rooms" as they are called) will be in our

world. They may be indoor or outdoor, and you may want them to have special characteristics—secret panels, for example. In our scenario, we have limited the number of locations to 12 due to memory limitations. Still, this will allow us enough variety to include a cave (with resident dragon), a castle, a treehouse, a marsh, and eight other locations.

Walk Through a World of Your Own Design

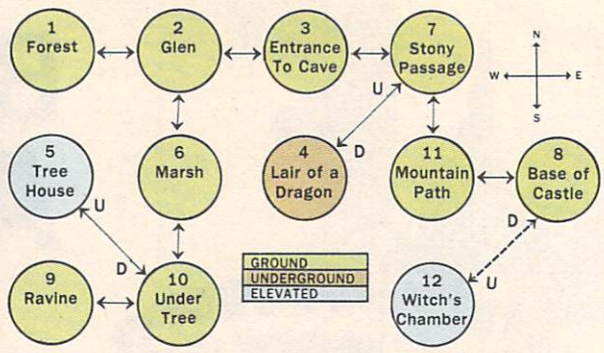
Having created a preliminary setting, we can now draw a map. The simplest way to do this is to use connecting circles with arrows pointing north, south, east and west. Again, because of memory limitations, we will not include directions like northeast and southeast, but we *will* include up and down. The map for our Adventure is shown in Figure 1. The rooms highlighted in green are at ground level, the underground lair of the dragon is in earth-tones, and the elevated rooms are shown in blue.

In studying the map, you may notice that the up/down path between the base of the castle and the witch's chamber is dotted. We did this to indicate that passage is conditional: the Adventurer will have to possess certain items in order to gain access.

Once you are satisfied with *your* map, you must convert the directions into numbers that the computer can understand. In our example, because the rooms are already numbered on the map, we can look at the number of the room we are moving into. For example, suppose we are in room number 7, the stony passage. Moving south will bring us to room number 11, a mountain path. If we go down, we will descend into room number 4, the lair of the dragon. Movement west will take us back to room number 3, the entrance to the cave. North, east, and up do not lead into any rooms, so the program will tell us that we can't go those ways.

To convert the map into data, we can draw up a chart called a movement table (see Figure 2) that will tell us the number of the room in which we will end up if we take a certain path. If we cannot go that way, we'll just mark it with a period (using periods instead of numbers will save about 45 nodes of storage). Note that on the movement table for this game, rooms 8 and 12 contain numbers marked with asterisks. This indicates that special conditions must be met in order to pass through. These conditions, and the procedures to handle this special case, will be included in the next installment. Placing periods in these positions in Listing 1 will only bar movement to the witch's chamber (room 12) in *this month*'s LOGO Adventure.

Figure 1
Map of Settings Used in LOGO Adventure



Converting Your Map to Lists

Now we are ready to convert our map and movement table into something LOGO understands very well: lists. We'll call one "ROOMS and fill it with 12 smaller lists that represent all of our locations, numbering them for easy reference. The other list will be called "DIRECTIONS and will also contain 12 sublists, these being numbers taken from the movement table. We'll put them into a procedure named SET.UP. As we mentioned above, entry to the witch's chamber will be conditional, so its corresponding list in :DIRECTIONS will be all periods, effectively sealing off the room. SET.UP will also include the list of verbs that the parser understands (see Listing 1).

This procedure, when run, will read-in all of the necessary information LOGO needs to follow

Listing 1

APPLE LOGO II
LOGO SYSTEMS LOGO — IBM PC & PCjr

```
TO MAKE.USE.LIST
MAKE "USE [[GET KEY] [DROP KEY] [EAT FOOD] [OPEN CHEST] [LIGHT LAMP]]
END
TO MAKE.VERB.LIST
MAKE "VERB [[N 1] [S 2] [E 3] [W 4] [U 5] [D 6] [GET 7] [DROP 8] [EAT 9] [OPEN 10] [LIGHT 11] [TAKE 12]]
END
TO SET.UP
MAKE.VERB.LIST
MAKE.USE.LIST
MAKE "ROOMS [[.] [.] [.] [.] [.] [.] [.] [.] [.] [.] [.] [.]]
MAKE "DIRECTIONS [[2 4 . . 3 . . .] [. 5 . 1 . . . .] [. . . 2 . . . .] [. . 1 . . . . .] [. . . . . 1 . .] [. . . . . 2 . .] [. . . . . 3 . .] [. . . . . 4 . .] [. . . . . 5 . .] [. . . . . 6 . .] [. . . . . 7 . .] [. . . . . 8 . .]]
END
```

TERRAPIN LOGO — C-64

```
TO MAKE.VERB.LIST
MAKE "VERB [[N 1] [S 2] [E 3] [W 4] [U 5] [D 6] [GET 7] [DROP 8] [EAT 9] [OPEN 10] [LIGHT 11] [TAKE 12]]
END
TO MAKE.USE.LIST
MAKE "USE [[GET KEY] [DROP KEY] [EAT FOOD] [OPEN CHEST] [LIGHT LAMP]]
END
TO SET.UP
MAKE.VERB.LIST
MAKE.USE.LIST
MAKE "ROOMS [[.] [.] [.] [.] [.] [.] [.] [.] [.] [.] [.] [.]]
MAKE "DIRECTIONS [[2 4 . . 3 . . .] [. 5 . 1 . . . .] [. . . 2 . . . .] [. . 1 . . . . .] [. . . . . 1 . .] [. . . . . 2 . .] [. . . . . 3 . .] [. . . . . 4 . .] [. . . . . 5 . .] [. . . . . 6 . .] [. . . . . 7 . .] [. . . . . 8 . .]]
END
```

Listing 2

APPLE LOGO II
LOGO SYSTEMS LOGO — IBM PC & PCjr

```
TO PRINT.LOC
PRINT [YOU ARE IN A]
PRINT :ROOMS
END
```

TERRAPIN LOGO — C-64

```
TO PRINT.LOC
PRINT [YOU ARE IN A]
PRINT :ROOMS
END
```

Listing 3

APPLE LOGO II
LOGO SYSTEMS LOGO — IBM PC & PCjr

```
TO MOVE :X
IF (ITEM :X :DIRECTIONS) = "." [PRINT [YOU CAN'T GO THAT WAY]]
IF (ITEM :X :DIRECTIONS) = 0 [PRINT [YOU CAN'T GO THAT WAY]]
IF (ITEM :X :DIRECTIONS) = 7 [PRINT [YOU CAN'T GO THAT WAY]]
IF (ITEM :X :DIRECTIONS) = 8 [PRINT [YOU CAN'T GO THAT WAY]]
MAKE "HERE ITEM :X :DIRECTIONS
END
```

TERRAPIN LOGO — C-64

```
TO MOVE :X
IF (ITEM :X :DIRECTIONS) = "." [PRINT [YOU CAN'T GO THAT WAY]]
IF (ITEM :X :DIRECTIONS) = 0 [PRINT [YOU CAN'T GO THAT WAY]]
IF (ITEM :X :DIRECTIONS) = 7 [PRINT [YOU CAN'T GO THAT WAY]]
IF (ITEM :X :DIRECTIONS) = 8 [PRINT [YOU CAN'T GO THAT WAY]]
MAKE "HERE ITEM :X :DIRECTIONS
END
```

Figure 2
Movement Table for Rooms in LOGO Adventure

	1 N	2 S	3 E	4 W	5 U	6 D
1. Forest	.	.	2	.	.	.
2. Glen	.	6	3	1	.	.
3. Entrance to Cave	.	.	7	2	.	.
4. Lair of Dragon	.	.	.	.	7	.
5. Tree House	.	.	.	.	.	10
6. Marsh	2	10	.	.	.	.
7. Stony Passage	.	11	.	3	.	4
8. Base of Castle	.	.	.	11	12*	.
9. Ravine	.	.	10	.	.	.
10. Under Tree	6	.	.	9	5	.
11. Mountain Path	7	.	.	8	.	.
12. Witch's Chamber	.	.	.	.	.	8*

**Passage between these points will be conditional.*

our commands. As the game further develops, we will add more to SET.UP. For now, notice the last line: MAKE "HERE 2. The variable :HERE will be used to store our current location in the game, and will constantly change. At the beginning of the game, it is set for room number 2, the glen. We need to modify the ID.LOC procedure published in the last issue so that it changes the number in the :HERE variable into the actual name stored in :ROOMS (see Listing 2).

"You'll find that once you have established a setting, your adventure will take on a life of its own."

ID.LOC prints a sentence consisting of YOU ARE and the BUTFIRST of the second item of the :ROOMS list. ITEM 2 of :ROOMS is the sublist (2 IN A GLEN) and the BUTFIRST of that is (IN A GLEN), which is combined with YOU ARE to print the message, YOU ARE IN A GLEN. Control then returns to GETCOMMAND, which prints out a prompt and waits for the player to type something. In the actual program the screen will look like this:

```
YOU ARE IN A GLEN
COMMAND?
```

Lists: The Key to Movement

Well, that is certainly an improvement over last time! But we'll get bored very quickly by just standing around in the glen. We still need a way to move from place to place. The answer? You guessed it—write another procedure (see Listing 3).

MOVE takes one input, :X (which will be a number from one to six), representing a direction for travel. Let's say that we want to go west:

```
COMMAND?
W
```

GETCOMMAND and PARSER will process W and send us to the procedure of the same name. To make movement possible throughout our fantasy world you will need to key-in all of the direction procedures: N, E, W, S, U, and D (see Listing 4). Note that N has been altered to work like the other direction procedures.

Because we are presently in the glen (:HERE = 2), MOVE first examines the second ITEM of the :DIRECTIONS list as defined in the SET.UP procedure: (. 6 3 1 . .). The procedure W passes the value 4 when it calls MOVE (see Listing 4), so MOVE checks to see whether ITEM 4 of the sublist (. 6 3 1 . .) is a period or a number. Because the fourth item is the number 1, MOVE makes "HERE equal to 1 and returns to GETCOMMAND. When we loop back up to ID.LOC, it displays:

```
YOU ARE IN A FOREST
COMMAND?
```

Conversely, had we wanted to move north, MOVE would have been called as MOVE 1. The program would have found that ITEM 1 in (. 6 3 1 . .) *is* equal to a period, skipped to the end of the MOVE procedure, and printed YOU CAN'T GO THAT WAY. Because there would have been no change in :HERE, we would find we are still in the glen.

We can put all this together in a super-procedure as seen in Listing 5: ADVENTURE.

Once you define these procedures, you will be able to move from place to place around this or any other map you wish to create. Yet, our fantasy world still seems sterile and empty, so in the next segment we will populate it with objects and treasures—and eventually, characters to interact with. In the meantime, try experimenting by creating new fantasy worlds, and think about what objects and characters will be part of your own story. **HCM**

Listing 4

APPLE LOGO II
LOGO SYSTEMS LOGO — IBM PC & PCjr

```
TO S
MOVE 2
END

TO D
MOVE 6
END

TO N
MOVE 1
END

TO U
MOVE 5
END

TO W
MOVE 4
END

TO E
MOVE 3
END
```

TERRAPIN LOGO — C-64

```
TO D
 MOVE 6
END

TO U
 MOVE 5
END

TO W
 MOVE 4
END

TO E
 MOVE 3
END

TO S
 MOVE 2
END

TO N
 MOVE 1
END
```

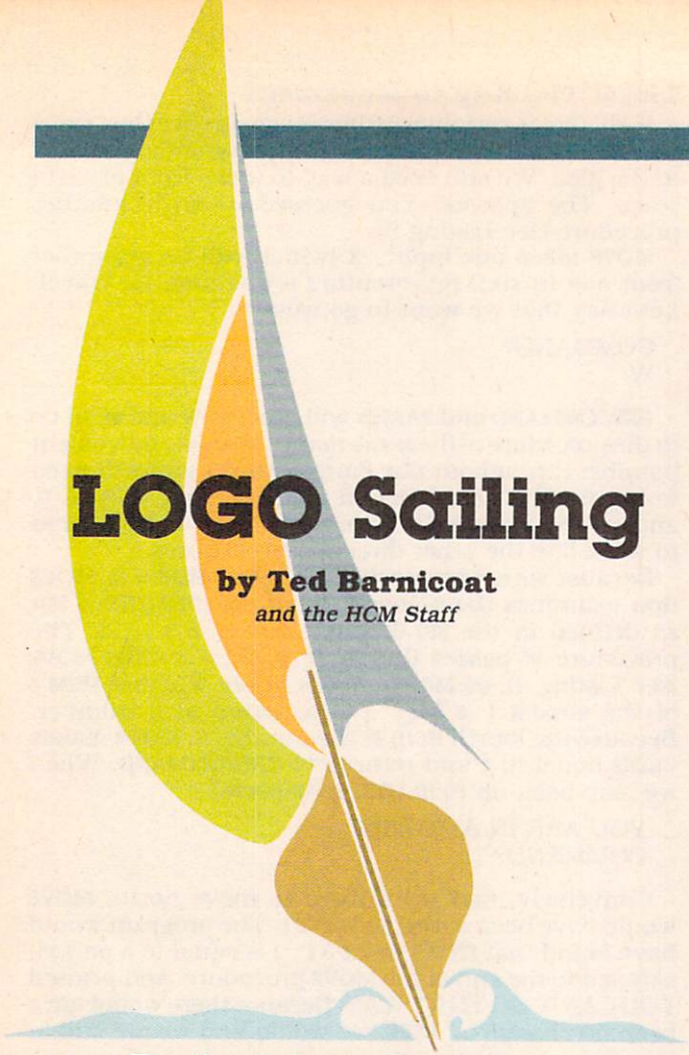
Listing 5

APPLE LOGO II
LOGO SYSTEMS LOGO — IBM PC & PCjr

```
TO ADVENTURE
TS CT
PR [WELCOME TO A NEW LO
GO ADVENTURE]
SET.UP
RECYCLE
GETCOMMAND
END
```

TERRAPIN LOGO — C-64

```
TO ADVENTURE
 ND
 PR [WELCOME TO A NEW L
OGO ADVENTURE]
 SET.UP
 .GCOLL
 GETCOMMAND
END
```



LOGO Sailing

by Ted Barnicoat
and the HCM Staff

Turtles take to the water as they sail for the America's Cup.

My passion is sailing. Unfortunately, I live in Edmonton Canada, a prairie city not known for its sailing facilities. I have therefore developed this model of a yacht race to quiet some of my passion, and pass the long northern nights. Although it is not a typical computer game, it should give sailor and non-sailor alike a feeling of the excitement of last year's America's Cup series.

Let's Bring America's Cup Home

The United States has dominated the America's Cup since the race began in 1851. That is, until the summer of '83, when the Aussies took the trophy down under. Well, it's time to bring the America's Cup back to North America—preferably to Canada! You can't start practicing soon enough, and with your LOGO-equipped TI-99/4A and joysticks, *LOGO Sailing* can help.

This game is a model of a "match" race between two yachts, just like the America's Cup. It can be used to teach the principles of sailing before venturing out in a real yacht. It could also help as a tactics analyzer for experienced yacht racers. Or, it might just get the adrenaline going as you and your crew are waiting for the start of the sailing season. Good sailing mate!

The Principles of Sailing

The sort of yacht used in *LOGO Sailing* has one mast and two sails. The "jib" is the sail in front of the mast, and the "mainsail" is the larger one behind or "aft" of the mast. The mainsail is secured along the bottom to a wood or metal pole called the "boom," which swings back and forth allowing the yacht to make maximum use of the wind.

The "rudder" is a plate secured at the back of the yacht which controls the yacht's direction. The "tiller" is a lever attached to the top of the rudder which allows you to turn the yacht. The rudder moves in the opposite direction to the tiller. That is, if the tiller is pulled to the left, the yacht moves to the right. Also, sailboats do not react immediately, so you must hold the tiller in the proper direction for the turn until the yacht responds. In *LOGO Sailing*, you use the joysticks exactly the same way.

A yacht moves because its sails convert wind energy into a driving force that acts roughly at right angles to the sails. This is split into a forward force (F) and a sideways force (S). Because a yacht has an underwater fin called a "keel," any S motion is reduced and converted into F motion—a bit like squeezing an orange pip between your fingers. These forces move the yacht forward dependent on the speed and direction of the wind in relation to the direction the yacht is moving. If the wind is right behind you, there is lots of F and little or no S. If you are moving into the wind, there is more S, and this reduces the F.

The four primary ways that a yacht moves in relation to the wind are called "points of sail." They are:

1. **Head to Wind** - When you point the front (bow) of the yacht directly into wind, the sails flap, the boom bangs around dangerously, and the yacht moves backward because the sails are generating no forward force.
2. **Tack** - When you pull the tiller until the yacht moves away from "head to wind" and the sails begin to fill with the wind. This is called a "port tack" if you are going to the right but the wind is coming over the left (port) side of the yacht as you look towards the bow. It is a "starboard tack" if you are going to the left with the wind coming from the right, or starboard side of the yacht.
3. **Reach** - When the yacht is moving at 90 degrees to the wind you are "reaching." It is probably the fastest point of sailing. Again, there is a "port" and "starboard" reach.
4. **Run** - When the wind is behind you, your sails act like kites and pull you forward. At this stage, your yacht uses one of those big balloon-shaped sails called a spinnaker, which allows the yacht to "run" faster.

Racing in LOGO

Before you begin, you must not only key in the procedures, but also define the 25 shapes and 3 characters (tiles) according to the figures on the right. Use the MAKESHAPE and MAKECHAR commands to define these shapes. Once they are created, be sure to SAVE ALL and not just the procedures, so you won't have to do it again. [To save you from the somewhat-tedious task of entering the shapes, we'll include them in this issue's edition of ON DISK and ON TAPE—Ed.] Begin the program by typing AMCUP from the LOGO command screen. This game requires the use of joysticks for control of the yachts.

A typical yacht race is run round a triangular course (see Figure 1). Although winds change in real yacht racing, due to memory limitations we always have the wind coming from the north.

LOGO Sailing keeps track of both yachts' progress in the CHK procedure through the letters (A-J) designated on the map in Figure 1. Each yacht's status is displayed

in the upper-left corner of the screen. The first yacht to successfuly complete the course (reach letter J) wins the race.

The yachts in this race are sprites, and while sprites "wrap" on the screen, yachts on real race courses don't. Thus, our rules prohibit going off the bottom or top of the screen—a yacht loses when it does so. Due to memory limitations, no such check is made for side-to-side wraps. The race is *always* between two yachts, and only because of this lack of checking can one person use the program alone. To practice the game alone, just set one yacht on a reach (moving horizontally across the screen) with one joystick, then practice racing with the other yacht.

If you have LOGO II, you have considerably more memory available to you. You might try enhancing the program to include the side-to-side check as well. The top-to-bottom check is made at the beginning of the `CHK` procedure, with the `:Y` and `:X` variable containing the respective row and column locations of the yacht. This enhancement would require you to add some method of anchoring one yacht so you could practice with the other. [Keep us informed of your enhancements in a "Letter to the Editor."—Ed.]

A Collision Ends the Race

When the two yachts are on a collision course, the yacht that is traveling more "with the wind" has to yield to the other. If they both have the wind in the same relation to their yacht (i.e., the same "point of sail"), the one with the wind coming over its starboard bow has the right-of-way. For example, a yacht tacking northeast (we say it is on a starboard tack) has the right-of-way over one tacking northwest (on a port tack). If the yachts do collide, they sink, and the yacht with the right-of-way is awarded that race.

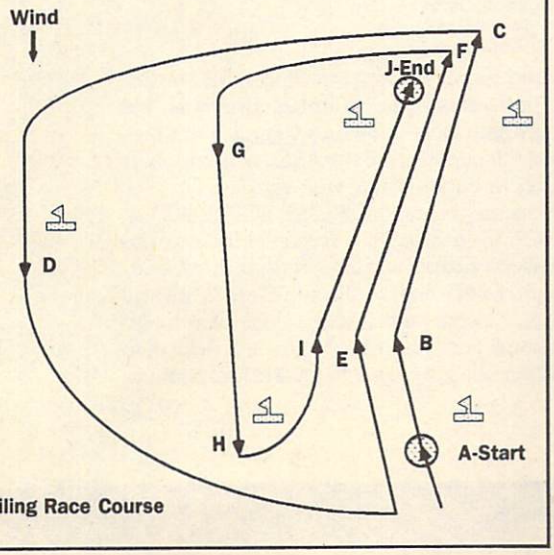
No part of your yacht, including the sail, is allowed to touch the mark. If you do, you must go back around that mark again without touching it.

Each race follows the same course and the America's Cup is awarded on the basis of the best 4 of 7 races. If, when asked whether you want to replay, you choose `N` (for No) the program is cleared with the `BYE` command.

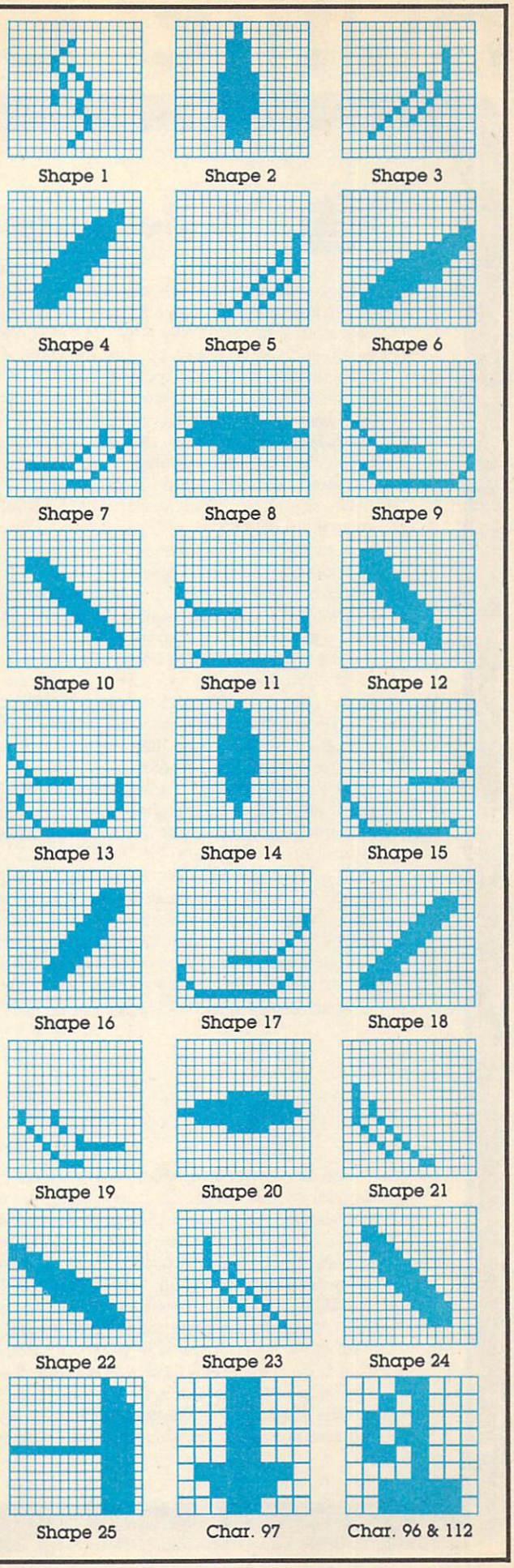
We hope you enjoy LOGO Sailing, and don't forget to give your TI-99/4A credit when you bring the America's Cup back to North America in 1987. **HCM**

For *your* key-in listings see HCM PROGRAM LISTINGS Contents.

Figure 1



Logo Sailing Race Course



Hi-Res Page Switching

High-resolution (hi-res) graphics accessibility from BASIC on the Apple *II* Family of computers is one of the computer's greatest assets to beginner and experienced programmers alike. There are 2 hi-res display Pages in the Apple. These hi-res Pages are 8K-byte-long memory areas, and should not be confused with the 256-byte pages such as zero page (the lowest 256 bytes of memory in the Apple) or the I/O page (between $C000 and $C100). To differentiate between the two, we will capitalize Page when referring to hi-res areas, and use a lower-case "p" when referring to the smaller 256-byte pages of the Apple's memory. The fact that two hi-res Pages are available opens up all sorts of graphic possibilities, but fluid interaction between the two Pages requires the programmer to have a thorough understanding of the "soft-switches" controlling these Pages.

The two hi-res Pages are located in distinct areas of memory—Page 1 from address 8192 through 16383 ($2000-3FFF) and Page 2 from address 16384 through 24575 ($4000-$5FFF). How the data on these Pages is transformed into graphics is beyond the scope of this article. In addition, you must be aware that large Applesoft programs can "over-write" these areas of memory, unless certain precautions are taken to protect them. The methods for such protection were discussed in detail in the Apple "Home Computer Tech Note" in Vol. 4, No. 1. To use hi-res Page 2, your larger BASIC programs should make use of those methods to ensure that the program is loaded above hi-res Page 2—starting no lower in memory than 24576 ($6000).

Having taken these necessary precautions, you can use the **HGR** statement to clear hi-res Page 1, and then **HPLOT**, **HCOLOR=**, **DRAW**, **SCALE**, **ROTATE**, etc. to create graphics on Page 1. Likewise, the **HGR2** command clears Page 2, and all subsequent hi-res commands default to Page 2. Neither of the **HGR** statements affect the other Page, so there is no danger of erasing the Page you're not addressing. However, if you set up one picture on Page 1 and then draw a different one on Page 2, using the **HGR** statement would clear the screen— preventing you from simply modifying the Page 1 drawing. Thus, you cannot use **HGR** and **HGR2** to alternate freely between the two Pages in a program. But, there is another way: the "soft-switches" located on the I/O page ($C000-$C100) in the computer.

To activate the soft-switches on the I/O page you may either **PEEK** or **POKE** the appropriate location. The act of accessing the memory location is what activates the switch, so it doesn't matter what value is contained there. It also doesn't matter what you do with the value you **PEEK** (the value is not significant), and as long as you **POKE** a legal value (between 0 and 255) it doesn't matter what value you **POKE** either. Here's a short list of the I/O page locations that let you modify the display modes:

POKE Location (In Hex)	POKE Location (in Decimal)	Switch Affected
$C050	49232 or -16304	Enter Graphics mode
$C051	49233 or -16303	Return to Text mode
$C052	49234 or -16302	Eliminate text on bottom of screen (all graphics)
$C053	49235 or -16301	Display text at bottom of screen (mixed mode)
$C054	49236 or -16300	Display Page 1
$C055	49237 or -16299	Display Page 2
$C056	49238 or -16298	Lo-res mode active
$C057	49239 or -16297	Hi-res mode active

One other critical memory location must be altered to switch between Pages: $E6 or 230 decimal on zero page. The locations in the above table affect what is being displayed, but location 230 is used by BASIC to determine which graphic Page is being written to. If it contains a 32, then hi-res commands write to Page 1, and if it contains a 64, BASIC writes to Page 2. Thus it is possible to display Page 1 while writing to Page 2, and vice versa.

The **HGR** statement effectively **POKE**s 230 with 32, accesses locations $C050, $C053, $C054, and $C057, and then resets all the bits in the area to 0, clearing the screen. Likewise, the **HGR2** statement does the same thing, except that $C055 takes the place of $C054, and 230 is set to 64. Thus, if you wish to write to Page 1 immediately after writing to Page 2, all you have to do is enter **POKE 230,32:POKE 49237,0** and Page 1 becomes the active Page. If you also wanted to go to all graphics from mixed mode at the same time you made this switch you could execute **POKE 230,32:POKE 49237,PEEK(49234)**. By **POKE**ing one location with the **PEEK** of another, you access two locations and set both switches.

—Roger Wood

Merging Files from BASIC

The Commodore 64 lacks some of the more sophisticated programing tools of some other systems—among them the ability to merge one program file with another. Each time you **LOAD** a program any program in memory is lost. With a disk drive and the following short MERGE listing, however, you can merge two programs of practically any length.

```
63800 INPUT "SEQ FILE TO BE MERGED:";FI$:
      FI$=LEFT$(FI$,16)
63810 OPEN 2,8,2,FI$+",S,R"
63820 CR$=CHR$(13):POKE 152,1:FLAG=0:L$="
      ":PRINT "■SHIFT CLR■2CRSRDOWN■"
63830 GET #2,CH$:IF CH$="" THEN 63830
63840 IF ST AND 64 THEN 63910
63850 IF CH$=CR$ AND FLAG=0 THEN 63830
63860 PRINT CH$;:FLAG=1:L$=L$+CH$
63870 IF CH$<>CR$ THEN 63830
63880 PRINT CHR$(31);"GOTO 63820";CHR$(15
      4);"■2CRSRDOWN■":POKE 198,0
63890 IF NOT(ASC(L$)>47 AND ASC(L$)<58)TH
      EN 63820
63900 GOTO 63970
63910 CLOSE2:LN=63800
63940 PRINT "■SHIFT CLR■2CRSRDOWN■":FOR
      N=LN TO LN+60 STEP10: IF N<64000 TH
      EN PRINT N:NEXT N
63950 PRINT "LN=";N;:IF N<63950 THEN PRIN
      T":GOTO 63940":GOTO 63970
63960 PRINT"■4CRSRDOWN■":GOTO 63970
63970 FOR I=631 TO 640:POKE I,13:NEXT I:P
      OKE 198,10:PRINT"■HOME■";:END
```

Let's say you have two programs: a main program (PROGRAM1) and a second one you want to merge with the first (PROGRAM2). The first step is to key in the listing shown here and **SAVE** it on disk as MERGE. This program erases itself from memory when it is through running, so don't try to **RUN** it—just proofread it very carefully before you **SAVE** it.

Next, **SAVE** the program file to be merged as a **SEQ**uential file. Just **LOAD** it normally, and then **LIST** it to disk. This is done in command mode with these lines:

```
OPEN 2,8,2,"PROGRAM2.L,S,W":CMD 2:LIST:PRINT#2
CLOSE2
```

Above, PROGRAM2.L should be a distinctive name for the **SEQ**uential version of the file to be merged so as not to confuse it with a BASIC **PR**o**G**ram file. Note that by appending a **.L** to the name used for the **PR**o**G**ram file, the **SEQ**uential version is easy to identify.

The next step is to **LOAD** the program to be merged to: **LOAD "PROGRAM1",8**

It's important that neither PROGRAM1 nor PROGRAM2 has a line number as big as the lowest line in the MERGE program listed here (i.e., all must be less than 63800). Now, the MERGE listing below must be added to the PROGRAM1 in memory. One way to do this is to key it in. This is time-consuming, and one error will lead to a "bomb."

Here's another way: Make sure you have the MERGE listing already saved to disk. Memory locations 45 and 46 are pointers to the end of BASIC text. By making the beginning of BASIC text pointers in locations 43 and 44 point two locations in front of the present end of BASIC, a **LOAD** command will place the MERGE program directly after PROGRAM1. Then, resetting the beginning of BASIC pointers to their normal values, the two programs will link together into one. For example, if when you type **PRINT PEEK(45)**, the number 35 is printed, and **PRINT PEEK(46)** yields 15. To reset the pointers, just type **POKE 43,35—2:POKE 44,15**.

Having made this adjustment, **LOAD** MERGE from disk, and type **POKE 43,1:POKE 44,8** to reset the beginning of BASIC text pointers back to the beginning of PROGRAM1. When you **LIST** the program you will find that PROGRAM1 now has the MERGE program appended to it. This method only works if all the line numbers in the program appended to the first one are higher than those in the first one.

Next, type **RUN 68300** and the MERGE program begins to execute. It will prompt you for the name of the **SEQ**uential file to be merged (i.e., PROGRAM2.L). When you input this name, the program reads the PROGRAM2 in from the disk one line at a time. It then prints the lines to the screen followed by a **GOTO 63940** command, fills the keyboard buffer with carriage returns (ASCII 13), places the cursor at the [**HOME**] position and **END**s. When any program halts, all characters in the buffer are printed to the screen—in this case, a series of carriage returns actually merge the program lines into the existing program and then restarts the program at line 63940, where the next line in PROGRAM2 is printed. The process continues until all of the lines in PROGRAM2 are added, then all lines above 63800 (i.e., the MERGE program) are deleted, leaving you with a program in memory that is a successful merging of PROGRAM1 and PROGRAM2. Be aware that if PROGRAM2 contains any line numbers that are identical with those of PROGRAM1, they will replace those in PROGRAM1.

—Roger Wood

PC Hard Copy Screen Graphics PCjr

It is easier than you think to capture that high score on the screen for posterity, or get a hard copy of the pie chart created with your financial analysis program. IBM PC and PCjr computers come equipped with the ability to dump the contents of any graphics screen to the printer—with some exceptions noted below—providing you have a graphics printer compatible with the IBM system.

In order to dump a graphics screen to the printer from BASIC, first invoke the **GRAPHICS** command from DOS. Before starting BASIC—while the system is still in DOS—simply type **GRAPHICS** and press **[RETURN]**. Be sure to have a DOS disk inserted in the default drive, usually drive A. If your system auto-boots BASIC, return to DOS by typing **SYSTEM** and pressing **[RETURN]**. Then enter **GRAPHICS**. After the graphics program has been added to the system, the DOS prompt will appear. Now start BASIC by typing either **BASIC** on the PCjr or **BASICA** on the PC. Then, to dump a screen to the printer, just press the **[PrtSc]** key.

To use this screen dump utility, you must have a compatible graphics printer connected to your system. The computer sends a sequence of codes to the printer which tells it to start printing graphics instead of ASCII characters. Your printer must be able to interpret this code to set itself up properly. The set up code is as follows:

```
LPRINT CHR$(27);CHR$(76);CHR$(n1);CHR$(n2);
```

The first two codes, 27 and 76, set the printer for double-density graphics mode. Your printer should be capable of printing at least 960 dots per row when in this mode. The values for n1 and n2 tell the printer that the next n2*256+n1 characters should be interpreted as dot graphics. The n1 and n2 values are calculated in the following manner:

n2 = INT(# of dots**/256)**

n1 = (# of dots) − **n2**

(# of dots) = **(n2*256)+n1**

You may need to make one final preparation before you can get perfect screen dumps. The normal line spacing for most printers is either 6 or 8 lines per inch. Most printers think of this as either 12/72" per line or 8/72" per line. But in order to get good screen dumps, you need to set the printer to 7/72" per line. If you can't do this, you may end up with white space between each printed row of graphics. For example, you can set this command on the Epson MX-80 with the following command sequence:

```
LPRINT CHR$(27);CHR$(65);CHR$(7);
```

Now for the exceptions: Because the **GRAPHICS** command was first implemented on the PC and carried over to the PCjr, there are several screen modes that will not work with the **GRAPHICS** command from DOS on the PCjr. Screen modes 0, 1, 2, and 4 all work the same way as the corresponding modes on the PC. (Screen mode 4 on the PCjr works identically to PC mode 1.) Screen modes 3, 5, and 6 will not work. Modes 3, 5, and 6 store the graphics screen differently than any of the modes available on the PC. Additionally, it would be difficult to modify the **GRAPHICS** command to work with these new modes because it would not be able to create shading for 16 colors, as it does with the 3 colors available in modes 1 and 4. Shading is accomplished on the printer by dispersing the dot pattern.

—William K. Balthrop

[Special note to PCjr owners: In Volume 4, Number 4, we published an article on adding a second disk drive to Junior. This project requires a special cable and two IC chips. We now renew our offer to supply these parts in kit form. Order now, while limited supplies are still available, by sending $49.95 to: Home Computer Magazine, Attn: PCjr Disk Kit, 1500 Valley River Drive, Suite 250, Eugene, OR 97401.]

Speeding Up Extended BASIC

One of the complaints we frequently hear about programming on the TI-99/4A is the speed of the computer when running BASIC or Extended BASIC programs. Compared to other home computer BASICs available, the TI variety is fairly slow. A number of reasons account for this—too many to explain fully in this limited space. However, if you have Extended BASIC, there are some things you can do to improve your computer's execution speed. And, if you have the 32K Memory Expansion card, you can increase that speed even more.

The first method involves inserting a special code in your programs that will turn the "pre-scan" on or off. Each time you **RUN** a program on the 99/4A, the computer pre-scans it, setting aside room in memory for variable storage. This is why the computer seems to hesitate—sometimes at considerable length—before executing a program.

It is not necessary to scan a whole program, however, and a lot of time is wasted by doing so. By turning the pre-scan off and on at selected points, you can decrease the time it takes your program to start. Use a bit of caution here, though. All variables and subprograms must be pre-scanned before a program starts executing. The following lines will ensure that these items do come under pre-scan:

```
(Program starts in pre-scan mode)
100  VAR1 :: VAR2 :: SVAR2$ :: VAR3 :: VAR4 :: DIM A(100)
110  CALL CHAR :: CALL SOUND :: CALL HCHAR :: CALLVCHAR :: CALL KEY
120  !@P-   (turn pre-scan off)
.
.
(place main program here)
.
.
750  !@P+   (turn pre-scan on)
CALL SPRITE :: CALL SUB1 :: CALL SUB2 :: !@P-   (turn pre-scan off)
.
.
(place main program or sub-routines here)
```

In the example shown above, **SUB1** and **SUB2** are Extended BASIC subprograms. The codes to turn the pre-scan off and on are as follows:

!@P- Turns pre-scan off.
!@P+ Turns pre-scan on.

The second method requires that you have the 32K Memory Expansion card installed in your system. To increase overall execution speed, you can disable the automatic motion of unneeded sprites—which eats up processor time—so that your computer has more time to work on other tasks. Usually, the computer is constantly updating all 28 sprites—even when they are not being used. You can make the computer update fewer sprites, or disable them completely if your program doesn't need them. The following command will disable all of the sprites:

```
100  CALL INIT :: CALL LOAD(-31878,0)
```

By changing the second parameter in the **CALL LOAD** statement from 0 to 5, you would enable 5 sprites (#1 through #5), while still significantly increasing execution speed. Just change this parameter to enable the maximum number of sprites which your program needs to use.

—William K. Balthrop

Group Grapevine

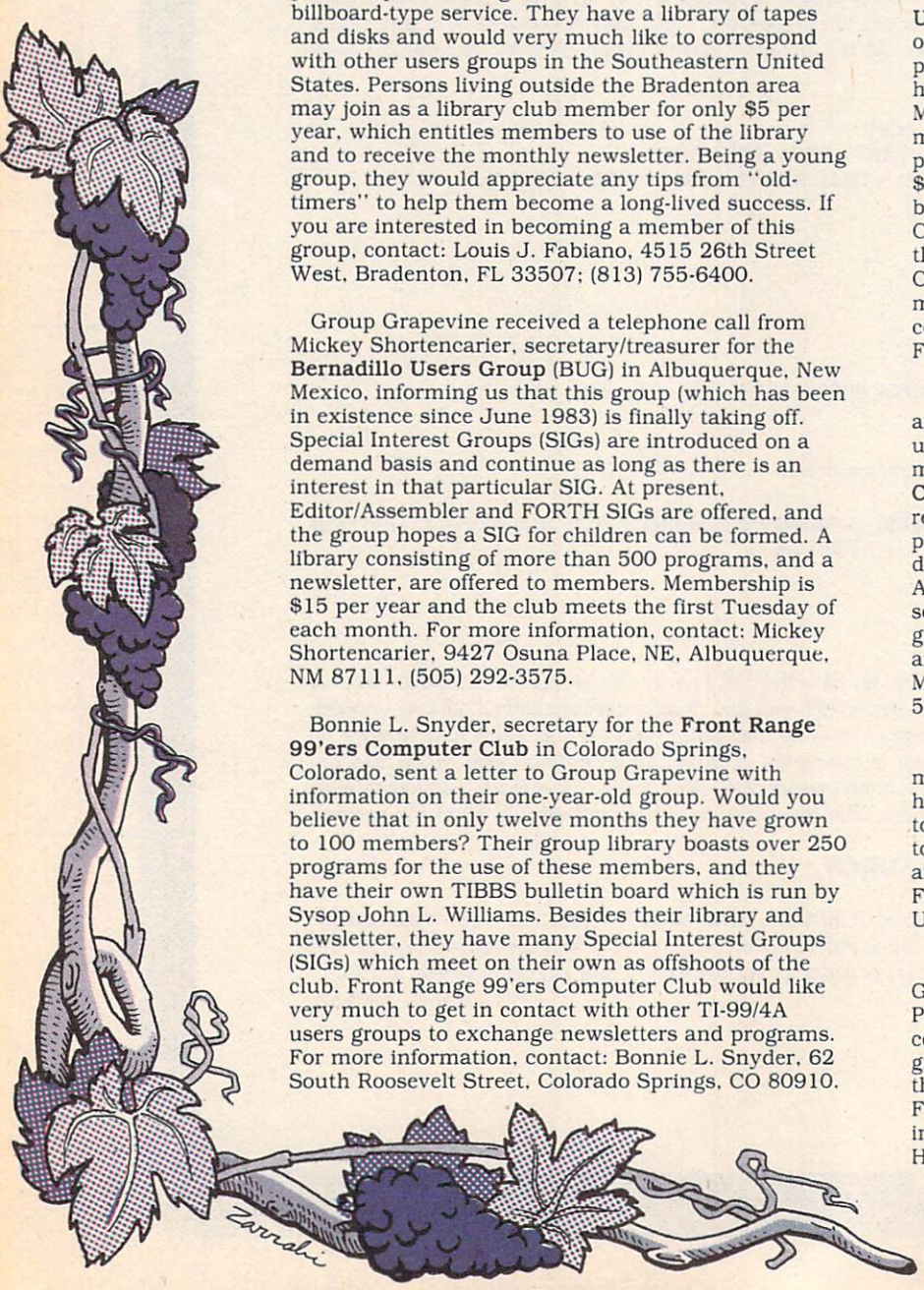
News, information and upcoming events of home computer users groups around the world.

Looking to join a users group, exchange newsletters or software, increase your users group's membership or pep up your next meeting's agenda? For the latest users group news, put your ear to the Group Grapevine. And if you have a message to put out to other groups, if you are starting a new group, or have an interesting item to share, send a note or picture—or better yet, a group newsletter—to the Users Group Editor, Home Computer Magazine, 1500 Valley River Drive, Suite 250, Eugene, OR 97401, (503) 485-8796.

Bradenton, Florida now boasts its own TI-99/4A users group. The **Bradenton Users Group** is made up of over 30 members who are considering the possibility of forming a statewide users council and a billboard-type service. They have a library of tapes and disks and would very much like to correspond with other users groups in the Southeastern United States. Persons living outside the Bradenton area may join as a library club member for only $5 per year, which entitles members to use of the library and to receive the monthly newsletter. Being a young group, they would appreciate any tips from "old-timers" to help them become a long-lived success. If you are interested in becoming a member of this group, contact: Louis J. Fabiano, 4515 26th Street West, Bradenton, FL 33507; (813) 755-6400.

Group Grapevine received a telephone call from Mickey Shortencarier, secretary/treasurer for the **Bernadillo Users Group** (BUG) in Albuquerque, New Mexico, informing us that this group (which has been in existence since June 1983) is finally taking off. Special Interest Groups (SIGs) are introduced on a demand basis and continue as long as there is an interest in that particular SIG. At present, Editor/Assembler and FORTH SIGs are offered, and the group hopes a SIG for children can be formed. A library consisting of more than 500 programs, and a newsletter, are offered to members. Membership is $15 per year and the club meets the first Tuesday of each month. For more information, contact: Mickey Shortencarier, 9427 Osuna Place, NE, Albuquerque, NM 87111, (505) 292-3575.

Bonnie L. Snyder, secretary for the **Front Range 99'ers Computer Club** in Colorado Springs, Colorado, sent a letter to Group Grapevine with information on their one-year-old group. Would you believe that in only twelve months they have grown to 100 members? Their group library boasts over 250 programs for the use of these members, and they have their own TIBBS bulletin board which is run by Sysop John L. Williams. Besides their library and newsletter, they have many Special Interest Groups (SIGs) which meet on their own as offshoots of the club. Front Range 99'ers Computer Club would like very much to get in contact with other TI-99/4A users groups to exchange newsletters and programs. For more information, contact: Bonnie L. Snyder, 62 South Roosevelt Street, Colorado Springs, CO 80910.



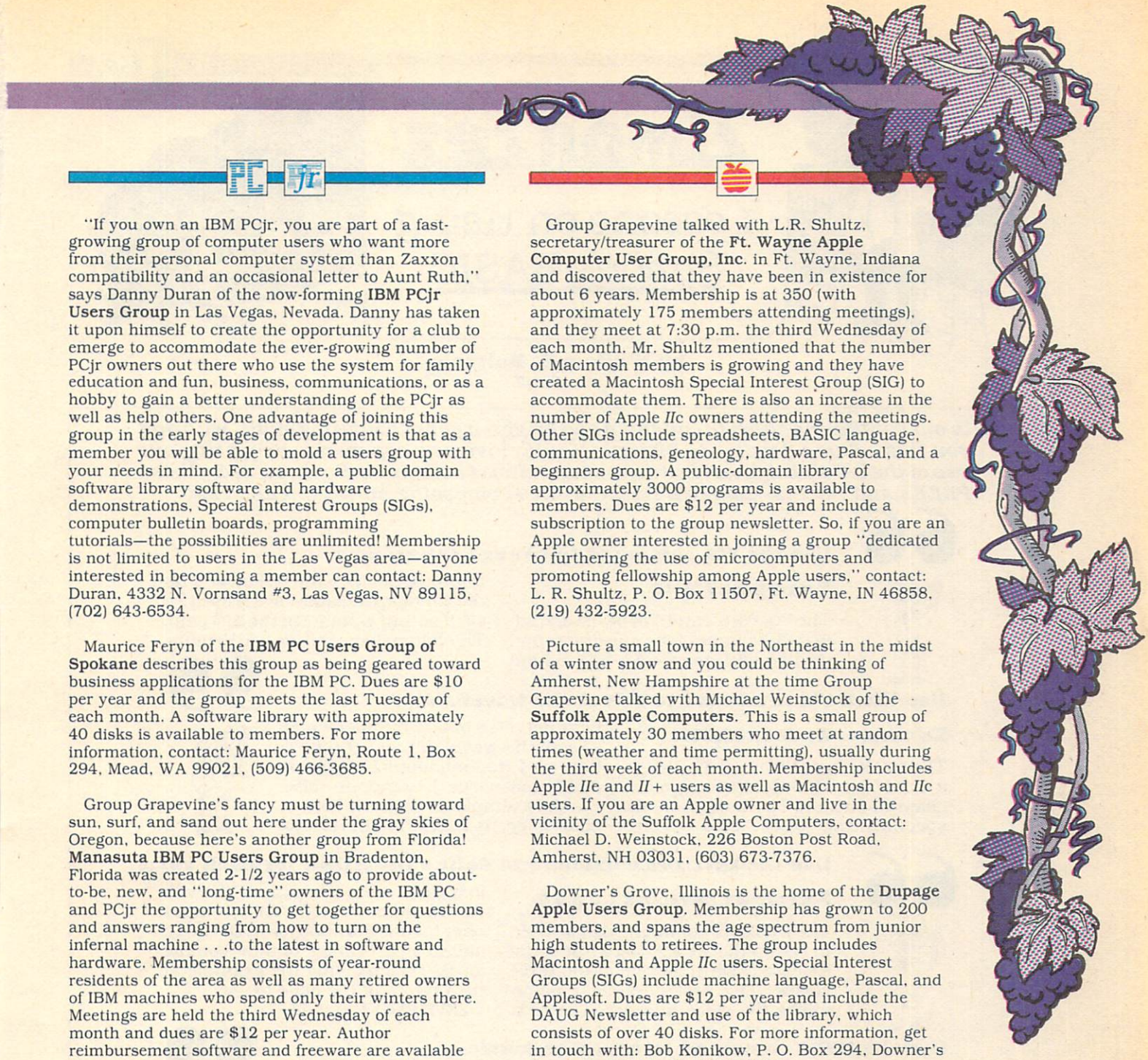
Licking, Missouri is the home of the **Licking Users Group** (LUG). A representative of this group called Group Grapevine to inform TI-99/4A readers of their existence. LUG produces a software catalog of the programs they have in their library, which consists of close to 300 programs in BASIC, Extended BASIC, and some Editor/Assembler languages. The cost of the catalog is $1 and software can be purchased by members for $3 per program and for $4 by non-members (the group provides the cassette). If you are interested in joining this active club, they meet the first Monday of each month. Or, you can contact: LUG, P. O. Box 439, Licking, MO 65542-0439, (314) 674-3922.

Secretary Debbie Johnston of the **Commodore Users Group of Pensacola** would like to "welcome one and all to the world of Commodore." The purpose of this group is to expand each member's horizons regarding their Commodore computers. Meetings are conducted the third Tuesday of each month in the Hygeia Coca Cola Bottling Company plant at 7 p.m. Dues are $1 a month per person and $1.50 a month per family. Members have numerous benefits, which include access to VIC-20 tapes and C-64 disks containing public-domain programs. Also, their library contains various books pertaining to the Commodore computer which are available to paid members. Anyone desiring further information can contact Debbie Johnston, P. O. Box 3533, Pensacola, FL 32516, (904) 455-5804.

Commodore users in the Morrisonville, New York area can now get together with fellow Commodore users on the fourth Thursday of every month at the meetings of the recently formed **Plattsburgh Commodore Users Group**. Demonstrations and reviews of software and hardware, tutorials on programming techniques, and informal open discussions are some of the features this club offers. A newsletter, a "disk of the month" of public-domain software, and a bulletin board are also offered by this group. Dues are $12 per year. For information, anyone can feel free to contact Steve Nolan, 61 East Main Street, Morrisonville, NY 12952, (518) 563-5764.

Tri-County Commodore Users Group is the monicker the Commodore users in Ocala, Florida have chosen for their recently-formed club. According to President Don Vandeventer, the group has grown to more than 45 members in less than six months and now offers a newsletter, "Syntax," to members. For more information, write: Tri-County Commodore Users Group, P. O. Box 1151, Ocala, FL 32678.

Curtis Miller of the **Suncoast 64s** wrote to Group Grapevine with information regarding their group in Palm Harbor, Florida. Their public domain library contains over 1,000 programs on 34 disks and is growing rapidly. This is the group's second year, and they have members as far away as West Germany, France, Italy, and Belgium. For more information, get in touch with: Curtis Miller, 2419 US 19 North, Palm Harbor, FL 33563, (813) 785-1036.



"If you own an IBM PCjr, you are part of a fast-growing group of computer users who want more from their personal computer system than Zaxxon compatibility and an occasional letter to Aunt Ruth," says Danny Duran of the now-forming **IBM PCjr Users Group** in Las Vegas, Nevada. Danny has taken it upon himself to create the opportunity for a club to emerge to accommodate the ever-growing number of PCjr owners out there who use the system for family education and fun, business, communications, or as a hobby to gain a better understanding of the PCjr as well as help others. One advantage of joining this group in the early stages of development is that as a member you will be able to mold a users group with your needs in mind. For example, a public domain software library software and hardware demonstrations, Special Interest Groups (SIGs), computer bulletin boards, programming tutorials—the possibilities are unlimited! Membership is not limited to users in the Las Vegas area—anyone interested in becoming a member can contact: Danny Duran, 4332 N. Vornsand #3, Las Vegas, NV 89115, (702) 643-6534.

Maurice Feryn of the **IBM PC Users Group of Spokane** describes this group as being geared toward business applications for the IBM PC. Dues are $10 per year and the group meets the last Tuesday of each month. A software library with approximately 40 disks is available to members. For more information, contact: Maurice Feryn, Route 1, Box 294, Mead, WA 99021, (509) 466-3685.

Group Grapevine's fancy must be turning toward sun, surf, and sand out here under the gray skies of Oregon, because here's another group from Florida! **Manasuta IBM PC Users Group** in Bradenton, Florida was created 2-1/2 years ago to provide about-to-be, new, and "long-time" owners of the IBM PC and PCjr the opportunity to get together for questions and answers ranging from how to turn on the infernal machine . . .to the latest in software and hardware. Membership consists of year-round residents of the area as well as many retired owners of IBM machines who spend only their winters there. Meetings are held the third Wednesday of each month and dues are $12 per year. Author reimbursement software and freeware are available through the group library, and public domain software will soon be added to their collection of programs. For information on how to join, contact: Richard Reynolds, 2204 Palmasola Boulevard, Bradenton, FL 33529, (813) 792-5400.

Group Grapevine has, of course, heard through the "grapevine" that there are lots and lots of IBM PC users groups out there. But now that PCjr ownership is expanding dramatically, we are expecting a surge in PCjr-specific clubs. So put out the word that Group Grapevine is hungry for information on your club. We've been told many times over that mention in this column gets a group great exposure, helping it attract new members and friends.

Group Grapevine talked with L.R. Shultz, secretary/treasurer of the **Ft. Wayne Apple Computer User Group, Inc.** in Ft. Wayne, Indiana and discovered that they have been in existence for about 6 years. Membership is at 350 (with approximately 175 members attending meetings), and they meet at 7:30 p.m. the third Wednesday of each month. Mr. Shultz mentioned that the number of Macintosh members is growing and they have created a Macintosh Special Interest Group (SIG) to accommodate them. There is also an increase in the number of Apple *IIc* owners attending the meetings. Other SIGs include spreadsheets, BASIC language, communications, geneology, hardware, Pascal, and a beginners group. A public-domain library of approximately 3000 programs is available to members. Dues are $12 per year and include a subscription to the group newsletter. So, if you are an Apple owner interested in joining a group "dedicated to furthering the use of microcomputers and promoting fellowship among Apple users," contact: L. R. Shultz, P. O. Box 11507, Ft. Wayne, IN 46858, (219) 432-5923.

Picture a small town in the Northeast in the midst of a winter snow and you could be thinking of Amherst, New Hampshire at the time Group Grapevine talked with Michael Weinstock of the **Suffolk Apple Computers**. This is a small group of approximately 30 members who meet at random times (weather and time permitting), usually during the third week of each month. Membership includes Apple *IIe* and *II+* users as well as Macintosh and *IIc* users. If you are an Apple owner and live in the vicinity of the Suffolk Apple Computers, contact: Michael D. Weinstock, 226 Boston Post Road, Amherst, NH 03031, (603) 673-7376.

Downer's Grove, Illinois is the home of the **Dupage Apple Users Group**. Membership has grown to 200 members, and spans the age spectrum from junior high students to retirees. The group includes Macintosh and Apple *IIc* users. Special Interest Groups (SIGs) include machine language, Pascal, and Applesoft. Dues are $12 per year and include the DAUG Newsletter and use of the library, which consists of over 40 disks. For more information, get in touch with: Bob Konikow, P. O. Box 294, Downer's Grove, IL 60515, (312) 968-3897.

Schools in the Omaha, Nebraska area seem to be benefitting from the existence of the **Omaha Apple Sauce** users group. Member John Anderson mentioned that this group has gone out into the area's schools to promote computers in education to teachers who are interested in this concept. Membership in this Omaha group is quite large—200 members, with 90 to 100 members attending the meetings (held the first Wednesday of each month). Special Interest Groups are offered encompassing machine language, finances, and Macintosh. Apple *IIc* owners are beginning to make their presence known there. Members can take advantage of an extensive software library, as well as a library containing books and magazines pertaining to Apple machines. For more membership information, contact: John Anderson, 3103 South 32nd Avenue, Omaha, NE 68105, (402) 346-8429.

simon sez:

Lessons on Using Simon's BASIC

by William K. Balthrop
HCM Staff

In our continuing effort to enlighten and brighten your computing skills, we bring you this column devoted to Simon's BASIC. Five new commands help bring out two of the C-64's brightest features: sound effects and music. So throw away those PEEKs and POKEs—Simon's BASIC makes composing music fun and easy.

Use the VOL command to change the volume.

Example: VOL 15

The VOL command can be used to adjust the volume of the music or sound created with the C-64's SID (Sound Interface Device) chip. When you set the volume, the same level is used for each output channel in the SID chip. The volume ranges from 0 to 15 units.

Use the WAVE command to select the waveform.

Example: WAVE 1,10000000

The WAVE command is used to assign a waveform to one of three sound channels (voices). The first parameter specifies the voice. The second parameter is a binary number which selects the waveform, synchronization, and ring modulation. This example tells voice 1 to output noise. For the example given at the bottom of the page, try using: WAVE 1,00010000

Use the ENVELOPE command to define the ADSR for each voice.

Example: ENVELOPE 1,0,6,0,0

The ENVELOPE command can be used to set up the Attack-Decay-Sustain-Release (ADSR) for one of the three voices. The first parameter determines the voice number. The next four parameters set up the ADSR with values from 0 to 15. These values refer to time duration for A, D, and R; and to a percentage of total volume for S. Try changing the example at the bottom of the page to use:

ENVELOPE 1,5,10,10,10

Use the MUSIC command to compose music.

Example: MUSIC 8,"☞SHIFT CLR/HOME☜**1B5**☞F1☜**Z**☞F1☜**C6**☞F1☜**Z**☞F1☜**D6**☞F1☜**Z**☞F1☜☞SHIFT CLR/HOME☜**G"**

The MUSIC command can be used to compose music. You can specify the notes with a single letter such as A or D, or skip a beat with the letter Z. The octave in which the note is to play is set with the number that follows each letter. The function keys between each note specify the note's length.

Use the PLAY command to play music composed with the MUSIC command.

Example: PLAY 2

The parameter following the PLAY command can have a value from 0 to 2; 0 turns off sound; 1 plays music and continues the program only after the music has finished; and 2 continues the program, with the music playing in the background. Now try RUNning this "machine gun":

```
1 VOL15:WAVE 1,10000000:MUSIC
  4,"☞SHIFT CLR/HOME☜1B5☞F1☜Z
  ☞F1☜C6☞F1☜Z☞F1☜D6☞F1☜Z
  ☞F1☜☞SHIFT CLR/HOME☜G"
2 ENVELOPE 1,0,6,0,0:PLAY 2:FOR TD=1
  TO RND(0)*100:NEXT:GOTO 2
```

HCM

HOME COMPUTER™ product news

Each month we publish items of interest and news of recently or soon-to-be released computer products. Our publication of information from manufacturers of computers, peripherals, software, and accessories is not to be construed as product endorsement. Prices quoted are the manufacturers' suggested retail prices and are subject to change.

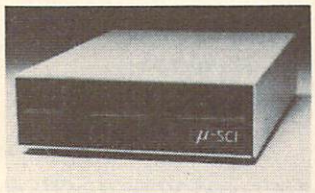
Send press releases to:

Product News Editor
Home Computer Magazine
1500 Valley River Drive., Suite 250
Eugene, OR 97401

❖ ❖ ❖ ❖

Same Power, Smaller Size

Two Half-Height Drives for the Apple



Microsci Corp. has introduced two half-height floppy disk drives, the A.5 and A.5c, that are compatible with the Apple *II* Family of computers. Both machines operate on a direct-drive motor, and each stands less than 2 inches high. The 143K, 35-track A.5 drive can be attached to any Apple disk controller. The A.5c, with the same features as the A.5, was designed to be a second disk drive for the Apple *IIc*, and plugs directly into the *IIc*. The A.5 is $269 and the A.5c is $299.

Microsci Corp.
2158 South Hathaway St.
Santa Ana, CA 92705
(714) 241-5600

❖ ❖ ❖ ❖

The Sound of Music

Music Keyboard Plugs Into C-64

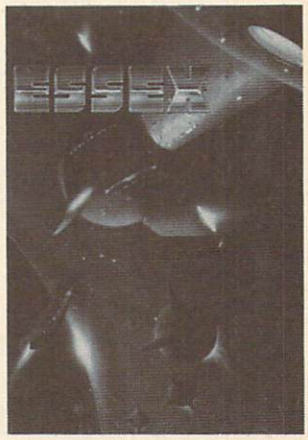
Sequential Circuits, Inc. has introduced the MusicMate, a music keyboard for the Commodore 64 that attaches to the computer through the joystick port. It features full-size playing keys and is polyphonic, allowing users to play 3 notes simultaneously to produce chords. The Model 970 software, included with the $99 package, lets users record up to 10 continuous minutes of their music and play it back, and see a four-color monitor display of the sound values of the Sound Interface Device chip. The sound is adjustable, and a transpose function moves recorded songs up or down in pitch by octaves. Four separate software packages, Song Builder, Song Editor, Song Printer, and Sound Maker allow users to further edit and display their music, and cost $39.95 each.

Sequential Circuits, Inc.
3051 North First St.
San Jose, CA 95134
(408) 946-5240

❖ ❖ ❖ ❖

Write Your Own Adventure Novel

Interactive Programs Aid Novel Writers



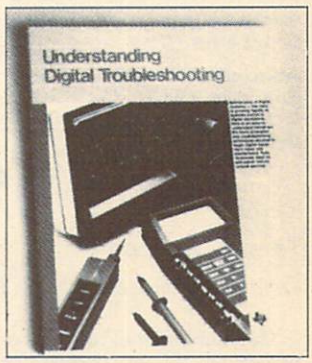
Synapse Software has introduced their series of Electronic Novels for IBM PC, Apple IIe and IIc, and Commodore 64 computer users. Electronic Novels are packaged in hardbound book form—early chapters set the scenes, the story line, and introduce the characters and information relevant to the "journey," which the user completes using floppy disks included in the package. Blank pages are included for map-drawing or notes. The first titles in the series are Mindwheel, a journey into the minds of four deceased people of extraordinary power, and Essex, the story of an intergalactic search-and-rescue mission. The novels are $44.95 for IBM and Apple computers, and $39.95 for Commodore computers.

Synapse Software Corp.
5221 Central Ave.
Richmond, CA 94804
(415) 527-7751

❖ ❖ ❖ ❖

High-Tech Instruction for Beginners

Texas Instruments Issues Book Series



Texas Instruments is releasing its Understanding Series of books, designed for people who want to learn about today's technology. The Series introduces newcomers to timely subjects in science and technology, or gives readers with prior knowledge on the subjects a quick review. Ten titles will be issued in early 1985, and plans are underway to expand the series to 36 books. Included in the first ten books are: Understanding Automation Systems, Understanding Automotive Electronics, Understanding Communications Systems, Understanding Computer Science, Understanding Data Communications, Understanding Digital Electronics, Understanding Digital Troubleshooting, Understanding Microprocessors, Understanding Solid State Electronics, and Understanding Telephone Electronics. These books will retail for $14.95 each.

Texas Instruments Inc.
P.O. Box 225474, MS 8218
Dallas, TX 75265
(214) 997-3926

❖ ❖ ❖ ❖

Get Back to Your Roots

Record Your Ancestry

Roots II, a program that tracks family trees back as many as 99 generations, is now available for the IBM PC and PCjr from Comm-Soft. Information for more than 4000 individuals can be stored and rapidly recalled. Once a family's data has been entered, Roots II prints a family book (up to 999 pages long) containing 4 types of forms or charts, biographical sketches, source documentation, and an alphabetized index of the book's contents. In addition, the program will store maps and pictures of people for display on a monitor. A 220-page manual covers program operation and instructions for starting a search for family roots. The package's price is $99.

Comm-Soft
2452 Embarcadero Way
Palo Alto, CA 94303
(415) 493-2184

❖❖❖❖

Do More Than Before With the C-64

Word Processing, Assembler, & Pascal

A word processor and two utility programs, all for the Commodore 64, compose some of the latest offerings from Abacus Software. Textomat-64 accommodates form letters, chaining of documents, block operations, and a complete printer setup. Horizontal scrolling allows for 80-column line width. Assembler/Monitor 64 assists with the development of machine language programs. The assembler features a fast macro assembler, full screen editing of source programs, and a complete symbol table listing; the monitor has 15 other functions. Pascal-64 is a Pascal compiler and language development package with file-handling capabilities for sequential and relative data management. It also features multi-dimensional arrays, high resolution graphics, and sprites. All three products are priced at $39.95 apiece.

Abacus Software
P.O. Box 7211
Grand Rapids, MI 49510
(616) 241-5510

❖❖❖❖

Tennis, Anyone?

Tennis Simulation With the Pros

Gamestar, Inc. has introduced On-Court Tennis for the Commodore 64. It offers full racquet control over groundstroke angles and spins, serves, lobs, and smashes, and a choice of grass, clay, or hard courts. When playing against one of the computer's 4 superstar challengers, the computer's play intelligence "floats" according to your skill. The real characteristics and playing style of each superstar are incorporated into the opponents. Beating the computer entitles you to join Gamestar's exclusive "Top Seeds" Tennis Club. On Court Tennis retails for $31.95.

Gamestar, Inc.
1302 State St.
Santa Barbara, CA 93101
(805) 963-3487

❖❖❖❖

Getting A Good Return

A Year-End Tax Planning Program

A way to compare the effects of financial decisions on taxes is available with the Tax Command Planner, a program by Practical Programs, Inc. for Commodore 64, Apple *II* Family, and IBM PC systems. Designed for end-of-year tax planning, the program lets the taxpayer quickly try up to 6 different strategies for periods of up to five years simultaneously. For example, it can assist with decisions on how to depreciate equipment, whether to sell stock, or make contributions at the lowest cost. Recalculations are automatically completed when an entry is changed or added. Recently revised tax tables for 1984, including the new rules on income-averaging passed by Congress, have been incorporated into the program. Tax Command Planner for the C-64 is $49.95, for the IBM PC $99.95, and for the Apple *II* Family $79.95.

Practical Programs, Inc.
P.O. Box 93104
Milwaukee, WI 53203
(414) 278-0829

❖❖❖❖

Keyboard With the Works

A Keyboard/Trackball Peripheral

Wico Corp. has released its SmartBoard, a combination keyboard/trackball peripheral for IBM PC and Apple *II*, *II+*, and *IIe* computers. The SmartBoard is fully programmable, allocating 256 bytes to the 10 function keys according to need. The mouse-emulator trackball can be programmed with up to eight characters in any of the 4 primary directions. It comes factory-programmed with both the QWERTY and DVORAK layouts. A type-ahead buffer and N-key rollover are standard, with a serial expansion port also provided. The SmartBoard retails for $399.95, and an adapter for Apple owners is an additional $50.

Wico Corp.
6400 West Gross Point Rd.
Niles, IL 60648
(312) 647-7500

❖❖❖❖

A More Flexible Connection

TI Interconnect Turns Corner

The Peripheral Expansion Box Interconnect by Ten-X Precision is a new device that plugs into the TI-99/4A's I/O port or speech synthesizer and the existing cable, which then plugs in from behind the console. It is designed to alleviate the problems associated with the expansion box's heavy cable and interface connection, preventing accidental disconnection of the system. The Ten-X Peripheral Expansion Box Interconnect is available for machines with or without the speech synthesizer for $43.85.

Ten-X Precision
P.O. Box 163
Concord, CA 94522

❖❖❖❖

Of Magic, Math, and Mayhem

Develop Math and Memory Skills

Math Magic, for ages 4 to 9, and Race The Clock, for ages 5 to 12, are two recent educational program releases from Mindplay. Math Magic is an arcade game that helps children develop counting, addition, and subtraction skills while knocking down walls to make monsters disappear. Race The Clock helps kids sharpen their memory and thinking skills as they race to match pictures and words. Graphics are used to illustrate active verbs. Both programs can be customized by parents or teachers, and are available for $39.99 each. They run on the IBM PC and PCjr as well as Apple *II* Family computers.

Methods & Solutions, Inc.
82 Montvale Ave.
Stoneham, MA 02180
(617) 438-5454

❖❖❖❖

It's All In the Wrist

A New Joystick for the C-64 & 99/4A

Suncom has designed a new joystick for TI-99/4A and Commodore 64 computers. The TAC-3, which stands for Totally Accurate Controller includes three fire buttons: two buttons on the base, and one on top of the handle. The joystick has a two-year warranty, and costs $14.95.

Suncom
260 Holbrook Dr.
Wheeling, IL 60090
(312) 459-8000

❖❖❖❖

Make Your Home Your Castle

Design Your Own Home and Yard

Design Your Own Home is a new, three-package series of programs by Avant-Garde Publishing Corp. for Apple *II* Family computers. Architectural Design ($99.95) offers 126 individual detail shapes that can be rotated and arranged in any way. The program calculates distances, diagonals, and angles, and lets the user construct floor plans, top views and side views of each creation. Interior Design ($69.95) allows the user to move scaled down furniture around to discover where it will best fit, choose multi-colored patterns for wallpaper or fabric designs, and look at arrangements from the top and side. Landscape Design ($69.95) lets the user draw an outline of home and property lines, then arrange flowers, trees, and shrubs around the house or property. Young plants can be aged to preview how they will look when grown, and a "paintbrush" graphics utility allows more artistic expression. Also, north, south, east, and west perspectives are provided, as well as side and aerial views.

Avant-Garde Publishing Corp.
P.O. Box 30160
Eugene, OR 97403
(503) 345-3043

❖❖❖❖

Wage An Historic Battle

Simulate The Ancient Art of War

The Ancient Art of War, Broderbund Software's new strategy game for the IBM PC and PCjr, contains 11 built-in war campaigns from history, each fought by barbarians, archers, and knights from the pre-gunpowder era. The program also features a game generator which allows users to design their own battles—right down to the condition of the troops and the difficulty of the terrain. Action takes place in real time, and a zoom feature allows the user to see close-ups of battles and issue commands that the troops carry out in detailed animation showing individual soldiers. It retails for $44.95.

Broderbund Software
17 Paul Dr.
San Rafael, CA 94903-2101
(415) 479-1170

❖❖❖❖

In Search of the Lost Rules

Indiana Jones Meets the Home Computer

Indiana Jones in the Lost Kingdom, developed by Mindscape Inc. in conjunction with Lucasfilm, Ltd., is a unique game for the Commodore 64—it has no rules. Players must first find out what the puzzle is and then solve it for each of six different rooms. The objective is to retrieve an artifact containing the complete history and knowledge of a lost, forgotten civilization. Frustrated players can consult the manual for encrypted clues or dial a telephone "hotline." The game retails for $29.95.

Mindscape Inc.
3444 Dundee Rd.
Northbrook, IL 60062
(312) 480-7667

❖❖❖❖

Do-It-Yourself Investment Planning

An Investment and Statistical Program

Programmed Press has announced that its Investment and Statistical Software package, containing 50 programs for statistical forecasting, stocks, bonds, options, futures, and foreign exchange, is ready for the Commodore 64. The software also supports Apple and IBM personal computers. A 220-page Computer-Assisted Investment Handbook by Dr. Albert Bookbinder lists, explains, and gives sample RUN illustrations for all 50 of the BASIC programs. The Investment and Statistical Software package is priced at $100, and the Handbook is $19.95.

Programmed Press
2301 Baylis Ave.
Elmont, NY 11003
(516) 775-0933

❖❖❖❖

An Assembly Language Guide

TI Programmers Get E/A & MM Help

Learning TI-99/4A Home Computer Assembly Language Programming by Ira McComic includes the basic concepts of assembly language programming and the structure of TI's existing programs. Information is also provided about the Editor/Assembler Package, and the line-by-line assembler and debugger that are included in the Mini Memory module. Program examples accompany the guide's tutorial material. The book retails for $16.95.

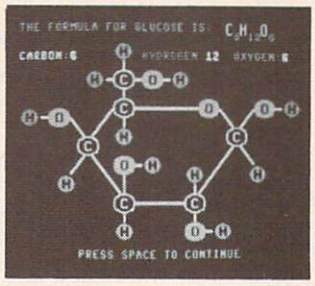
Wordware Publishing, Inc.
1104 Summit Ave. Suite 104
Plano, TX 75074
(214) 423-0090

❖ ❖ ❖ ❖

Light Up Your Learning

Science Programs for the C-64

Tech Sketch, Inc. has introduced 3 new high-school level science-education programs for the Commodore 64. Each of the new titles is the first in a series planned for three subject areas: biology, chemistry, and the structure of matter. The first programs are Structure of Leaves, Molecules and Atoms, and Passive Transport, which explains how substances move through membranes. The new series



of programs is compatible with Tech Sketch's line of light pens.

Tech Sketch, Inc.
26 Just Rd.
Fairfield, NJ. 07006

❖ ❖ ❖ ❖

Home Nutritional Planning

Track Nutrients in the Foods You Eat

Three new packages designed to track 15 nutrients in more than 800 foods have been released by Nutritional Data Resources. Nutrient Goal Graph 15-1 analyzes 15 nutrients for one day of food intake and graphically compares them to the Recommended Dietary Allowance goals by sex, age, and weight. Nutrient Calculator 15 analyzes the nutritional content of recipes, menus, and daily food intake. Nutrient Trackor 15 allows the user to select 20 foods from the data base and provides an analysis of 15 nutrients in

each of those foods. The programs will run on the IBM PC and PCjr, and the Apple *IIe* with 80-column card. Prices range from $40 to $60 each.

Nutritional Data Resources
P.O. Box 540
Willoughby, OH 44094
(216) 951-6593

❖ ❖ ❖ ❖

A Turtle Learning Guide

BASIC and Story Structure for Kids

Telly Turtle, a turtle graphics program based on the LOGO language, is one of two new educational programs from Hayden Software Company. Telly Turtle helps children as young as 4 understand BASIC programming through the use of icons and graphic creations. It is available for the Commodore 64 for $29.95. Adventures in Colorland: Space Sagas was created to provide structure to story writing for 6 to 10 year olds. Once a child chooses one of four stories, selects the main character, the space ship for travel, and the means to solve the proposed problem, the choices come to life, allowing the child to see how his or her choices affected the story. It is priced at $29.95 for the Commodore 64, and will soon be available for the Apple *II* Family and the IBM PCjr.

Hayden Software Company, Inc.
600 Suffolk St.
Lowell, MA 01854
(617) 937-0200

❖ ❖ ❖ ❖

Removing Language Barriers

International Characters Available

IsraComp has released four new programs for the Commodore 64. HebrewWriter II is a Hebrew word processor that prints and displays Hebrew from either the HebrewWriter's right-to-left environment or from user-written programs. Editing and storage is also possible, and the $19.95 program will print on any C-64 dot matrix printer. Creatagraphics allows a user to create and store character sets (graphic, foreign alphabets, etc.) and sprites. The Creatagraphics package includes Hebrew, Arabic, Greek, Russian, and Korean character sets, and will print in both directions. It sells for $24.95. IsraQuiz and Maps & Facts require the user to identify geographical entities and answer randomly-generated questions regarding the geography and politics of Israel (in IsraQuiz), and the world (in Maps & Facts). Both IsraQuiz and Maps & Facts are $9.95 apiece.

IsraComp
P.O. Box 1091
King of Prussia, PA 19406
(215) 386-0408

❖ ❖ ❖ ❖

Professor Pixel Holds Class

Create Graphics and Sound Effects

Individual Software, Inc. has released Professor Pixel, a program that teaches users how to use the BASIC programming language to create their own graphics, melodies, and sound effects, and then bring it all to life with color animation. Professor Pixel is menu-driven and uses non-technical dialogue and interactive exercises for all ages and ability levels. It runs on IBM PC and PCjr computers, and costs $59.95.

Individual Software Inc.
1163-I Chess Dr.
Foster City, CA 94404
(415) 341-6116

❖ ❖ ❖ ❖

Think In TinkTonk Land

Series Features Fantasy Characters

Mindscape, Inc. has released four new programs from their Tink! Tonk! series, which is part of their Sprout line of educational software for children ages 4 through 8. In Tink's Adventure, Tink leads children through a search for treasure as they learn the alphabet. Tonk In The Land Of Buddy-Bots consists of five games that help children develop concentration and critical thinking skills. By helping Tinka travel through TinkTonk Land in Tinka's Mazes, kids develop basic math concepts and solve addition problems. Tuk Goes To Town is an interactive story featuring five different games designed to teach spelling, build vocabulary, and strengthen reading skills. These programs retail for $39.95 each, and will run on Apple, IBM, and Commodore computers.

Mindscape, Inc.
3444 Dundee Rd.
Northbrook, IL. 60062

❖❖❖❖

A Striking Offense

Realistic Air-Combat Simulation

Two new programs from MicroProse Software, F-15 Strike Eagle and MIG Alley Ace, simulate modern air combat from an electronic cockpit. F-15 Strike Eagle includes many of the real plane's flight weapons and information systems to use in 7 realistic missions. The Combat Environment has enemy aircraft, radar and infra-red guided missiles, air-to-air missiles, and ground targets. The F-15's defenses include electronic counter measures, after burners, flares, and surface-to-air missile launch indicators. MIG Alley Ace operates in three-dimensional airspace where the pilot must take into account airspeed, turn rates, gravity, and relative position in 3 axes, as well as make tradeoff decisions on the proper time to attack, defend, or escape. Both programs run on the Commodore 64, with versions of F-15 Strike Eagle to be released soon for Apple and IBM PC and PCjr systems. Each retails for $34.95.

MicroProse Software
10616 Beaver Dam Rd.
Hunt Valley, MD 21030
(301) 667-1151

❖❖❖❖

Helping At Home

Data Base & Reading Aids for 99/4A

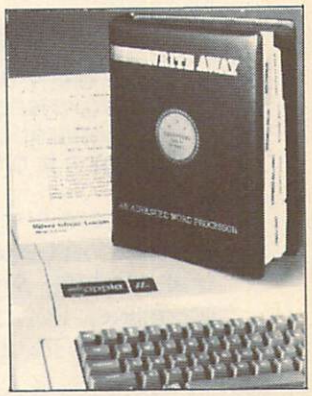
Four new programs for the TI-99/4A have been released by Navarone Industries, Inc. Data Base Management System for home and small businesses ($69) keeps track of inventory, customer files, and even personalized form letters. Users can create data bases with up to 35 fields in 32,000 records. The System requires a disk drive. Homework Helper for children 8 years and older features a built-in 20,500 word spelling checker with its word processor. The $49.95 program includes a standard format for book reports and class projects. Homework Helper requires a disk drive. Speed Reading, complete with workbook, is designed to improve reading speed and comprehension in versions for both children and adults. Speed Reading is a cartridge, and needs no other equipment. It is $49.95. The Console Writer cartridge (49.95) allows those without expanded memory and peripheral equipment to do word processing with just the console and a printer. It features a lower-case character set for easy reading, and a full-screen text editor.

Navarone Industries, Inc.
510 Lawrence Expressway #800
Sunnyvale, CA 94086
(408) 866-8579

❖❖❖❖

It's In The Mail . . .

Electronic-Mail Word Processing



Write Away, Midwest Software Associates, Inc.'s word processing system for the Apple *II* Family of computers, has been enhanced to accommodate electronic mail. In addition to creating, editing, and printing documents, a user can now transmit them with Terminal, a terminal communications program that has been integrated into the Write Away system. It offers these functions: auto dial, auto answer, read file from disk, write file to disk, transmit file, capture file, full duplex, half duplex, and unattended operations. The Write Away system also includes form letter and mailing list support; logical operators and conditional text features; data base file reading; block delete, move, or copy; and soft hyphenation. It is priced at $175.

Midwest Software Associates, Inc.
1160 Appleseed Ln.
St. Louis, MO 63132
(314) 997-2369

❖❖❖❖

Want to Get Published?
Fame, Fortune, Recognition!
See Your Name in Print!

Home Computer Magazine is looking for articles and programs in all areas of interest relevant to Apple, IBM, Commodore, and Texas Instruments home computers. Here are some of the kinds of material we would like you to submit:

Software

Have you written any programs in the areas of home productivity, education, or entertainment? Perhaps you've created unique software to help monitor personal finances, or a new contribution to computer-assisted instruction (CAI). Maybe you have an unusual new game—or a routine that makes certain computer operations easier to perform. Don't be shy. Even if you think your piece is "unpolished," it may still be a good idea. We will be glad to follow through with your concept—enhancing the program and converting it to work on the other machines we cover.

Product Reviews

Have you recently purchased a piece of hardware or software that hasn't come up to your expectations—or has, on the other hand, impressed you with its performance? We're looking for comprehensive product reviews from different perspectives.

Hardware Tips

Perhaps you've modified your microcomputer or have interfaced it with some unique or useful hardware. Send us your how-to-do-it story, complete with photos and/or diagrams.

Tutorials

Many of our articles are purely instructional. If you have extensive experience in some area of programming or other computer application, put your specialized knowledge down on paper and let us pass it on to our readers.

These are just some ideas. Perhaps you have others. If you're not a professional writer, don't worry. Our friendly editorial staff stands ready to help you polish your manuscripts. And we'll be more than happy to send you a copy of our author guidelines. Here are some comments from happy writers who have already published their work in our magazine:

"The people at Home Computer Magazine are fun to work with. And it's sure nice to get paid for writing about my favorite subject."
—Patricia Swift
Author of "Multiplan Medium" and other articles

"The artwork and layout are creative and contribute a lot to the presentation of my articles."
—Roger Kirchner
Author of "Missionary Impossible" and other articles

"It was gratifying to finally see my name in print after all the work I've done on my computer."
—Brian Lee
Author of "Market Madness"

"I was extremely impressed with the way my program was printed in HCM. It was very interesting to see the way the program was translated into the languages of the other popular computers and to read the comments of the people who reviewed the program. Truly a first class job! Thank you!"
—Craig Blazakis
Author of "Bird Brain"

"I was very pleased with the final presentation of my article. It is gratifying to see such judicious handling of an outside submission. The HCM staff fixed a program bug and expanded the application of the article to three other computers, while preserving the style of the article as submitted. The illustration added to the overall readability."
—Andrew Keith
Author of "Build a LOGO Adventure"

Please send your double-spaced, typed or printed manuscripts, photos, and disks or cassettes (recorded on both sides) if the article includes program material, to:

Attn: Editorial Submissions
Home Computer Magazine
1500 Valley River Drive, Suite 250
Eugene, OR 97401

HOME COMPUTER™
PROGRAM LISTINGS

CONTENTS

PROGRAM LISTING

Your Guide to Typing in Programs from *HCM*

Within these pages is a software bonanza: entertainment, education, home and business applications, utilities, and tutorials—just for you. All you need to do is type them into your computer. *HCM* has taken most of the strain out of this process:

- Typeset listings with numbers in boldface.
- A bold, double vertical bar separating the line numbers from the program statements in BASIC listings.
- A vertical background grid to aid entry of the spaces.

Look at the Key-in-Reference (Figure 1 below) see how each character actually appears in the listing. By checking any questionable characters with the Key-in Reference, you can reduce errors to a minimum.

Figure 1: Key-in Reference

Before You Begin

Since *HCM* publishes for several different computers, the first thing you should do is make sure that you are looking at the listing designed for your machine. If, for example, you have an Apple IIe, make sure you look for the following black bar above the listing: APPLE II Family The computer model name will likewise appear on each subsequent page of each listing, so always look for the name before you begin typing from a new page of listings.

Before you begin typing in the program, you will want to set up a system to save your program. Whether you are using a cassette or diskette storage system, now is the time to be certain it is properly connected, powered up, and loaded with a blank cassette or an initialized disk. As you type in your program, you should get in the habit of saving your work after every twenty or so lines.

One of the most common errors in entering a listing is typing one symbol for another. These transpositions include substituting the letter O for the number 0, the letter l for the number 1, the letter S for the $, and the uppercase B for the number 8. The last error is especially likely when working in hexadecimal numbers which are composed of 0-9 and the uppercase letter A-F.

The listings in *HCM* are always the same number of characters wide, but the number of characters put on any line of the video display will vary from computer to computer. Don't try to make your listings *look like* the type-set listing—instead make sure you key in the listings character for character and space for space.

A Special Note on C-64 Listings

Commodore uses more than 90 special symbols to represent various keyboard operations: for instance, the symbol [♥] in a program represents the operation of holding down the [SHIFT] key and pressing the key which has CLR on its upper half (second key from the right on the top row). This operation clears the screen.

Rather than reproducing these symbols, *HCM's* listings include key-stroke instructions, between two hands with pointing fingers. For example, when you find SHIFT CLR in an *HCM* listing, you will know to hold down the [SHIFT] key and press the key with CLR on it.

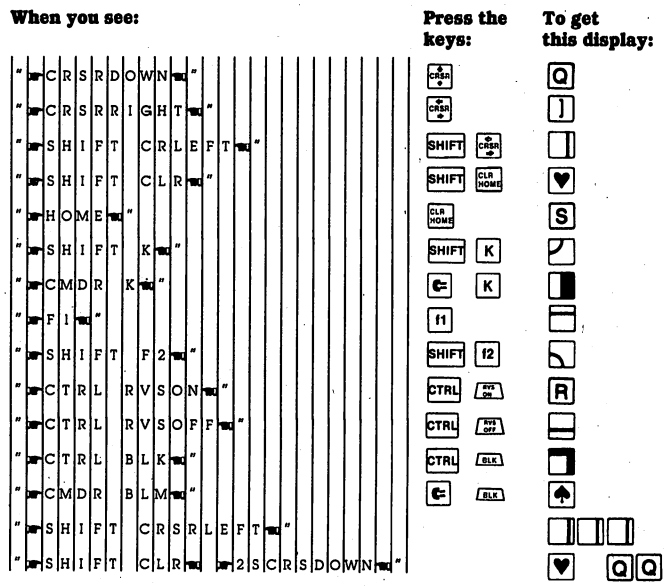
A number is included if you need to repeat the operation: 8SHIFT CRSRLEFT tells you to hold the [SHIFT] key down and press the cursor left key (on the bottom right of the keyboard) eight times.

When you come to the hand symbols, remember:

- Each operation is enclosed in its own set of hand symbols.
- If any key action requires you to press two keys, press the control key or the Commodore key or the shift key *first* and hold it down before pressing the second key.
- Everything between a pair of hand symbols is set in a different place.

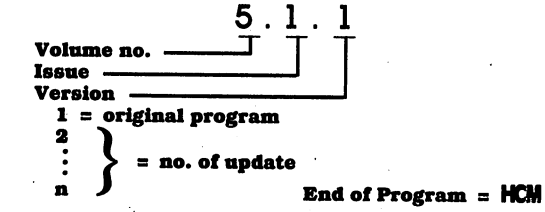
In Figure 2 below, we have included a chart showing you a representative sample of the symbols that appear when you use keystrokes enclosed by the hand symbols. (Notice that the hand symbols always appear within quotation marks—as in a print statement.)

Figure 2: C-64 Special symbols

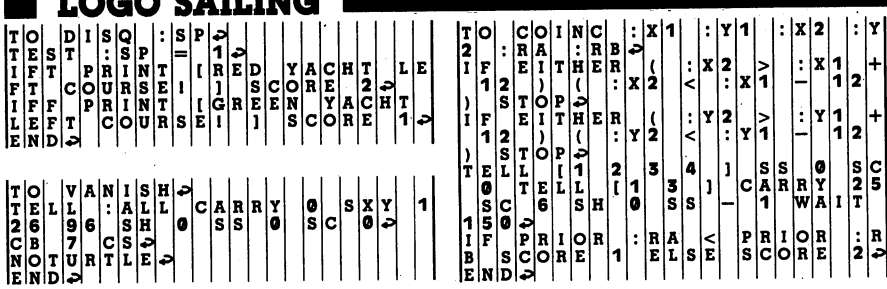


Program Identification

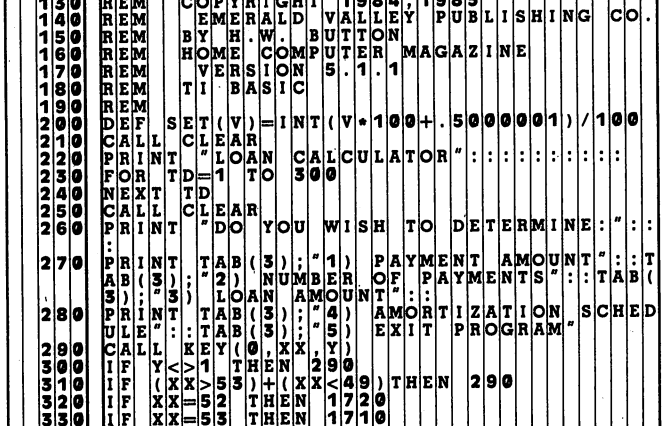
Each program header (the first few lines of the program) contains information giving the language the program is written in (e.g., TI Extended BASIC, Applesoft, etc.) and any special system components that are required (special memory cards, Speech Synthesizer, etc.). The first two digits of the version number tell you in which volume and issue of *HCM* the program *initially* appeared. The third digit of the version number indicates the version of the program. When a program initially appears, in *HCM*, it is version 1. Any subsequent revisions to the program if later published in the magazine or in the software available on magnetic medium from *HCM* will bear a revised version number.

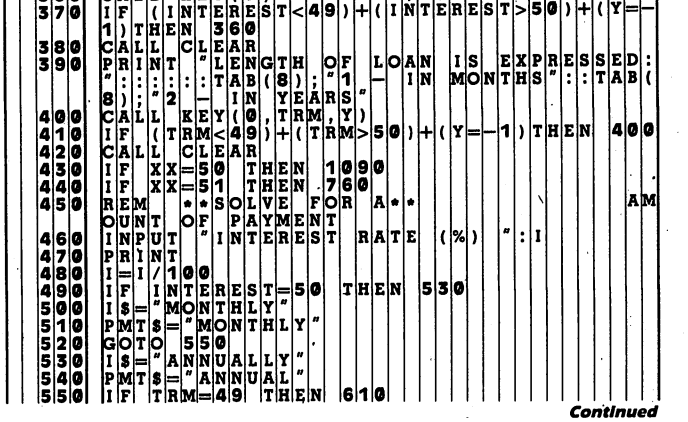


LOGO SAILING



LOAN CALCULATOR





Continued

HCM

PROGRAM LISTING

LOAN CALCULATOR *Continued*

```
570 PRINT
580 TERM=N
590 TERM$="YEARS"
600 GOTO 660
610 INPUT "MONTHS OF LOAN ":N
620 PRINT
630 TERM=N
640 N=N/12
650 TERM$="MONTHS"
660 INPUT "AMOUNT OF LOAN $":T
670 CALL CLEAR
680 IF INTEREST=50 THEN 710
690 N=N*12
700 I=I/1200
710 A=(I*((1+I)^N))/(((1+I)^N)-1)*T
720 IF INTEREST=50 THEN 740
730 I=I*1200
740 I=I*100
750 GOTO 1520
760 CALL CLEAR
770 REM **SOLVE FOR T** TOT
    AL LOAN AMOUNT
780 INPUT "INTEREST RATE (%) ":I
790 PRINT
800 I=I/100
810 I$="ANNUALLY"
820 IF INTEREST=50 THEN 860
830 I=I/12
840 I$="MONTHLY"
850 GOTO 890
860 INPUT "ANNUAL PAYMENT $":A
870 GOTO 900
880 PRINT
890 INPUT "MONTHLY PAYMENT $":A
900 PRINT
910 IF TERM=50 THEN 980
920 INPUT "HOW MANY MONTHS? ":N
930 TERM=N
940 TERM$="MONTHS"
950 IF INTEREST=49 THEN 1030
960 N=N/12
970 GOTO 1030
980 INPUT "HOW MANY YEARS? ":N
990 TERM=N
1000 TERM$="YEARS"
1010 IF INTEREST=50 THEN 1030
1020 N=N*12
1030 PRINT
1040 T=(A*(((1+I)^N)-1))/(I*((1+I)^N))
1050 IF INTEREST=50 THEN 1070
1060 I=I*12
1070 I=I*100
1080 GOTO 1520
1090 CALL CLEAR
1100 REM **SOLVE FOR N** NUM
    BER OF PAYMENTS
1110 INPUT "INTEREST RATE (%) ":I
1120 I=I/1200
1130 IF INTEREST=49 THEN 1180
1140 I$="ANNUALLY"
1150 PRINT "ANNUAL PAYMENT $":A
1160 INPUT
1170 GOTO 1220
1180 I$="MONTHLY"
1190 I=I/12
1200 PRINT "MONTHLY PAYMENT $":A
1210 INPUT
1220 PRINT "AMOUNT OF LOAN $":T
1230 CALL CLEAR
1240 IF T*I<A THEN 1300
1250 PRINT "INSUFFICIENT PAYMENT TO
    COVER INTEREST"
1260 FOR DELAY=1 TO 1000
1270 NEXT DELAY
1280 GOTO 250
1290 L=A/(A-(T*I))
1300 L1=1+I
1310 LL=LOG(L)
1320 LL=LOG(L1)
1330 N=LL/LL
1340 I=I*100
1350 IF INTEREST=50 THEN 1450
1360 I=I*12
1370 IF TERM=50 THEN 1420
1380 TERM$="MONTHS"
1390 TERM=N
1400 GOTO 1520
1410 TERM$="YEARS"
1420 TERM=N/12
1430 GOTO 1520
1440 IF TERM=49 THEN 1490
1450 TERM$="YEARS"
1460 TERM=N
1470 GOTO 1520
1480 TERM$="MONTHS"
1490 TERM=N*12
1500 REM **CALCULATE**
    FINAL REPORT
1510 CALL CLEAR
1520 PRINT "INTEREST RATE: ";I;"%"::"CO
    MPOUNDED";I$
1530 PRINT "LOAN AMOUNT:";TAB(17);T
1540 IF INTEREST=49 THEN 1590
1550 PRINT "ANNUAL PAYMENT";TAB(17);SET(
    A)
1560 GOTO 1600
1570 PRINT "MONTHLY PAYMENT";TAB(17);SET
    (A)
1580 PRINT "NO. OF PAYMENTS";TAB(17);SET
    (N)
1590 PRINT "TERM IN ";TERM$;TAB(17);SET(
    TERM)
1600 TP=A*N
1610 PRINT "TOTAL COST";TAB(17);SET(TP)
1620 TPI=TP-T
1630 PRINT "TOTAL INTEREST";TAB(17);SET(
    TPI)
1640 PRINT :::"TO CONTINUE PRESS ANY KEY"
1650 CALL KEY(0,X,Y)
1660 IF Y<>1 THEN 1670
1670 IZ$="END SESSION"
1680 GOTO 250
1690 END
1700 CALL CLEAR
1710 REM AMORTIZATION
1720 SCHEDU++LE ++
1730 INPUT "LOAN AMOUNT? $":LOAN
1740 INPUT "NO OF MONTHLY PAYMENTS? ":N
1750 INPUT "INTEREST RATE? (%)":IN
1760 IN=IN/1200
1770 PMT=LOAN*(IN/(1-(1+IN)^(-N))))
1780 TOTSET(PMT*N)
1790 TOTPAY=SETT(PMT)
1800 LASTPT=TOT-PAY*(N-1)
1810 PRINT "MONTHLY PAYMENT =$";PAY
1820 PRINT "FINAL PAYMENT =$";LASTP
1830 INPUT "SHOW SCHEDULE FROM
    PAYMENT #":STRT
1840 INPUT "TO PAYMENT #":STP
1850 IF STP<>N THEN 1930
1860 IFLGP=N-1
1870 IF LGP=N-1 THEN 1940
1880 PRINT "PRESS ANY KEY TO SEE NEX
    T";
1890 FOR Z=STRT TO STP
1900 K=(1+IN)^(-Z)
1910 BAL=(PMT*(K-1))/IN+LOAN)
1920 PRINT "PAYMENT #";Z
1930 PRINT BAL-PAY
1940 NEXT Z
1950 GOTO 1940
1960 PRINT "INTEREST= $";SET(I):"PRINCIP
    AL=$";SET(PR)
1970 PRINT "BALANCE= $";SET(BAL)
1980 CALL KEY(0,K,S)
1990 IF S<>1 THEN 2030
2000 IF FLG=0 THEN 2130
2010 PRINT "PAYMENT #";N
2020 BAL=0
2030 PRINT "INTEREST= $";SET(I):"PRINCIP
    AL=$";SET(PR)
2040 PRINT "BALANCE= $";SET(BAL)
2050 CALL KEY(0,K,S)
2060 IF S<>1 THEN 2130
2070 IZ$="END SESSION"
2080 GOTO 250
```

HCM

SET.THE.DATE

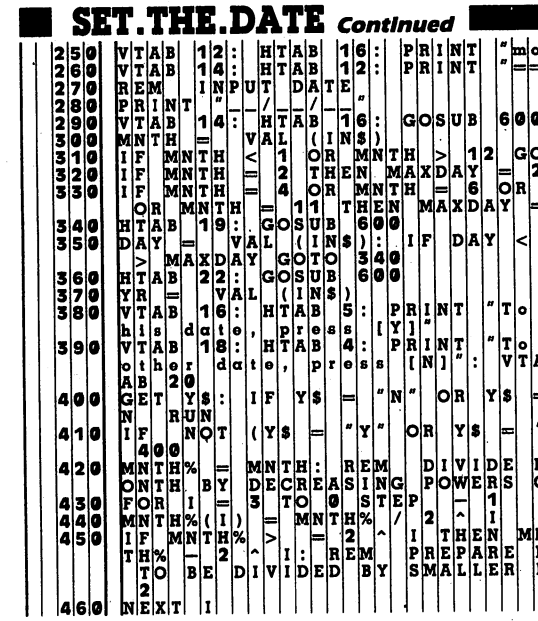
APPLE // Family

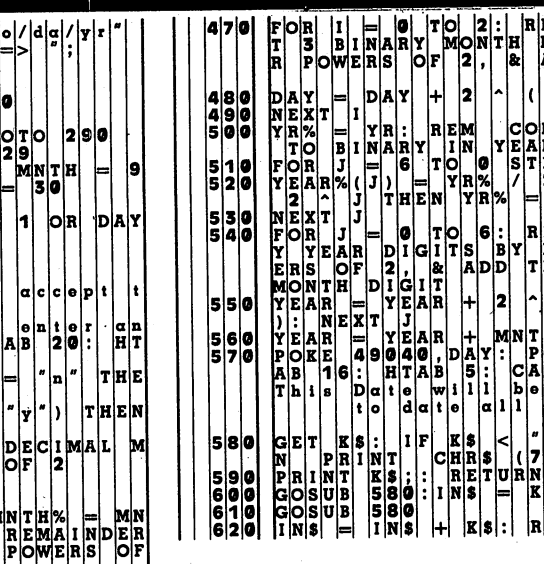
```
100 REM **********
110 REM * SET.THE.DATE *
120 REM **********
130 REM COPYRIGHT 1984,1985
140 REM EMERALD VALLEY PUBLISHING CO.
150 REM BY ANDERS NEREIM
160 REM HOME COMPUTER MAGAZINE
170 REM VERSION 5.1.1
180 REM APPLE II FAMILY APPLESOFT
190 REM PRODOS ONLY
200 YEAR=0:MAXDAY=31
210 DIM MNTH$(3),YEAR$(6):HOME:INVER
    SE
220 PRINT "THIS IS A PRODOS DATE-SETTI
    NG PROGRAM:":NORMAL
230 TAB 8:HTAB 7:CALL -958:PRINT "
    Enter date to mark files"
240 VTAB 10:HTAB 5:PRINT "using only
    two digit numerals:"
```

Continued

SET.THE.DATE *Continued*

APPLE // Family

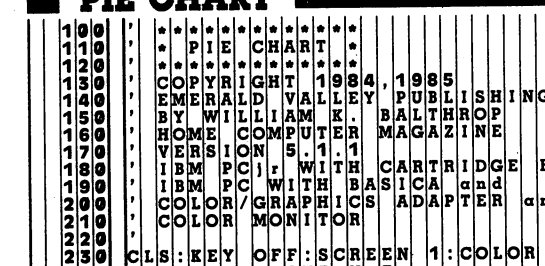


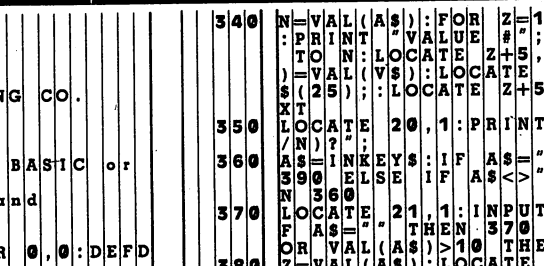


HCM

PIE CHART

IBM PC & IBM PCjr

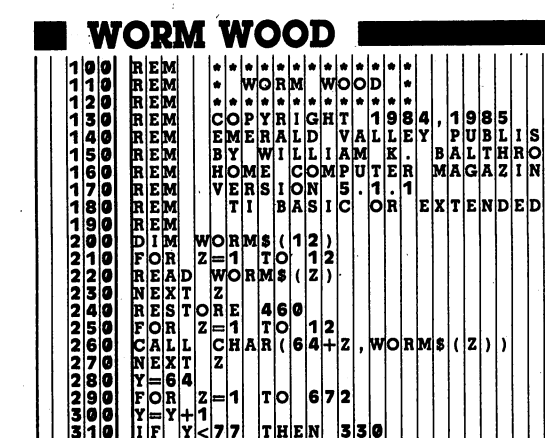


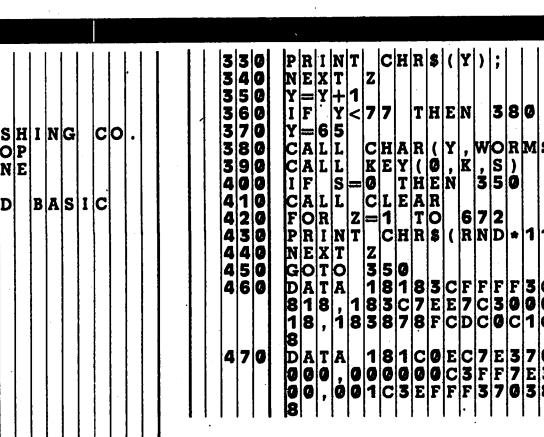


HCM

WORM WOOD

TI-99/4A





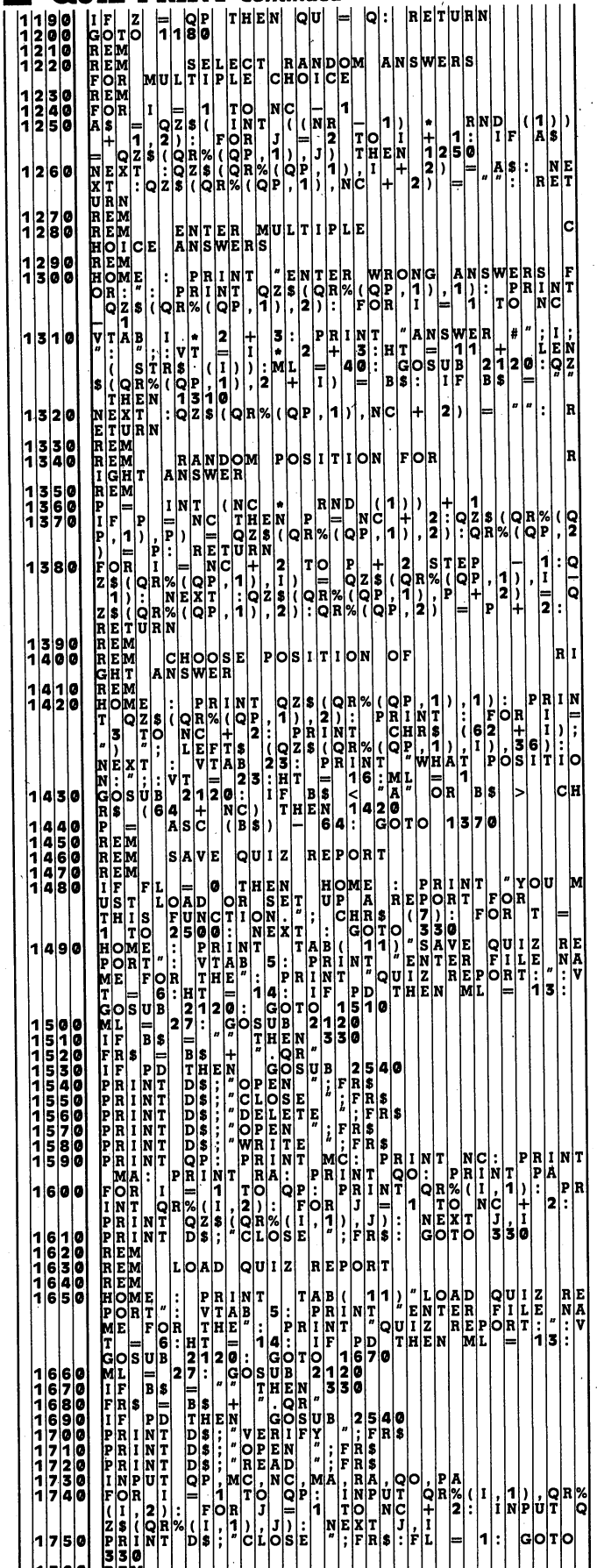
HCM

QUIZ-PRINT


```
100 REM  **************
110 REM  * QUIZ-PRINT *
120 REM  **************
130 REM  COPYRIGHT 1984,1985
140 REM  EMERALD VALLEY PUBLISHING CO
150 REM  BY WILLIAM K. BALTHROP
160 REM   AND THE HCM STAFF
170 REM  HOME COMPUTER MAGAZINE
180 REM  VERSION 5.1.1
190 REM  APPLE II FAMILY APPLESOFT
200 REM    INITIALIZE PROGRAM--DISPLAY TITLE SCREEN
210 REM
220 ONERR  GOTO 2360
230 MX = 250: DIM QZ$(MX,12),QR%(MX,2),EM$(1,21): FOR I = 0 TO 1: FOR J = 1 TO 21: READ EM$(I,J): NEXT J: NEXT I
240 DATA  LANGUAGE NOT AVAILABLE,RANGE ERROR,RANGE ERROR,WRITE PROTECTED,END OF DATA,FILE NOT FOUND,VOLUME MISMATCH,I/O ERROR,DISK FULL,FILE LOCKED,DOS SYNTAX ERROR
250 DATA  NO BUFFERS AVAILABLE,FILE TYPE MISMATCH,PROGRAM TOO LARGE,NOT DIRECT COMMAND,SYNTAX ERROR,"","","","",""
260 DATA  "","",RANGE ERROR ,NO DEVICE CONNECTED,WRITE PROTECTED,END OF DATA,PATH NOT FOUND,PATH NOT FOUND,I/O ERROR,DISK FULL,FILE LOCKED,INVALID OPTION
270 DATA  NO BUFFERS AVAILABLE,FILE TYPE MISMATCH,PROGRAM TOO LARGE,NOT DIRECT COMMAND,SYNTAX ERROR,DIRECTORY FULL,FILE NOT OPEN,DUPLICATE FILENAME,FILE BUSY,FILE(S) STILL OPEN
280 HOME : VTAB 12: HTAB 16: PRINT "QUIZ PRINT":PD = 0: IF PEEK (48905) = 76 AND PEEK (48911) = 0 THEN PD = 1
290 GOSUB 2090
300 REM
310 REM  MAIN MENU SELECTION
320 REM
330 D$ = CHR$ (4): HOME : VTAB 1: HTAB 15: PRINT "QUIZ-PRINT": VTAB 4: HTAB 5: PRINT "MAKE A SELECTION:": POKE 33,32: POKE 32,8
340 VTAB 6: PRINT "1) SET UP": PRINT "2) PRINT": PRINT "3) SAVE": PRINT "4) LOAD": PRINT "5) EXIT": POKE 32,0: POKE 33,40
350 GOSUB 2100:KB = KB - 176: IF KB < 1 OR KB > 5 THEN 350
360 ON KB GOTO 400,1790,1480,1650,2530
370 REM
380 REM  SET UP INITIAL REPORT
390 REM
400 VTAB 2: CALL - 958: VTAB 3: HTAB 14: PRINT "SET UP QUIZ": PRINT : PRINT "ENTER QUIZ FILE NAME FROM QUIZ-MAKE:": PRINT ">":VT = 6:HT = 2: IF PD THEN ML = 16: GOSUB 2120: GOTO 420
410 ML = 30: GOSUB 2120
420 IF B$ = "" THEN 330
430 F$ = B$
440 IF PD THEN GOSUB 2540
450 PRINT D$;"VERIFY ";F$
460 PRINT D$;"OPEN ";F$
470 PRINT D$;"READ ";F$
480 INPUT TT$,DT$,CN$,QH$,AH$,NR,QT,LC,TL
490 PRINT CHR$ (1);"READING ";NR;" RECORDS...": FOR I = 1 TO NR: INPUT QZ$(I,1),QZ$(I,2): NEXT
500 PRINT D$;"CLOSE ";F$:FL = 1:QP = 0: GOSUB 2090
510 HOME : VTAB 8: PRINT "MULTIPLE CHOICE (Y/N)?";: GOSUB 2100:KB = KB - 128:K$ = CHR$ (KB): IF K$ < > "Y" AND K$ < > "N" THEN 510
520 PRINT K$: IF K$ = "N" THEN MC = 0:NC = 0:MA = 0:RA = 0: GOTO 600
530 MC = 1: VTAB 10: HTAB 1: PRINT "HOW MANY CHOICES (2 TO 10)";:VT = 10:HT = 28:ML = 2: GOSUB 2120: IF B$ = "" THEN 530
540 NC = VAL (B$): IF (NC < 2 OR NC > 10) OR (NC > (NR - 2)) THEN 530
550 PRINT : PRINT "SELECT ONE:": POKE 33,35: POKE 32,5: VTAB 13: PRINT "1) USE RANDOM ANSWERS FROM QUIZ": PRINT "2) ENTER YOUR OWN ANSWERS"
560 GOSUB 2100:KB = KB - 176: IF KB < 1 OR KB > 2 THEN 560
570 MA = KB - 1: POKE 32,0: POKE 33,40: PRINT : PRINT "SELECT ONE:": POKE 33,35: POKE 32,5: VTAB 17: PRINT "1) RANDOM ANSWER POSITION": PRINT "2) SELECT ANSWERS POSITION"
580 GOSUB 2100:KB = KB - 176: IF KB < 1 OR KB > 2 THEN 580
590 RA = KB - 1: POKE 32,0: POKE 33,40
600 HOME : VTAB 3: PRINT "SELECT ONE:": POKE 33,37: POKE 32,3: VTAB 4: PRINT "1) QUESTIONS IN SAME ORDER AS FILE": PRINT "2) RANDOM QUESTION ORDER": PRINT "3) PICK AND CHOOSE QUESTION ORDER"
610 GOSUB 2100:KB = KB - 176: IF KB < 1 OR KB > 3 THEN 610
620 POKE 32,0: POKE 33,40:QO = KB - 1: VTAB 8: PRINT "PRINT ANSWERS AT END OF REPORT (Y/N)?";
630 GOSUB 2100:KB = KB - 128:K$ = CHR$ (KB): IF K$ < > "Y" AND K$ < > "N" THEN 630
640 PRINT K$: IF K$ = "Y" THEN PA = 1: GOTO 660
650 PA = 0
660 IF QO < > 2 THEN 690
670 IF QP > = NR THEN 330
680 QP = QP + 1: GOSUB 930: GOTO 760
690 IF QO < > 1 THEN 730
700 IF QP > = NR THEN 330
710 HOME : PRINT : PRINT " RECORD ";QP + 1;" BEING SELECTED";: FOR TD = 1 TO 500: NEXT
720 QP = QP + 1: GOSUB 1170: GOTO 760
730 IF QP > = NR THEN 330
740 HOME : PRINT : PRINT " RECORD ";QP + 1;" BEING SELECTED";: FOR TD = 1 TO 500: NEXT
750 QU = QU + 1:QP = QP + 1
760 QR%(QP,1) = QU:QR%(QP,2) = 2: IF MC = 0 THEN 840
770 IF MA = 0 THEN GOSUB 1240: GOTO 790
780 GOSUB 1300
790 IF RA = 0 THEN GOSUB 1360: GOTO 810
800 GOSUB 1420
810 HOME : PRINT "IS THIS OK (Y/N)?": PRINT : PRINT QZ$(QR%(QP,1),1): PRINT QZ$(QR%(QP,1),2): PRINT : FOR I = 1 TO NC: PRINT CHR$ (I + 64);") "; LEFT$ (QZ$(QR%(QP,1),2 + I),36): NEXT : VTAB 1: HTAB 18
820 GOSUB 2100:KB = KB - 128:K$ = CHR$ (KB): IF K$ < > "Y" AND K$ < > "N" THEN 820
830 PRINT K$: IF K$ = "N" THEN 760
840 IF QO = 2 OR MC THEN HOME
850 VTAB 2: HTAB 12: CALL - 868: PRINT : PRINT "ADD ANOTHER QUESTION? (Y/N)";
860 GOSUB 2100:KB = KB - 128:K$ = CHR$ (KB): IF K$ < > "Y" AND K$ < > "N" THEN 860
870 IF K$ = "Y" THEN IF QP < NR THEN 660
880 IF K$ = "N" THEN 330
890 PRINT : PRINT "ALL QUESTIONS USED"; CHR$ (7): FOR T = 1 TO 2000: NEXT : GOTO 330
900 REM
910 REM  PICK & CHOOSE QUESTIONS
920 REM
930 TS = 0:SQ = 0
940 HOME : PRINT "SELECT A QUESTION BY ENTERING ITS NUMBER": PRINT "PRESS ";: INVERSE : PRINT "<-";: NORMAL : PRINT " OR ";: INVERSE : PRINT "->";: NORMAL : PRINT " TO SCROLL": PRINT
950 IF QP > NR THEN RETURN
960 IF TS < 0 THEN TS = 0
970 FOR SQ = 1 TO 5:TS = TS + 1:NO = 0
980 IF TS > NR THEN NO = 1: GOTO 1010
990 FOR I = 1 TO QP - 1: IF TS = QR%(I,1) THEN NO = 1
1000 NEXT : IF NO = 1 THEN TS = TS + 1
1010 IF NO = 0 THEN 1040
1020 IF TS > NR THEN 1050
1030 NO = 0: GOTO 980
1040 PRINT TS;":": PRINT QZ$(TS,1)
1050 NEXT SQ
1060 VTAB 23: PRINT "YOUR CHOICE: ";:VT = 23:HT = 14:ML = 3: GOSUB 2120
1070 IF KB = 136 THEN TS = TS - 10: GOTO 940
1080 IF KB = 149 THEN IF TS > = NR THEN TS = NR - 5: GOTO 940
1090 IF KB = 149 THEN 940
1100 Q = VAL (B$):I = 1
1110 IF Q = QR%(I,1) THEN PRINT CHR$ (7):TS = TS - 5: GOTO 940
1120 IF I < > QP THEN I = I + 1: GOTO 1110
1130 QU = VAL (B$): RETURN
1140 REM
1150 REM  SELECT AT RANDOM
1160 REM
1170 Q = INT (NR * RND (1)) + 1:Z = 0
1180 Z = Z + 1: IF Q = QR%(Z,1) THEN 1170
```

Continued

QUIZ-PRINT *Continued*



Continued

QUIZ-PRINT *Continued*

APPLE // Family

```
2240 IF KB - 128 = 1 OR KB - 128 = 17 THEN KB$ = CHR$ (KB - 128): RETURN
2250 IF KB - 128 < 32 THEN 2140
2260 VTAB VT: HTAB HT: PRINT CHR$ (KB - 128);:HT = HT + 1:B$ = B$ + CHR$ (KB - 128): IF HT = 41 THEN VT = VT + 1:HT = 1
2270 GOTO 2320
2280 IF GC = 0 THEN 2140
2290 IF GC = 1 THEN B$ = "":GC = GC - 1: VTAB VT: HTAB HT: PRINT " ":HT = HT - 1: GOTO 2310
2300 GC = GC - 1:B$ = LEFT$ (B$,( LEN (B$) - 1)): VTAB VT: HTAB HT: PRINT " ":HT = HT - 1: IF HT = 0 THEN HT = 40:VT = VT - 1
2310 GOTO 2140
2320 NEXT GC
2330 RETURN
2340 REM
2350 REM ERROR TRAPPING ROUTINE
2360 IF PEEK (222) > 21 OR PEEK (222) < 1 THEN 2470
2370 PRINT EM$(PD, PEEK (222)); CHR$ (7)
2380 ONERR GOTO 2480
2390 IF PEEK (222) < > 5 THEN 2420
2400 PRINT D$;"VERIFY ";F$
2410 PRINT D$;"DELETE ";F$
2420 IF PEEK (222) < > 6 THEN GOTO 2450
2430 PRINT D$;"CLOSE ";F$
2440 PRINT D$;"CLOSE ";FR$
2450 ONERR GOTO 2360
2460 FOR T = 1 TO 1500: NEXT : GOTO 330
2470 PRINT : PRINT "BREAK IN LINE "; PEEK (218) + PEEK (219) * 256: END
2480 PRINT D$;"DELETE ";FR$
2490 GOTO 2410
2500 REM
2510 REM EXIT PROGRAM
2520 REM
2530 HOME : PRINT "ALL DONE.": END
2540 PRINT : PRINT "PLACE DISK IN DRIVE 1 AND PRESS RETURN"
2550 GOSUB 2100: IF KB < > (141) THEN 2550
2560 PRINT D$;"PREFIX,D1"
2570 RETURN
```

HCM

QUIZ-PRINT

COMMODORE 64

```
100 REM ***************
110 REM * QUIZ-PRINT *
120 REM ***************
130 REM COPYRIGHT 1984,1985
140 REM EMERALD VALLEY PUBLISHING CO.
150 REM BY WILLIAM K. BALTHROP
160 REM AND THE HCM STAFF
170 REM HOME COMPUTER MAGAZINE
180 REM VERSION 5.1.1
190 REM C-64 BASIC
200 REM INITIALIZE AND DISPLAY TITLE
210 PRINT"{SHIFT CLR}"
220 MX=40:DIM QZ$(MX,12),QR(MX,2):POKE 650,128
230 POKE 53280,6:POKE 53281,12:POKE 646,0:POKE 53272,21:POKE 657,128
240 X=11:Y=12:GOSUB 2720:PRINT"** QUIZ-PRINT **"
250 X=8:Y=24:GOSUB 2720:PRINT"{CTRL RVS ON} PRESS RETURN TO START {CTRL RVS OFF}";
260 GOSUB 2870
270 IF ASC(K$)<>13THEN 260
280 REM MAIN MENU
290 POKE 198,0
300 PRINT"{SHIFT CLR}":PRINTTAB(15)"QUIZ-PRINT"
310 X=0:Y=6:GOSUB 2720
320 PRINT"     1) SETUP":PRINT:PRINT"     2) PRINT":PRINT
330 PRINT"     3) SAVE":PRINT:PRINT"     4) LOAD":PRINT:PRINT"     5) EXIT"
340 X=4:Y=24:GOSUB 2720:PRINT"{CTRL RVS ON} INPUT A NUMBER BETWEEN 1 AND 5 {CTRL RVS OFF}";
350 GOSUB 2870
360 IF ASC(K$)<49 OR ASC(K$)>53 THEN 350
370 ON VAL(K$)GOSUB 390,2260,1790,2030,2970
380 GOTO 280
390 REM READ IN QUIZ FROM QUIZ-MAKE
400 PRINT"{SHIFT CLR}":X=15:Y=1:GOSUB 2720
410 PRINT"QUIZ-PRINT":X=14:Y=3:GOSUB 2720::PRINT"SET UP QUIZ"
420 X=0:Y=5:GOSUB 2720:PRINT"INPUT QUIZ-MAKE FILE NAME:";
430 LL=14:KL=65:KU=90:GOSUB 2740:X$=ST$:IF X$="" THEN RETURN
440 PRINT:PRINT"TAPE OR DISK(T/D)? ";
450 GOSUB 2870::IF K$<>"T" AND K$<>"D" THEN 450
460 DV$="{CTRL RVS ON}TAPE{CTRL RVS OFF}":IF K$="D"THEN DV$="{CTRL RVS ON}DISK{CTRL RVS OFF}"
470 PRINTDV$:PRINT
480 IFDV$="{CTRL RVS ON}TAPE{CTRL RVS OFF}"THEN 500
490 OPEN 1,8,8,"0:"+X$+".Q"+",S,R":PRINT"READING RECORD:":GOSUB 510:GOTO 690
500 OPEN 1,1,0,X$+".Q":PRINT"READING RECORD:":GOSUB 510:GOTO 690
510 INPUT#1,TL$
520 INPUT#1,DT$
530 INPUT#1,CN$
540 INPUT#1,QT$
550 INPUT#1,AT$
560 INPUT#1,NR
570 INPUT#1,QT
580 INPUT#1,LCP
590 INPUT#1,TD:IF TL$="" THEN VIN=0:GOTO 640
600 IF NR=0 THEN 640
610 FOR I=1 TO NR
620 INPUT#1,QZ$(I,1)
630 INPUT#1,QZ$(I,2):X=15:Y=9:GOSUB 2720::PRINTI:NEXT
640 IF DV$="{CTRL RVS ON}TAPE{CTRL RVS OFF}"THEN 680
650 OPEN 15,8,15:INPUT#15,V,S$:CLOSE15
660 IFV<>0 THEN GOTO 680
670 CLOSE1:RETURN
680 PRINT:PRINTS$".PLEASE WAIT":FOR TM=1TO2500:NEXTTM:CLOSE1:GOTO 280
690 X=0:Y=11:GOSUB 2720:PRINT"PRINT MULTIPLE CHOICE (Y/N)?";:GOSUB 2870
700 IF K$<>"N" AND K$<>"Y" THEN 690
710 PRINT" "K$:IF K$="N"THEN MC=0:NC=0:MA=0:AP=0:QO=0:GOTO 890
720 MC=1
730 X=0:Y=13:GOSUB 2720::PRINT"HOW MANY CHOICES (2 TO 10) ? ";
740 NC$="":GOSUB 2870
750 IFK$="1"THEN NC$=NC$+K$:PRINTK$;:GOSUB2870:PRINT K$:NC$=NC$+K$:NC=VAL(NC$)
760 NC=VAL(NC$)
770 IF VAL(K$)>1 AND VAL(K$)<=9 THEN PRINTK$:NC=VAL(K$)
780 IF NC<2 OR NC>10 THEN 730
790 X=0:Y=15:GOSUB 2720:PRINT"SELECT ONE:":X=3:Y=16:GOSUB 2720
800 PRINT"1) USE RANDOM ANSWERS FROM THE QUIZ":X=3:Y=17:GOSUB 2720
810 PRINT"2) ENTER YOUR OWN ANSWERS":X=12:Y=15:GOSUB 2720
820 GOSUB 2870:IF K$<"1" OR K$>"2" THEN 820
830 PRINT K$:MA=VAL(K$)-1
840 X=0:Y=19:GOSUB 2720:PRINT"SELECT ONE:":X=3:Y=20:GOSUB 2720
850 PRINT"1) RANDOM ANSWER POSITION":X=3:Y=21:GOSUB 2720
860 PRINT"2) SELECT ANSWERS POSITION":X=12:Y=19:GOSUB 2720
870 GOSUB 2870:IF K$<"1" OR K$>"2"THEN 870
880 PRINT K$:AP=VAL(K$)-1
890 PRINT"{SHIFT CLR}":X=0:Y=3:GOSUB 2720:PRINT"SELECT ONE:":X=3:Y=4:GOSUB 2720
900 PRINT"1) QUESTIONS IN SAME ORDER AS FILE":X=3:Y=5:GOSUB 2720
910 PRINT"2) RANDOM QUESTION ORDER":X=3:Y=6:GOSUB 2720
920 PRINT"3) PICK AND CHOOSE QUESTION ORDER":X=12:Y=3:GOSUB 2720
930 GOSUB 2870::IF K$<"1" OR K$>"3" THEN 930
940 QO=VAL(K$)-1:PRINT K$
950 X=0:Y=8:GOSUB 2720:PRINT"PRINT ANSWERS END OF REPORT (Y/N)?";
960 X=35:Y=8:GOSUB 2720:GOSUB 2870:IF K$<>"Y" AND K$<>"N"THEN 960
970 IF K$="N"THENPA=0:PRINT K$
980 IF K$="Y"THENPA=1:PRINT K$
990 PRINT"{SHIFT CLR}":QU=0:QP=0
1000 IF QO=2 THEN IF QP>NR THEN VIN=1:GOTO 280
1010 IF QO=2 AND QP<NR THEN QP=QP+1:GOSUB 1220:GOTO 1060
1020 IF QO=1 THEN IF QP>NR THEN VIN=1:GOTO 280
1030 IF QO=1 THEN IF QP<NR THEN QP=QP+1:GOSUB 1520:GOTO 1060
1040 IF QP>=NR THEN 280
1050 QU=QU+1:QP=QP+1
1060 QR(QP,1)=QU:QR(QP,2)=2:IF MC=0 THEN GOSUB 2990:VIN=1:GOTO 1160
1070 IF MA=0 THEN GOSUB 1570
1080 IF MA<>0 THEN GOSUB 1620
1090 IF AP=0 THEN GOSUB 1690
1100 IF AP<>0 THEN GOSUB 1740
1110 PRINT"{SHIFT CLR}":X=0:Y=1:GOSUB 2720:PRINT"IS THIS OK (Y/N)?{CRSR DOWN}"
```

Continued

QUIZ-PRINT *Continued*

```
1120 PRINT"QUIZ-PRINT":PRINT:PRINT"DO YOU WANT TO (C)REATE A QUIZ, (P)RINT A QUIZ,"
1130 INPUT"OR (E)DIT A QUIZ";Q$:IF Q$="C" THEN 1170
1140 IF Q$="P" THEN 1500
1150 IF Q$="E" THEN 1760
1160 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
1170 INPUT"DO YOU WANT TO SAVE (S) OR LOAD (L) A QUIZ";Q$:IF Q$="L" THEN 1300
1180 INPUT"DO YOU WANT TO (S)AVE OR (L)OAD A QUIZ";Q$:IF Q$="L" THEN 1300
1190 INPUT"DO YOU WANT TO (S)AVE OR (L)OAD A QUIZ";Q$:IF Q$="L" THEN 1300
1200 INPUT"DO YOU WANT TO (S)AVE OR (L)OAD A QUIZ";Q$:IF Q$="L" THEN 1300
1210 INPUT"DO YOU WANT TO (S)AVE OR (L)OAD A QUIZ";Q$:IF Q$="L" THEN 1300
1220 INPUT"DO YOU WANT TO (S)AVE OR (L)OAD A QUIZ";Q$:IF Q$="L" THEN 1300
1230 INPUT"DO YOU WANT TO (S)AVE OR (L)OAD A QUIZ";Q$:IF Q$="L" THEN 1300
1240 INPUT"DO YOU WANT TO (S)AVE OR (L)OAD A QUIZ";Q$:IF Q$="L" THEN 1300
1250 OPEN15,8,15,"I":OPEN2,8,2,"0:QUIZ,S,W":PRINT#2,CHR$(NC);:CLOSE2:CLOSE15
1260 RETURN
1270 OPEN15,8,15,"I":OPEN2,8,2,"0:QUIZ,S,R":INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1280 INPUT#2,NC:PRINT"LOADING ";:FOR Z=1 TO NC:INPUT#2,QZ$(Z):PRINT".";:NEXT:CLOSE2:CLOSE15:RETURN
1290 IF Q$="S" THEN 1330
1300 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1310 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1320 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1330 INPUT"DO YOU WANT TO (E)DIT OR (A)DD QUESTIONS";Q$:IF Q$="A" THEN NC=NC+1:Z=NC:GOTO 1360
1340 INPUT"ENTER QUESTION NUMBER";Z:IF Z<1 OR Z>NC THEN PRINT"OUT OF RANGE":GOTO 1340
1350 PRINT"CURRENT QUESTION:":PRINT QZ$(Z):INPUT"ENTER NEW QUESTION";QZ$(Z):RETURN
1360 INPUT"ENTER QUESTION";QZ$(Z):RETURN
1370 INPUT"ENTER QUESTION NUMBER TO DELETE";Z:IF Z<1 OR Z>NC THEN PRINT"OUT OF RANGE":GOTO 1370
1380 FOR Z=Z TO NC-1:QZ$(Z)=QZ$(Z+1):NEXT:NC=NC-1:RETURN
1390 IF NC<1 THEN PRINT"NO QUESTIONS":GOTO 1420
1400 FOR Z=1 TO NC:PRINT QZ$(Z):NEXT
1410 INPUT"ENTER QUESTION NUMBER";Z:IF Z<1 OR Z>NC THEN PRINT"OUT OF RANGE":GOTO 1410
1420 RETURN
1430 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1440 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1450 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1460 IF Q$="E" THEN 1490
1470 IF Q$="A" THEN 1330
1480 IF Q$="D" THEN 1390
1490 PRINT"PLEASE ANSWER E, A, OR D.":GOTO 1420
1500 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1510 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15
1520 INPUT"HOW MANY QUESTIONS";QU:IF QU>NC THEN QU=NC
1530 FOR Q=1 TO QU:Z=INT(RND(1)*NC+1):PRINT QZ$(Z):NEXT:RETURN
1540 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1550 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1560 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1570 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
1580 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1590 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1600 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1610 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
1620 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1630 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1640 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1650 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1660 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1670 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1680 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
1690 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1700 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1710 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1720 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1730 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1740 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1750 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
1760 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1770 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1780 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1790 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
1800 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1810 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1820 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1830 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
1840 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1850 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1860 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1870 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
1880 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1890 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1900 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1910 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
1920 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1930 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1940 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1950 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
1960 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
1970 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1980 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
1990 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
2000 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
2010 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2020 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2030 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
2040 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
2050 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2060 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2070 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
2080 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
2090 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2100 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2110 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
2120 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
2130 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2140 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2150 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
2160 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
2170 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2180 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2190 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
2200 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
2210 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2220 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2230 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
2240 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
2250 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2260 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2270 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
2280 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
2290 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2300 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2310 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
2320 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
2330 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2340 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2350 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
2360 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
2370 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2380 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2390 PRINT"PLEASE ANSWER C, P, OR E.":GOTO 1130
2400 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,R":OPEN15,8,15:OPEN2,8,2,Q$:INPUT#15,EN,EM$:IF EN>20 THEN PRINT"DISK ERROR":PRINT EN;EM$:STOP
2410 INPUT#2,NC:FOR Z=1 TO NC:INPUT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
2420 INPUT"ENTER QUIZ NAME";Q$:Q$="0:"+Q$+",S,W":OPEN15,8,15:OPEN2,8,2,Q$:PRINT#2,CHR$(NC);:FOR Z=1 TO NC:PRINT#2,QZ$(Z):NEXT:CLOSE2:CLOSE15:RETURN
```

Continued

PROGRAM LISTING

QUIZ-PRINT *Continued*

COMMODORE 64

```
2430 FOR B=1 TO 79:PRINT#4,CHR$(61);:NEXT B:PRINT#4,""
2440 NP=INT(60/(7+NC)):FOR Z=1 TO QP-1 STEP NP:NL=0:PRINT#4,"PAGE ";INT(Z/NP)+1
2450 NL=1:IF Z=1 THEN NL=5
2460 FOR B=1TO79:PRINT#4,CHR$(61);:NEXT B:PRINT#4,"":NL=NL+1
2470 FOR X=0 TO NP-1:IF Z+X>QP THEN PRINT#4,CHR$(13):NL=NL+2:GOTO 2540
2480 PRINT#4,STR$(Z+X);". ";QZ$(QR(Z+X,1),1):PRINT#4,CHR$(13):NL=NL+3
2490 FOR B=1 TO (40*ABS(MC-1)+5):PRINT#4,CHR$(45);:NEXT:PRINT#4,"":NL=NL+1
2500 IFMC=1THENFORY=1TONC:PRINT#4,CHR$(64+Y)") "QZ$(QR(Z+X,1),Y+2):NL=NL+1:NEXT
2510 FORB=1TO79:PRINT#4,CHR$(45);:NEXT B:PRINT#4,CHR$(13):NL=NL+2:NEXT X
2520 PRINT#4,"":NL=NL+1:FOR ZZ=1 TO 66-NL:PRINT#4,""
2530 NEXTZZ:NEXTZ
2540 X=O:Y=7:GOSUB 2720::PRINT"QUIZ REPORT IS COMPLETE"
2550 IF PA<>0 THEN 2600
2560 PRINT#4,"":CLOSE4
2570 X=7:Y=24:GOSUB 2720::PRINT"{CTRL RVSON} PRESS RETURN TO CONTINUE {CTRL RVSOFF}";
2580 GOSUB 2870::IF K$<>CHR$(13)THEN 2580
2590 RETURN
2600 REM PRINT ANSWER SHEET
2610 X=0:Y=9:GOSUB2720:PRINT"PRINTING ANSWER SHEET":PRINT#4,"ANSWER SHEET FOR:"
2620 PRINT#4,T$:PRINT#4,C$:PRINT#4,D$:IF MC=0 THEN 2700
2630 SP$="      ":FOR Z=1TO QP STEP 5:FOR Y=0TO4:IF Z+Y>QP THEN 2660
2640 PRINT#4,RIGHT$(STR$(Z+Y),2);") ";CHR$(QR(Z+Y,2)+62);SP$;:NEXT Y
2650 PRINT#4,CHR$(13):NEXT Z
2660 PRINT#4,"":CLOSE4:X=0:Y=11:GOSUB 2720::PRINT"ANSWER SHEET COMPLETE"
2670 X=10:Y=24:GOSUB 2720:PRINT"{CTRL RVSON} PRESS RETURN TO CONTINUE {CTRL RVSOFF}";
2680 GOSUB 2870:IF K$<>CHR$(13)THEN 2680
2690 RETURN
2700 FOR Z=1 TO QP:PRINT#4, STR$(Z);") ";QZ$(QR(Z,1),2):NEXT Z:GOTO 2660
2710 STOP
2720 REM ROUTINE TO SIMULATE THE PRINT AT STATEMENT
2730 POKE 781,Y:POKE 782,X:POKE 783,0:SYS 65520:RETURN
2740 REM INPUT ROUTINE
2750 POKE 198,0:L=0:ST$=""
2760 PRINT"{CMDR @}{SHIFT CRSR LEFT}";:GOSUB 2870
2770 IF ASC(K$)=13 THEN PRINT" ":GOTO 2860
2780 IFASC(K$)=216 AND LL=39 THEN 2860
2790 IF ASC(K$)<>20 OR L=0 THEN 2810
2800 ST$=LEFT$(ST$,LEN(ST$)-1):PRINT"{2 SHIFT CRSR LEFT}";:L=L-1:GOTO 2760
2810 IF ASC(K$)<KL OR ASC(K$)>KU THEN 2760
2820 IF ASC(K$)=34 THEN 2760
2830 PRINT K$;:ST$=ST$+K$:L=L+1
2840 IF L=LL THEN 2860
2850 GOTO 2760
2860 RETURN
2870 REM SINGLE KEY INPUT
2880 GET K$:IF K$="" THEN 2880
2890 POKE 198,0:RETURN
2900 REM ILLEGAL ENTRY MESSAGE
2910 PRINT"{SHIFT CLR}"
2920 X=10:Y=9:GOSUB 2720::PRINT"YOU NEED TO SETUP":PRINT
2930 PRINT TAB(7)"A QUIZ FILE BEFORE USING":PRINT
2940 PRINT TAB(13)"THIS OPTION"
2950 FOR TM=1 TO 3000:NEXT:RETURN
2960 REM EXIT ROUTINE
2970 PRINT"{SHIFT CLR}":PRINT"BYE BYE, SEE YOU NEXT TIME..."
2980 POKE53280,14:POKE53281,6:POKE646,14:POKE650,0:POKE657,0:END
2990 X=2:Y=18:GOSUB 2720:PRINT"RECORD #";QP:RETURN
```

HCM

QUIZ-PRINT

IBM PC & IBM PCjr

```
100 '  **************
110 '  * QUIZ-PRINT *
120 '  **************
130 '  COPYRIGHT 1984,1985
140 '  EMERALD VALLEY PUBLISHING CO.
150 '  BY WILLIAM K. BALTHROP
160 '  HOME COMPUTER MAGAZINE
170 '  VERSION 5.1.1
180 '  IBM PCjr WITH CARTRIDGE BASIC
190 '  IBM PC WITH BASICA FROM DOS 2.10
200 '
210 '  INITIALIZE PROGRAM--DISPLAY TITLE SCREEN
220 '
230 ON ERROR GOTO 1460
240 OPTION BASE 1:SCREEN 0:DEFINT A-Z:RANDOMIZE TIMER:KEY OFF:MXREC=600:DIM QZ$(MXREC,12),QR(MXREC,2),ERM$(14),ERCD(14):RESTORE 1490:FOR Z=1 TO 14:READ ERCD(Z),ERM$(Z):NEXT
250 CLS:LOCATE 12,15:PRINT "QUIZ-PRINT":GOSUB 1400
260 '
270 '  MAIN MENU SELECTION SCREEN
280 '
290 CLS:LOCATE 1,15:PRINT "QUIZ-PRINT":LOCATE 4,5:PRINT "MAKE A SELECTION:":LOCATE 6,8:PRINT "1) SET UP":LOCATE 8,8:PRINT "2) PRINT":LOCATE 10,8:PRINT "3) SAVE":LOCATE 12,8:PRINT "4) LOAD":LOCATE 14,8:PRINT "5) EXIT"
300 GOSUB 1410:IF K$<"1" OR K$>"5" THEN 300 ELSE ON VAL(K$) GOTO 340,1200,1100,1150,1540
310 '
320 '  SET UP INITIAL QUIZ REPORT
330 '
340 CLS:LOCATE 1,15:PRINT "QUIZ-PRINT":LOCATE 3,14:PRINT "SET UP QUIZ":FOR Z=1 TO MXREC:QR(Z,1)=0:QR(Z,2)=0:NEXT:QP=0
350 LOCATE 5,1:PRINT "ENTER QUIZ FILE NAME FROM QUIZ-MAKE:":INPUT F$:F$=F$+".QZ":OPEN F$ FOR INPUT AS #1
360 PRINT "READING RECORD: ":INPUT #1,TTL$,DT$,NM$,QT$,AT$,NR,QT,LCP,TLIM:FOR Z=1 TO NR:INPUT #1,QZ$(Z,1),QZ$(Z,2):LOCATE 7,16:PRINT Z;:NEXT:CLOSE #1:GOSUB 1400
370 CLS:LOCATE 8,1:PRINT "PRINT MULTIPLE CHOICE (Y/N)?";:GOSUB 1410:IF (K$<>"Y") AND (K$<>"y") AND (K$<>"N") AND (K$<>"n") THEN GOTO 370 ELSE PRINT K$:IF K$="N" OR K$="n" THEN MC=0:NC=0:MCA=0:RAP=0:GOTO 450 ELSE MC=1
380 LOCATE 10,1:INPUT "HOW MANY CHOICES (2 TO 10)";NC:IF NC<2 OR NC>10 THEN 380
390 LOCATE 12,1:PRINT "SELECT ONE:":LOCATE 13,5:PRINT "1) USE RANDOM ANSWERS FROM THE QUIZ":LOCATE 14,5:PRINT "2) ENTER YOUR OWN ANSWERS":LOCATE 12,12,1
400 GOSUB 1410:IF K$<"1" OR K$>"2" THEN 400 ELSE MCA=VAL(K$)-1:PRINT K$
410 LOCATE 16,1:PRINT "SELECT ONE:":LOCATE 17,5:PRINT "1) RANDOM ANSWER POSITION":LOCATE 18,5:PRINT "2) SELECT ANSWERS POSITION"
420 LOCATE 16,12,1
430 GOSUB 1410:IF K$<"1" OR K$>"2" THEN 430 ELSE RAP=VAL(K$)-1
440 PRINT K$
450 CLS:LOCATE 3,1:PRINT "SELECT ONE:":LOCATE 4,5:PRINT "1) SAME ORDER AS FILE":LOCATE 5,5:PRINT "2) RANDOM QUESTION ORDER":LOCATE 6,5:PRINT "3) PICK AND CHOOSE QUESTION ORDER":LOCATE 3,12,1
460 GOSUB 1410:IF K$<"1" OR K$>"3" THEN 460 ELSE QO=VAL(K$)-1:PRINT K$
470 LOCATE 8,1,1:PRINT "PRINT ANSWERS AT END OF REPORT (Y/N)?";:LOCATE 8,39,1
480 GOSUB 1410:IF K$<>"Y" AND K$<>"y" AND K$<>"N" AND K$<>"n" THEN 480 ELSE IF K$="Y" OR K$="y" THEN PANS=1 ELSE PANS=0
490 PRINT K$
500 QUEST=0:QP=0
510 IF QO=2 THEN IF QP>NR THEN 290 ELSE QP=QP+1:GOSUB 650:GOTO 540
520 IF QO=1 THEN IF QP>NR THEN 290 ELSE QP=QP+1:GOSUB 820:GOTO 540
530 IF QP>=NR THEN 290 ELSE QUEST=QUEST+1:QP=QP+1
540 QR(QP,1)=QUEST:QR(QP,2)=2:IF MC=0 THEN 590
```

Continued

PROGRAM LISTING

QUIZ-PRINT *Continued*

```
10800 '
10900 '
11000 '
11100 '
11110 '
11120 '
11130 '
11140 '
11150 '
11160 '
11170 '
11180 '
11190 '
11200 '
11210 '
11220 '
11230 '
11240 '
11250 '
11260 '
11270 '
11280 '
11290 '
11300 '
11310 '
11320 '
11330 '
11340 '
11350 '
11360 '
11370 '
11380 '
11390 '
11400 '
11410 '
11420 '
11430 '
11440 '
11450 '
11460 '
11470 '
11480 '
11490 '
11500 '
11510 '
15200 '
15300 '
15400 '
```

```
550 '
560 '
570 '
580 '
590 '
600 '
610 '
620 '
630 '
640 '
650 '
660 '
670 '
680 '
690 '
700 '
710 '
720 '
730 '
740 '
750 '
760 '
770 '
780 '
790 '
800 '
810 '
820 '
830 '
840 '
850 '
860 '
870 '
880 '
890 '
900 '
910 '
920 '
930 '
940 '
950 '
960 '
970 '
980 '
990 '
1000 '
1010 '
1020 '
1030 '
1040 '
1050 '
1060 '
1070 '
```


HCM

QUIZ-PRINT


```
100 REM   ***************
110 REM   * QUIZ PRINT  *
120 REM   ***************
130 REM   COPYRIGHT 1985
140 REM   EMERALD VALLEY PUBLISHING CO.
150 REM   BY WILLIAM K. BALTHROP
160 REM   HOME COMPUTER MAGAZINE
170 REM   VERSION 5.1.1
180 REM   TI BASIC or
190 REM   TI EXTENDED BASIC
200 REM
210 RANDOMIZE
220 DIM QZ$(40,12),QR(40,2)
230 CALL CLEAR
240 PRINT TAB(9);"QUIZ-PRINT"::::"MAKE
A SELECTION:"::" 1) SET UP"::" 2) P
RINT"::" 3) SAVE"::" 4) LOAD"
250 PRINT :" 5) EXIT"::::::::
260 GOSUB 3240
270 IF (K<49)+(K>53)THEN 260
280 ON K-48 GOTO 320,2210,2880,3060,331
0
290 REM
300 REM   SET UP INITIAL REPORT
310 REM
320 CALL CLEAR
330 PRINT TAB(9);"QUIZ-PRINT":TAB(8);"S
ET UP QUIZ":::
340 FOR Z=1 TO 40
350 QR(Z,1)=0
360 QR(Z,2)=0
370 NEXT Z
380 PRINT "ENTER THE DEVICE NAME":"AND
FILE NAME FOR DISKS":"OR":"CS1 FOR
CASSETTE OPERATION"::
390 PRINT "PRESS ENTER TO RETURN":::
400 INPUT "ENTER QUIZ FILE NAME FROM
QUIZ MAKE:":F$
410 IF F$="" THEN 230
420 OPEN #1:F$,FIXED 128,INTERNAL
430 INPUT #1:TTL$,DT$,CN$,QT$
440 INPUT #1:AT$,NR,QT,LCP,TLIM
450 PRINT ::"READING";NR;"RECORDS...":: 
460 FOR Z=1 TO NR
470 INPUT #1:QZ$(Z,1),QZ$(Z,2)
480 NEXT Z
490 CLOSE #1
500 PRINT :::"PRESS ENTER TO CONTINUE"
510 GOSUB 3240
520 RESTORE 540
530 READ MC,NC,MCA,RAP
540 DATA 0,0,0,0
550 PRINT :::"PRINT MULTIPLE CHOICE (Y/
N)?";
560 MC=0
570 GOSUB 3240
580 IF (K<>89)*(K<>78)THEN 570
590 IF K=78 THEN 720
600 MC=1
610 PRINT ::"HOW MANY CHOICES (2 to 10)
?"
620 INPUT NC
630 IF (NC<2)+(NC>10)THEN 610
640 PRINT ::"SELECT ONE:":" 1) USE RAND
OM ANSWERS":" 2) ENTER YOUR OWN ANS
WERS"
650 GOSUB 3240
660 IF (K<49)+(K>50)THEN 650
670 MCA=K-49
680 PRINT ::"SELECT ONE:":" 1) RANDOM A
NSWER POSITION":" 2) SELECT ANSWER
POSITION"
690 GOSUB 3240
700 IF (K<49)+(K>50)THEN 690
710 RAP=K-49
720 PRINT ::"SELECT ONE:":" 1) SAME ORD
ER AS FILE":" 2) RANDOM ORDER":" 3)
 PICK AND CHOOSE"
730 GOSUB 3240
740 IF (K<49)+(K>51)THEN 730
750 QO=K-49
760 PRINT ::"PRINT ANSWERS WITH REPORT
 (Y/N)?";
770 GOSUB 3240
780 IF (K<>89)*(K<>78)THEN 770
790 PANS=SGN(K-78)
800 QUEST=0
810 QP=0
820 IF QO<>2 THEN 870
830 IF QP>NR THEN 230
840 QP=QP+1
850 GOSUB 1250
860 GOTO 950
870 IF QO<>1 THEN 920
880 IF QP>NR THEN 230
890 QP=QP+1
900 GOSUB 1750
910 GOTO 950
920 IF QP>=NR THEN 230
930 QUEST=QUEST+1
940 QP=QP+1
950 QR(QP,1)=QUEST
960 QR(QP,2)=2
970 IF MC=0 THEN 1160
980 IF MCA=0 THEN 1010
990 GOSUB 1900
1000 GOTO 1020
1010 GOSUB 1810
1020 IF RAP=0 THEN 1050
1030 GOSUB 2110
1040 GOTO 1060
1050 GOSUB 2000
1060 CALL CLEAR
1070 PRINT "IS THIS OK (Y/N)?"
1080 PRINT QZ$(QR(QP,1),1)
1090 PRINT QZ$(QR(QP,1),2)
1100 FOR Z=1 TO NC
1110 PRINT CHR$(Z+64);") ";QZ$(QR(QP,1),
2+Z)
1120 NEXT Z
1130 GOSUB 3240
1140 IF (K<>89)*(K<>78)THEN 1130
1150 IF K=78 THEN 950
1160 CALL CLEAR
1170 PRINT QP;"RECORDS IN QUIZ":::"ADD A
NOTHER QUESTION (Y/N)?";
1180 GOSUB 3240
1190 IF (K<>89)*(K<>78)THEN 1180
1200 IF K=78 THEN 230
1210 IF QP<NR THEN 820
1220 PRINT ::"ALL QUESTIONS ARE USED!"
1230 GOSUB 3280
1240 GOTO 230
1250 TOS=0
1260 CALL CLEAR
1270 Q$=""
1280 IF QP>NR THEN 1740
1290 PRINT "SELECT QUESTION BY ENTERING
ITS NUMBER"::"USE 'E' AND 'X' TO SC
ROLL":::
1300 FOR SQ=1 TO 5
1310 TOS=TOS+1
1320 IF TOS>NR THEN 1420
1330 FOR Z=1 TO QP
1340 IF TOS<>QR(Z,1)THEN 1390
1350 TOS=TOS+1
1360 IF (TOS>NR)*(SQ=1)THEN 1260
1370 IF TOS>NR THEN 1420
1380 GOTO 1330
1390 NEXT Z
1400 PRINT TOS:QZ$(TOS,1)
1410 NEXT SQ
1420 PRINT ::"YOUR CHOICE:"
1430 GOSUB 3240
1440 IF K<>69 THEN 1490
1450 TOS=TOS-10
1460 IF TOS>0 THEN 1480
1470 TOS=0
1480 GOTO 1260
1490 IF K<>88 THEN 1530
1500 IF TOS<NR THEN 1520
1510 TOS=NR-5
1520 GOTO 1260
1530 IF ((K<48)+(K>57))*(K<>13)THEN 1430
1540 IF K<>13 THEN 1570
1550 Q=VAL(Q$)
1560 GOTO 1610
1570 Q$=Q$&CHR$(K)
1580 PRINT CHR$(K);
1590 IF LEN(Q$)=2 THEN 1550
1600 GOTO 1430
1610 IF (Q<=NR)*(Q<>0)THEN 1670
1620 CALL SOUND(50,110,0)
1630 TOS=TOS-5
1640 IF TOS>=0 THEN 1260
1650 TOS=0
1660 GOTO 1260
1670 FOR Z=1 TO QP
1680 IF Q<>QR(Z,1)THEN 1720
1690 CALL SOUND(50,110,0)
1700 TOS=TOS-6
1710 GOTO 1260
1720 NEXT Z
1730 QUEST=Q
1740 RETURN
1750 Q=INT(RND*NR+1)
1760 FOR Z=1 TO QP
1770 IF Q=QR(Z,1)THEN 1750
1780 NEXT Z
1790 QUEST=Q
1800 RETURN
1810 FOR Z=1 TO NC-1
1820 A$=QZ$(INT(RND*NR)+1,2)
1830 FOR Y=2 TO Z+1
1840 IF A$=QZ$(QR(QP,1),Y)THEN 1820
1850 NEXT Y
1860 QZ$(QR(QP,1),Z+2)=A$
1870 NEXT Z
1880 QZ$(QR(QP,1),NC+2)=""
1890 RETURN
1900 CALL CLEAR
1910 PRINT "ENTER WRONG ANSWERS FOR:":QZ
$(QR(QP,1),1):QZ$(QR(QP,1),2)
1920 FOR Z=1 TO NC-1
1930 PRINT "ANSWER #";STR$(Z);
1940 INPUT "":QZ$(QR(QP,1),2+Z)
1950 QZ$(QR(QP,1),2+Z)=SEG$(QZ$(QR(QP,1)
,2+Z),1,28)
1960 IF QZ$(QR(QP,1),2+Z)="" THEN 1930
1970 NEXT Z
1980 QZ$(QR(QP,1),NC+2)=""
1990 RETURN
2000 P=INT(RND*NC+3)
2010 IF P<NC+2 THEN 2050
2020 QZ$(QR(QP,1),P)=QZ$(QR(QP,1),2)
```

Continued

QUIZ-PRINT *Continued*

TI-99/4A

```
2030 QR(QP,2)=P
2040 RETURN
2050 FOR Z=NC+2 TO P+1 STEP -1
2060 QZ$(QR(QP,1),Z)=QZ$(QR(QP,1),Z-1)
2070 NEXT Z
2080 QZ$(QR(QP,1),P)=QZ$(QR(QP,1),2)
2090 QR(QP,2)=P
2100 RETURN
2110 CALL CLEAR
2120 PRINT QZ$(QR(QP,1),1):QZ$(QR(QP,1),
     2)
2130 FOR Z=3 TO NC+2
2140 PRINT CHR$(62+Z);")  ";QZ$(QR(QP,1),
     Z)
2150 NEXT Z
2160 PRINT ::"WHAT POSITION A TO ";CHR$(
     64+NC)
2170 INPUT P$
2180 IF (P$<"A")+(P$>CHR$(64+NC))THEN 21
     70
2190 P=ASC(P$)-62
2200 GOTO 2010
2210 CALL CLEAR
2220 INPUT "REPORT TITLE:":T$
2230 INPUT "CLASS OR SECOND HEADER:":C$
2240 INPUT "DATE:":D$
2250 PRINT "ENTER PRINTER PARAMETERS:"
2260 INPUT P$
2270 IF P$="" THEN 230
2280 OPEN #1:P$
2290 PRINT :::"REPORT BEING GENERATED...
     "
2300 PRINT #1:TAB((80-LEN(T$))/2);T$:TAB
     ((80-LEN(C$))/2);C$:TAB((80-LEN(D$)
     )/2);D$
2310 GOSUB 2800
2320 NPP=INT(60/(5+NC+(SGN(NC))))
2330 FOR R=1 TO QP STEP NPP
2340 PRINT #1:"PAGE";INT(R/NPP+1)
2350 GOSUB 2800
2360 FOR X=0 TO NPP-1
2370 IF R+X>QP THEN 2550
2380 PRINT #1:STR$(R+X);".  ";QZ$(QR(R+X,
     1),1)
2390 IF MC=1 THEN 2450
2400 PRINT #1:""
2410 ZZ=40
2420 GOSUB 2840
2430 PRINT #1:""
2440 GOTO 2490
2450 PRINT #1:"":"_____":""
2460 FOR Y=1 TO NC
2470 PRINT #1:CHR$(64+Y);")  ";QZ$(QR(R+X
     ,1),Y+2)
2480 NEXT Y
2490 ZZ=80
2500 GOSUB 2840
2510 NEXT X
2520 IF R+X=QP THEN 2550
2530 PRINT #1:CHR$(12);
2540 NEXT R
2550 PRINT "ALL DONE WITH REPORT":::
2560 IF PANS=1 THEN 2580
2570 GOTO 230
2580 PRINT "PRINTING ANSWER SHEET...":::
2590 PRINT #1:CHR$(12);"ANSWER SHEET FOR
     :":T$:C$:D$::
2600 IF MC=0 THEN 2690
2610 FOR R=1 TO QP STEP 5
2620 FOR Y=0 TO 4
2630 IF R+Y>QP THEN 2740
2640 PRINT #1:TAB(Y*15+1);STR$(R+Y);")  "
     ;CHR$(QR(R+Y,2)+62);
2650 NEXT Y
2660 PRINT #1:""
2670 NEXT R
2680 GOTO 2740
2690 FOR R=1 TO QP
2700 PRINT #1:STR$(R);")  ";QZ$(QR(R,1),2
     )
2710 IF INT(R/50)<>R/50 THEN 2730
2720 PRINT #1:CHR$(12);
2730 NEXT R
2740 PRINT #1:CHR$(12);
2750 PRINT ::"ANSWER SHEET DONE":::
2760 PRINT "PRESS ENTER TO CONTINUE";
2770 GOSUB 3240
2780 CLOSE #1
2790 GOTO 230
2800 FOR Z=1 TO 80
2810 PRINT #1:"=";
2820 NEXT Z
2830 RETURN
2840 FOR Z=1 TO ZZ
2850 PRINT #1:"_";
2860 NEXT Z
2870 RETURN
2880 CALL CLEAR
2890 PRINT "SAVE QUIZ REPORT"::::
2900 PRINT "ENTER THE DEVICE NAME":"AND
     FILE NAME FOR DISKS":"OR":"CS1 FOR
     CASSETTE OPERATION"::
2910 PRINT "PRESS ENTER TO RETURN":::
2920 PRINT "ENTER REPORT FILE NAME:"
2930 INPUT F$
2940 IF F$="" THEN 230
2950 OPEN #1:F$,OUTPUT,INTERNAL
2960 PRINT #1:QP;MC;NC;MCA
2970 PRINT #1:RAP;QO;PANS
2980 FOR Z=1 TO QP
2990 PRINT #1:QR(Z,1);QR(Z,2)
3000 FOR Y=1 TO NC+2
3010 PRINT #1:QZ$(QR(Z,1),Y)
3020 NEXT Y
3030 NEXT Z
3040 CLOSE #1
3050 GOTO 230
3060 CALL CLEAR
3070 PRINT "LOAD QUIZ REPORT"::::
3080 PRINT "ENTER THE DEVICE NAME":"AND
     FILE NAME FOR DISKS":"OR":"CS1 FOR
     CASSETTE OPERATION"::
3090 PRINT "PRESS ENTER TO RETURN":::
3100 PRINT "ENTER REPORT FILE NAME:"
3110 INPUT F$
3120 IF F$="" THEN 230
3130 OPEN #1:F$,INPUT ,INTERNAL
3140 INPUT #1:QP,MC,NC,MCA
3150 INPUT #1:RAP,QO,PANS
3160 FOR Z=1 TO QP
3170 INPUT #1:QR(Z,1),QR(Z,2)
3180 FOR Y=1 TO NC+2
3190 INPUT #1:QZ$(QR(Z,1),Y)
3200 NEXT Y
3210 NEXT Z
3220 CLOSE #1
3230 GOTO 230
3240 CALL KEY(0,K,S)
3250 IF S=0 THEN 3240
3260 CALL SOUND(1,660,0)
3270 RETURN
3280 FOR TD=1 TO 200
3290 NEXT TD
3300 RETURN
3310 CALL CLEAR
3320 PRINT "ALL DONE"
3330 END
```

HCM

KORS-ELF

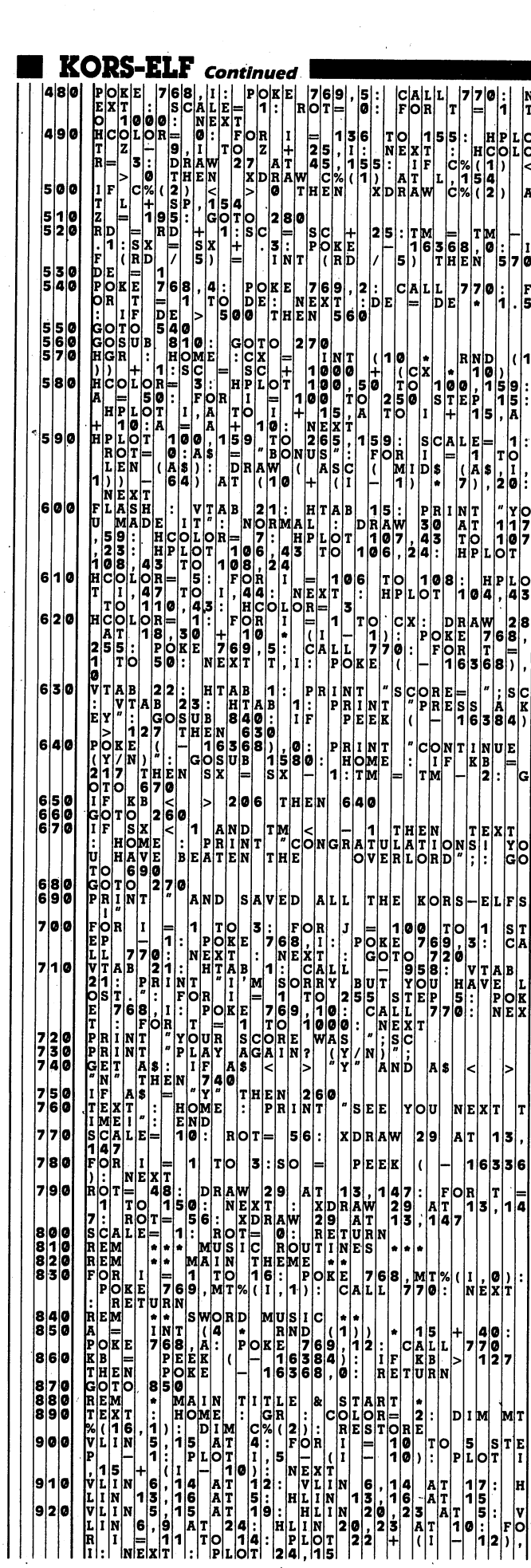
APPLE // Family

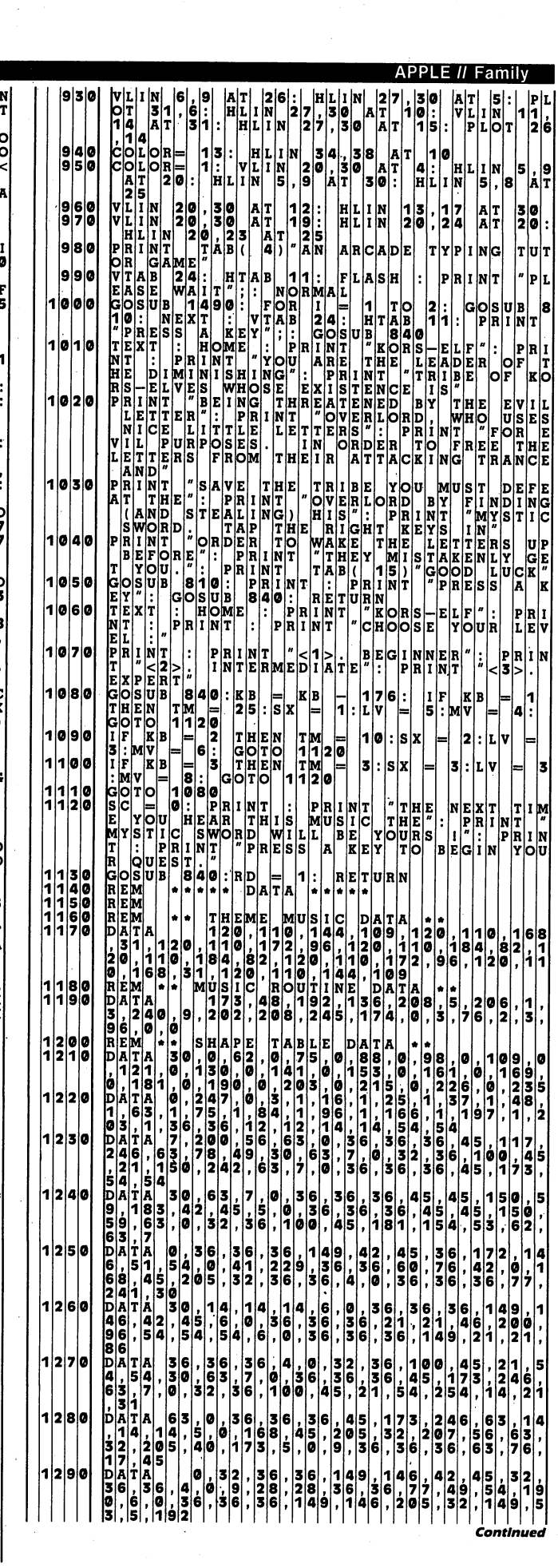
```
100 REM   *************
110 REM   * KORS-ELF  *
120 REM   *************
130 REM   COPYRIGHT 1984,1985
140 REM   EMERALD VALLEY PUBLISHING CO.
150 REM   BY SHAWN BLEVINS
160 REM     AND THE HCM STAFF
170 REM   HOME COMPUTER MAGAZINE
180 REM   VERSION 5.1.1
190 REM   APPLE II FAMILY APPLESOFT
200 REM
210 REM
220 IF  PEEK (103) = 1 AND  PEEK (104)
    = 64 THEN  GOTO 250
230 POKE 103,1: POKE 104,64: POKE 16384
    ,0: POKE 16385,0: POKE 16386,0
240 PRINT  CHR$ (4);"RUN KORSELF"
250 GOSUB 880
260 GOSUB 1060
270 Z = 195: GOSUB 1380
280 L = MV: FOR I = 1 TO 2:C%(I) =  INT
     (25 *  RND (1)) + 1:X = C%(I): IF
    X = C%(I - 1) THEN 280
290 NEXT :SP =  INT (3 *  RND (1)) + 4:
    SP = SP * 7
300 XDRAW C%(1) AT L - MV,154: XDRAW C%
    (2) AT L + SP - MV,154
310 DRAW 30 AT Z,155
320 FOR I = MV TO 0 STEP  - MV: IF C%(1
    ) = 0 THEN 340
330 XDRAW C%(1) AT L - I,154
340 IF C%(2) = 0 THEN 360
350 XDRAW C%(2) AT L + SP - I,154
360 NEXT
370 FOR TN = 1 TO TM:KB =  PEEK ( - 163
    84): IF KB < 128 THEN  NEXT TN: GOT
    O 410
380 POKE  - 16368,0
390 CH = KB - 192: IF CH = C%(1) AND C%
    (1) <  > 0 THEN SC = SC + 5: XDRAW
    C%(1) AT L,154:C%(1) = 0
400 IF CH = C%(2) AND C%(2) <  > 0 THEN
     SC = SC + 5: XDRAW C%(2) AT L + SP
    ,154:C%(2) = 0
410 IF C%(1) = 0 AND C%(2) = 0 THEN  GO
    SUB 770: GOTO 280
420 IF C%(1) <  > 0 THEN  IF L + 7 >  =
     Z THEN C%(1) = 0: GOTO 470
430 IF C%(2) <  > 0 THEN  IF L + 7 + SP
     >  = Z THEN C%(2) = 0: GOTO 470
440 L = L + MV: IF Z < 65 THEN 520
450 VTAB 22: HTAB 1: PRINT "SCORE= ";SC
    ;: TAB( 14)"LIVES= ";LV;: TAB( 26)"RO
    UND= ";RD: IF LV < 1 THEN 710
460 XDRAW 30 AT Z,155:Z = Z - 3: GOTO 3
    10
470 LV = LV - 1: SCALE= 20: FOR I = 136
     TO 155: XDRAW 29 AT Z - 2,I
```

Continued

PROGRAM LISTING

KORS-ELF *Continued*





Continued

KORS-ELF *Continued*


```
1300 DATA 36,36,36,4,0,0,44,40,40,56,56,56,32,7,7,49
     ,23,22,21,54,9,27,36,28,28,28,12,12,12,108,9,19
     ,0,6,0,45,45,220,27,12,12,12,60,60,60,96
1310 DATA 63,0,36,36,36,36,36,36,36,36,36,36,36,36,3
     ,63,44,54,45,54,54,54,54,54,54,54,54,54,54,63
     ,63,54,54,45,54,54,54,54,54,54,54,54,54,54,63
1320 DATA 88,88,63,63,63,88,63,88,88,88,88,88,88,64,
     88,73,73,84,88,41,53,63,30,30,30,54,21,85,45,45,
     20,55,32,73,45,228,4,0,36,44,45,36,36,36,45,45,
     0,54,5,54,45,6,45,63,45,45,45,45,45,45,45,45,54
1330 DATA 5,45,45,5,0,45,45,45,45,45,45,45,45,45,45,
     54,45,45,5,0,63,53,100,40,63,63,63,63,63,63,63,
     36,6,45,63,45,12,60,12,40,69,45,32,45,53,12,53,
     63,63,55,55,45,12,63,33,45,45,45,45,45,45,45,45
1340 DATA 54,54,21,63,32,36,63,63,63,63,63,63,63,63,
     63,63,54,54,21,63,32,55,63,45,100,45,36,45,45,36,
     6,63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,0,36,54,54,54,54,54,54,54,54,54,54,54,54,63
1350 DATA 54,54,21,63,32,36,63,63,63,63,63,63,63,63,
     63,63,54,54,21,63,32,55,63,45,100,45,36,45,45,36,
     6,63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,0,36,54,54,54,54,54,54,54,54,54,54,54,54,63
1360 DATA 54,54,62,63,7,0,UP,PCL,*,*,=,3,THEN,CL,=,
     63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,0,36,54,54,54,54,54,54,54,54,54,54,54,54,63
1370 DATA 54,54,62,EN,1:,SE,IF,PCL,=,3,THEN,CL,=,
     63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,0,36,54,54,54,54,54,54,54,54,54,54,54,54,63
1380 DATA 54,54,62,CL,=,8,THEN,CL,I,=,=,13,THEN,
     63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,0,36,54,54,54,54,54,54,54,54,54,54,54,54,63
1390 DATA IF,CL,=,8,THEN,CL,I,=,=,13,THEN,
     63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,0,36,54,54,54,54,54,54,54,54,54,54,54,54,63
1400 DATA CL,=,8,THEN,CL,I,=,=,13,THEN,
     63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,0,36,54,54,54,54,54,54,54,54,54,54,54,54,63
1410 DATA =,13,THEN,
     63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,0,36,54,54,54,54,54,54,54,54,54,54,54,54,63
1420 DATA 63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,63,54,54,63,54,54,54,63,63,100,45,36,45,45,36,
     6,0,36,54,54,54,54,54,54,54,54,54,54,54,54,63
```

```
1430 HOME:DRAW 27 AT 211,155:DRAW 27,15,0:DRAW 27,0,15:DRAW 27,
     -15,0:DRAW 27,0,-15:DRAW 27,-15,0:DRAW 27,0,-15:DRAW 27,15,
     0:DRAW 27,0,-15:DRAW 27,15,0:DRAW 27,0,15:DRAW 27,-15,0:
     DRAW 27,0,15:DRAW 27,15,0:DRAW 27,0,15:DRAW 27,-15,0:DRAW
     27,0,15:DRAW 27,15,0:DRAW 27,0,15:DRAW 27,-15,0:DRAW 27,0,
     15:DRAW 27,15,0:DRAW 27,0,15:DRAW 27,-15,0:DRAW 27,0,15:
     DRAW 27,15,0:DRAW 27,0,15:DRAW 27,-15,0:DRAW 27,0,15:DRAW
     27,15,0:DRAW 27,0,15:DRAW 27,-15,0:DRAW 27,0,15:DRAW 27,15,
     0:DRAW 27,0,15:DRAW 27,-15,0:DRAW 27,0,15:DRAW 27,15,0:
     DRAW 27,0,15:DRAW 27,-15,0:DRAW 27,0,15:DRAW 27,15,0:DRAW
     27,0,15:DRAW 27,-15,0:DRAW 27,0,15:DRAW 27,15,0:DRAW 27,0,
     15:DRAW 27,-15,0:DRAW 27,0,15:DRAW 27,15,0:DRAW 27,0,15:
     DRAW 27,-15,0:DRAW 27,0,15:DRAW 27,15,0:DRAW 27,0,15:DRAW
     27,-15,0:DRAW 27,0,15:DRAW 27,15,0:DRAW 27,0,15:DRAW 27,-15,
     0:DRAW 27,0,
```

KORS-ELF *Continued*


```
990 REM*****LETTER RELEASED********
1000 GOSUB 2740::S=40
1010 POKE54296,15:POKE2043,204
1020 POKE54277,3:POKE54276,0:S=S+1
1030 POKE54276,129:POKE54273,S:POKE54272,S:POKE54276,128
1040 POKE54296,0
1050 IFS=42THENGOSUB 2740::POKE2043,200:RETURN
1060 GOTO1010
1070 REM***KORS-ELF ZAPPED*******
1080 GOSUB 2740::S=40
1090 POKE54296,15
1100 POKE54277,3:POKE54276,0:S=S+10
1110 POKE54276,129:POKE54273,S:POKE54272,S:POKE54276,128
1120 IFS=240THENGOSUB 2740::RETURN
1130 GOTO 1100
1140 REM****SWORD MUSIC*************
1150 GOSUB 2740::S=40:POKE198,0
1160 POKE54296,15
1170 POKE54277,3:POKE54276,0
1180 POKE54276,17
1190 GETA$:IFA$<>""THENRETURN
1200 AZ=INT(RND(1)*4)+1
1210 IFAZ=1THENA=50:B=60:GOTO 1250
1220 IFAZ=2THENA=56:B=99:GOTO 1250
1230 IFAZ=3THENA=63:B=75:GOTO 1250
1240 IFAZ=4THENA=67:B=15:GOTO 1250
1250 POKE54273,A:POKE54272,B:POKE54276,16:GOTO 1170
1260 END
1270 REM**MAIN TITLE AND START********
1280 POKE53248+21,0
1290 GOSUB 2790
1300 PRINT"{SHIFT CLR}{3 CRSR DOWN}{CTRL BLU}";
1310 PRINT"  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}{CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  "
1320 PRINT"  {CTRL RVSON}  {CTRL RVSOFF} {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}"
1330 PRINT"  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}{CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}{CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}"
1340 PRINT"  {CTRL RVSON}  {CTRL RVSOFF} {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}{CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}{CTRL YEL}{CTRL BLU}"
1350 PRINT"  {CTRL RVSON}  {CTRL RVSOFF} {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}{CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}"
1360 PRINT"  {CTRL RVSON}  {CTRL RVSOFF} {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}"
1370 PRINT"{3 CRSR DOWN}{CTRL RED}";
1380 PRINT"      {CTRL RVSON}      {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}      {CTRL RVSOFF}"
1390 PRINT"      {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}"
1400 PRINT"      {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}"
1410 PRINT"      {CTRL RVSON}      {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}      {CTRL RVSOFF}"
1420 PRINT"      {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}"
1430 PRINT"      {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}"
1440 PRINT"      {CTRL RVSON}      {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}  {CTRL RVSON}  {CTRL RVSOFF}"
1450 PRINTSPC(4)"{CRSR DOWN}{CTRL GRN}AN ARCADE TYPING TUTOR GAME"
1460 PRINT"{HOME}           PLEASE WAIT":GOSUB 2450
1470 FORI=1TO2:GOSUB 830::NEXT:PRINT"{HOME}{CTRL WHT}       PRESS A KEY":GOSUB 1140
1480 PRINT"{SHIFT CLR}{CTRL RED}KORS-ELF{CTRL YEL} [£{CTRL RED}"
1490 PRINT"{CTRL BLU}YOU ARE THE LEADER OF THE DIMINISHING"
1500 PRINT"TRIBE OF KORS-ELFS WHOSE EXISTENCE IS"
1510 PRINT"BEING THREATENED BY THE EVIL LETTER"
1520 PRINT"OVERLORD,WHO USES NICE LITTLE LETTERS"
1530 PRINT"FOR EVIL PURPOSES. IN ORDER TO FREE THE"
1540 PRINT"LETTERS FROM THEIR ATTACKING TRANCE AND"
1550 PRINT"SAVE THE TRIBE YOU MUST DEFEAT THE"
1560 PRINT"OVERLORD BY FINDING (AND STEALING) HIS"
1570 PRINT"MYSTIC SWORD. TAP THE RIGHT KEYS IN"
1580 PRINT"ORDER TO WAKE THE LETTERS UP BEFORE"
1590 PRINT"THEY MISTAKENLY GET YOU."
1600 PRINTSPC(15)"{CTRL GRN}GOOD LUCK"
1610 GOSUB 830::PRINT"{CTRL WHT}PRESS A KEY":GOSUB 1140:RETURN
1620 GOSUB2790:PRINT"{SHIFT CLR}{CTRL RED}KORS-ELF"
1630 PRINT"{CTRL WHT}{CRSR DOWN}CHOOSE YOUR LEVEL:"
1640 PRINT"{CTRL GRN}{CRSR DOWN}   F1{CTRL RVSOFF}=BEGINNER {CTRL YEL}]-"
1650 PRINT"{2 CRSR DOWN}{CTRL YEL}   F3{CTRL RVSOFF}=INTERMEDIATE: {CTRL YEL}[£"
1660 PRINT"{2 CRSR DOWN}{CTRL RED}   F5{CTRL RVSOFF}=EXPERT: {CTRL YEL}[£ [£"
1670 SC=0:GOSUB 1140
1680 IFA$=CHR$(133)THENTM=100:SX=2:LV=5:GOTO 1720
1690 IFA$=CHR$(134)THENTM=40:SX=4:LV=3:GOTO 1720
1700 IFA$=CHR$(135)THENTM=10:SX=6:LV=3:GOTO 1720
1710 GOTO 1670
1720 PRINT"{CRSR DOWN}THE NEXT TIME YOU HEAR THIS MUSIC THE"
1730 PRINT"MYSTIC SWORD WILL BE YOURS."
1740 PRINT"{CRSR DOWN}{CTRL CYN}PRESS A KEY TO BEGIN THE QUEST."
1750 GOSUB 1140::RD=1:RETURN
1760 REM***MAIN THEME MUSIC DATA*****
1770 DATA15,210,18,209,15,210,21,31,15,210,22,96,15,210,23,181,15,210,23,181
1780 DATA15,210,22,96,15,210,21,31,15,210,18,209,-1,-1
1790 REM**KORS-ELF DATA******
1800 DATA0,120,0,0,252,0,1,254,0,3,251,0,7,248,192,2,8,192,2,68,0,7,4,0,1,68,0
1810 DATA0,130,0,25,254,0,31,154,0,7,153,0,0,153,192,0,219,192,0,129,0,0,72
1820 DATA128,0,84,64,0,82,32,1,243,224,1,243,224
1830 DATA0,120,0,0,252,0,1,254,0,3,251,0,7,248,192,2,8,96,2,68,0,7,4,0,1,68,0
1840 DATA0,130,0,0,254,0,0,154,0,1,153,0,3,153,192,0,255,192,0,153,0,0,130,0
1850 DATA0,170,0,0,146,0,1,246,0,3,158,0
1860 REM**CHARACTER DATA*****
1870 DATA0,132,202,177,128,213,64,127,0,35,85,137,1,170,4,252
1880 DATA127,64,63,16,35,67,32,31,234,2,252,4,194,193,2,252
1890 DATA170,85,127,63,31,15,7,3,170,85,254,252,248,240,224,192
1900 DATA255,255,255,255,255,255,255,255
1910 DATA0,0,0,0,28,0,0,30,0,0,36,0,0,34,16,0,124,56,0,71,240,0,84,16,0,84,16
1920 DATA32,72,16,112,72,16,248,127,56,255,255,255,0,0,17,63,255,57,66,0,197
1930 DATA153,0,85,153,0,69,66,0,249,63,255,1,255,255,255,0
1940 DATA0,255,0,3,255,192,7,255,224,7,255,224,15,255,240,15,255,240,15,255,240
1950 DATA31,255,248,63,255,252,63,255,252,63,255,252,63,255,252,127,255,254
1960 DATA127,255,254,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
1970 DATA255,255,255,255,255,255,255,0
1980 DATA255,255,255,128,0,1,189,221,225,165,85,33,165,85,225,189,221,17,128,0
1990 DATA1,129,255,1,128,108,1,128,198,1,129,131,1,131,1,129,131,1,129
2000 DATA129,131,1,128,254,1,128,124,1,128,0,1,128,0,1,128,0,1,255,255
2010 DATA255,0
2020 DATA0,8,0,0,57,0,0,57,0,0,57,0,0,57,0,0,57,0,0,57,0,0,57,0,0,57,0
```

Continued

KORS-ELF *Continued*

COMMODORE 64

```
2030 DATA 0,57,0,0,0,57,0,0,4,57,8,2,0,57,16,0,1,
     DATA 85,96,0,170,0,128,0,0,255,192,0,12,0,0,0,
     DATA 12,0
2040 DATA 0,12,0,0,0,21,0,0,8
2050 DATA 255,1,128,1,128,1,128,1,128,1,128,1,128,255
2060 DATA 255,255,1,1,0,0,0,0,0,0,0,0,255,255
2070 DATA 0,0,0,0,0,28,0,0,0,0,30,0
2080 DATA 0,0,0,0,0,36,0,0,0,34,0,64,0,0
2090 DATA 0,0,0,0,0,24,0,0,96,84,0,96,32,71,32,6,
2100 DATA 0,0,0,0,0,132,24,0,224,48,225,112,255,72,
2110 DATA 0,0,0,0,0,16,17,0,0,0,0,0,0,0,0,0,0,
2120 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2130 DATA 0,0,0,0,0,153,0,0,0,0,0,0,0,0,0,0,0,
2140 DATA 0,0,0,0,0,249,0,0,0,0,0,0,0,0,0,0,0,
2150 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2170 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2180 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2190 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2200 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2210 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2220 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2230 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2240 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2250 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2260 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2270 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2280 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2290 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2300 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2310 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2320 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2330 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2340 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2350 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2360 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2370 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2380 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2390 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2400 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2410 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2420 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2430 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2440 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2450 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2460 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2470 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2480 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2490 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2500 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2510 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2520 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2530 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2540 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2550 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2560 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2570 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2580 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2590 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2600 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2610 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2620 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2630 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2640 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2650 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2660 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2670 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2680 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2690 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2700 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2710 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2720 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2730 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2740 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2750 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2760 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2770 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2780 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2790 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2800 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2810 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2820 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2830 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2840 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2850 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2860 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2870 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2880 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2890 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2900 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2910 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2920 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2930 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2940 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2950 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2960 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2970 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2980 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
2990 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3000 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3010 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3020 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3030 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3040 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3050 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3060 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3070 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3080 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3090 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3100 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3110 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3120 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3130 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3140 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3150 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3170 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3180 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3190 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3200 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3210 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3220 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3230 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3240 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3250 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3260 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3270 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3280 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3290 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3300 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3310 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3320 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3330 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3340 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3350 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3360 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3370 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3380 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3390 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3400 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3410 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3420 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3430 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3440 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3450 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3460 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3470 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3480 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3490 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3500 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3510 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3520 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3530 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3540 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3550 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3560 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3570 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3580 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3590 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3600 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3610 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3620 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3630 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3640 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3650 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3660 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3670 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3680 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3690 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3700 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3710 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3720 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3730 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3740 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3750 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3760 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3770 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3780 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3790 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3800 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3810 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3820 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3830 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3840 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3850 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3860 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3870 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3880 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3890 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3900 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3910 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3920 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3930 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3940 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3950 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3960 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3970 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
3980 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3990 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4000 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4010 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
4020 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4030 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4040 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4050 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4060 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4070 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4080 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4090 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4100 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4110 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4120 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4130 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4140 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4150 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4170 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4180 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4190 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4200 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4210 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4220 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4230 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4240 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4250 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4260 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4270 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4280 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4290 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4300 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4310 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4320 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4330 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4340 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4350 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4360 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4370 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4380 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4390 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4400 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4410 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4420 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4430 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4440 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4450 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4460 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4470 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4480 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4490 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4500 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4510 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4520 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4530 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4540 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4550 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4560 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4570 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4580 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4590 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4600 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4610 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4620 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4630 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4640 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4650 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4660 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4670 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4680 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4690 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4700 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4710 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4720 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4730 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4740 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4750 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4760 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4770 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4780 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4790 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4800 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4810 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4820 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4830 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4840 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4850 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4860 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4870 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4880 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4890 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4900 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4910 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4920 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4930 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4940 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4950 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4960 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4970 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4980 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4990 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5000 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5010 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5020 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5030 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5040 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5050 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5060 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5070 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5080 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5090 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5100 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5110 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5120 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5130 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5140 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5150 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5170 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5180 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5190 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5200 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5210 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5220 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5230 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5240 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5250 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5260 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5270 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5280 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5290 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5300 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5310 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5320 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5330 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5340 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5350 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5360 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5370 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5380 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5390 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5400 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5410 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5420 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5430 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5440 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5450 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5460 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5470 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5480 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5490 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5500 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5510 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5520 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5530 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5540 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5550 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5560 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5570 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5580 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5590 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5600 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5610 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5620 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5630 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5640 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5650 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

HCM

Continued

KORS-ELF

IBM PC & IBM PCjr

```
100 '
110 '
120 '
130 '
140 '
150 '
160 '
170 '
180 '
190 '
200 '
210 '
220 '
230 '
240 '
250 '
260 '
270 '
280 '
290 '
300 '
310 '
320 '
330 '
340 '
350 '
360 '
370 '
380 '
390 '
400 '
410 '
420 '
430 '
440 '
450 '
460 '
470 '
480 '
490 '
500 '
510 '
520 '
530 '
540 '
550 '
560 '
570 '
580 '
590 '
600 '
610 '
620 '
630 '
640 '
650 '
```

Continued

KORS-ELF *Continued*


```
6600 REM
6700 CLS
6800 PRINT"                    Kors-Elf"
6900 PRINT"      Kors-Elf is the leader of the dark"
7000 PRINT"Knights of Kors-Elf who uses nice little"
7100 PRINT"Overlord, who uses evil purposes. In order"
7200 PRINT"to rid the evil Overlord by finding (and stealing)"
7300 PRINT"the magic sword. Tap the right"
7400 PRINT"arrow keys to move the Letters up"
7500 PRINT"and down to make them spell out the"
7600 PRINT"Overlord's instructions. Be"
7700 PRINT"careful not to make a mistake or they will"
7800 PRINT"Overlord's instructions. Be"
7900 PRINT"careful not to make a mistake or they will"
8000 PRINT"get you. Good luck!"
8100 PRINT"                    Press any key to begin"
8200 PRINT"                    Press any key to begin"
8300 PRINT"                    Press any key to begin"
8400 PRINT"                    Press any key to begin"
8500 PRINT"                    Press any key to begin"
8600 PRINT"                    Press any key to begin"
8700 PRINT"                    Press any key to begin"
8800 PRINT"                    Press any key to begin"
8900 PRINT"                    Press any key to begin"
9000 PRINT"                    Press any key to begin"
9100 PRINT"                    Press any key to begin"
9200 PRINT"                    Press any key to begin"
9300 PRINT"                    Press any key to begin"
9400 PRINT"                    Press any key to begin"
9500 PRINT"                    Press any key to begin"
9600 PRINT"                    Press any key to begin"
9700 GOTO 2950
9800 REM
9900 REM  Rick Wall  Detail
10000 CLS
10010 FOR I = 1 TO 14
10020 LINE (0,(I-1)*10)-(318,(I-1)*10),3,
10030 NEXT I
10040 FOR I = 12 TO 0
10050 FOR J = 18 TO 0
10060 IF I/2 = INT(I/2) THEN O=9 ELSE O=0
10070 LINE (J*17+O,I*10)-(J*17+O,I*10+10),3,BF
10080 NEXT J,I
10090 REM  Door in center
10100 REM  Black Door
10110 LINE (150,30)-(170,60),,BF
10120 LINE (150,30)-(170,60),3,B
10130 REM  Torch Shelves
10140 REM  Torch
10150 LINE (70*RND(1)+30,SY)-(70*RND(1)+30,SY+5),2,BF
10160 LINE (290-SX,SY)-(SX,SY+5),2,BF
10170 LINE (SX+7,SY)-(SX+15,SY-15):DRAW "R8H5F5G5L8"
10180 LINE (297-SX,SY-15)-(SX,SY-17):DRAW "R3H5F5G5L3"
10190 LINE (305-SX,SY-15):DRAW "R3H5F5G5L3"
10200 REM  Crown
10210 REM  Pot Shelf
10220 LINE (50*RND(1)+60,SY+7),2,BF
10230 LINE (200-SX+60,SY)-(SX+70,SY+20),,BF
10240 LINE (SX+7,SY-10)-(SX+20,SY-9):DRAW "C3"
10250 LINE (SX+42,SY-10)-(SX+55,SY-2),,B
10260 LINE (SX+45,SY-9):DRAW "R3H5F5G5L3"
10270 REM  Set Draw
10280 REM  Melt Draw
10290 LINE (0,174),3,BF
10300 LINE (280,135)-(313,138),,B
10310 LINE (280,135)-(313,138):DRAW "R33D3L33U3"
10320 LINE (280,135)-(313,138):DRAW "R33D3L33U3"
10330 LINE (271,168)-(257,135):GOSUB 1500
10340 GET (257,135)-(271,168),J
10350 PUT (RND(1)*280,RND(1)*135),J
10360 IF INKEY$="" THEN 10350
10370 CLS:GOSUB 2250
10380 REM  Continue
10390 REM  Set Up Variables
10400 REM  Set Up Screen
10410 REM  Set Up Graphics
10420 REM  Set Up Sound
10430 REM  Set Up Keyboard
10440 REM  Set Up Joystick
10450 REM  Set Up Mouse
10460 REM  Set Up Light Pen
10470 REM  Set Up Paddles
10480 REM  Set Up Music
10490 REM  Wizard Machine Lever
10500 REM  Hall Machine
10510 REM  Line
10520 LINE (22,167)-(22,160),,LT
10530 LINE (22,160)-(160,160),,LTTER1
10540 LINE (160,160)-(160,167),,LTTER1
10550 LINE (160,167)-(22,167),,LTTER1
10560 REM  Set Up Color
10570 REM  Set Up Sprites
10580 REM  Set Up Characters
10590 REM  Set Up Fonts
10600 REM  Set Up Windows
10610 REM  Set Up Menus
10620 REM  Set Up Dialog Boxes
10630 REM  Set Up Error Handling
10640 REM  Set Up Interrupts
10650 REM  Set Up Timers
10660 REM  Set Up Clocks
10670 REM  Set Up Calendars
10680 REM  Set Up Printers
10690 REM  Set Up Plotters
10700 REM  Set Up Modems
10710 REM  Set Up Networks
10720 REM  Set Up Communications
10730 REM  Set Up Data Bases
10740 REM  Set Up Spreadsheets
10750 REM  Set Up Word Processors
10760 REM  Set Up Desktop Publishing
10770 REM  Set Up Presentation Graphics
10780 REM  Set Up Project Management
10790 REM  Set Up Personal Information Managers
10800 REM  Set Up Integrated Software
10810 REM  Set Up Utilities
10820 REM  Set Up Operating Systems
10830 REM  Set Up Languages
10840 REM  Set Up Compilers
10850 REM  Set Up Interpreters
10860 REM  Set Up Assemblers
10870 REM  Set Up Debuggers
10880 REM  Set Up Editors
10890 REM  Set Up Linkers
10900 REM  Set Up Loaders
10910 REM  Set Up Libraries
10920 REM  Set Up Graphics Libraries
10930 REM  Set Up Sound Libraries
10940 REM  Set Up Keyboard Libraries
10950 REM  Set Up Joystick Libraries
10960 REM  Set Up Mouse Libraries
10970 REM  Set Up Light Pen Libraries
10980 REM  Set Up Paddle Libraries
10990 REM  Set Up Music Libraries
11000 REM  Set Up Wizard Machine Lever Libraries
11010 REM  Set Up Hall Machine Libraries
11020 REM  Set Up Line Libraries
11030 REM  Set Up Color Libraries
11040 REM  Set Up Sprite Libraries
11050 REM  Set Up Character Libraries
11060 REM  Set Up Font Libraries
11070 REM  Set Up Window Libraries
11080 REM  Set Up Menu Libraries
11090 REM  Set Up Dialog Box Libraries
11100 REM  Set Up Error Handling Libraries
11110 REM  Set Up Interrupt Libraries
11120 REM  Set Up Timer Libraries
11130 REM  Set Up Clock Libraries
11140 REM  Set Up Calendar Libraries
11150 REM  Set Up Printer Libraries
11160 REM  Set Up Plotter Libraries
11170 REM  Set Up Modem Libraries
11180 REM  Set Up Network Libraries
11190 REM  Set Up Communication Libraries
11200 REM  Set Up Data Base Libraries
11210 REM  Set Up Spreadsheet Libraries
11220 REM  Set Up Word Processor Libraries
11230 REM  Set Up Desktop Publishing Libraries
11240 REM  Set Up Presentation Graphics Libraries
11250 REM  Set Up Project Management Libraries
11260 REM  Set Up Personal Information Manager Libraries
11270 REM  Set Up Integrated Software Libraries
11280 REM  Set Up Utility Libraries
11290 REM  Set Up Operating System Libraries
11300 REM  Set Up Language Libraries
11310 REM  Set Up Compiler Libraries
11320 REM  Set Up Interpreter Libraries
11330 REM  Set Up Assembler Libraries
11340 REM  Set Up Debugger Libraries
11350 REM  Set Up Editor Libraries
11360 REM  Set Up Linker Libraries
11370 REM  Set Up Loader Libraries
11380 REM  Set Up Library Libraries
11390 REM  Set Up Graphics Library Libraries
11400 REM  Set Up Sound Library Libraries
11410 REM  Set Up Keyboard Library Libraries
11420 REM  Set Up Joystick Library Libraries
11430 REM  Set Up Mouse Library Libraries
11440 REM  Set Up Light Pen Library Libraries
11450 REM  Set Up Paddle Library Libraries
11460 REM  Set Up Music Library Libraries
11470 REM  Set Up Wizard Machine Lever Library Libraries
11480 REM  Set Up Hall Machine Library Libraries
11490 REM  Set Up Line Library Libraries
11500 REM  Set Up Color Library Libraries
11510 LINE (22,145)-(29,148),3
11520 LINE (29,148),3
11530 LINE (29,148)-(22,145),3
11540 LINE (22,145)-(22,160),3
11550 LINE (22,160)-(29,148),3
11560 LINE (29,148)-(22,145),3
11570 LINE (22,145)-(22,160),3
11580 LINE (22,160)-(29,148),3
11590 LINE (29,148)-(22,145),3
11600 LINE (22,145)-(22,160),3
11610 LINE (22,160)-(29,148),3
11620 LINE (29,148)-(22,145),3
11630 LINE (22,145)-(22,160),3
11640 LINE (22,160)-(29,148),3
11650 LINE (29,148)-(22,145),3
11660 LINE (22,145)-(22,160),3
11670 LINE (22,160)-(29,148),3
11680 LINE (29,148)-(22,145),3
11690 LINE (22,145)-(22,160),3
11700 LINE (22,160)-(29,148),3
11710 LINE (29,148)-(22,145),3
11720 LINE (22,145)-(22,160),3
11730 LINE (22,160)-(29,148),3
11740 LINE (29,148)-(22,145),3
11750 LINE (22,145)-(22,160),3
11760 LINE (22,160)-(29,148),3
11770 LINE (29,148)-(22,145),3
11780 LINE (22,145)-(22,160),3
11790 LINE (22,160)-(29,148),3
11800 LINE (29,148)-(22,145),3
11810 LINE (22,145)-(22,160),3
11820 LINE (22,160)-(29,148),3
11830 LINE (29,148)-(22,145),3
11840 LINE (22,145)-(22,160),3
11850 LINE (22,160)-(29,148),3
11860 LINE (29,148)-(22,145),3
11870 LINE (22,145)-(22,160),3
11880 LINE (22,160)-(29,148),3
11890 LINE (29,148)-(22,145),3
11900 LINE (22,145)-(22,160),3
11910 LINE (22,160)-(29,148),3
11920 LINE (29,148)-(22,145),3
11930 LINE (22,145)-(22,160),3
11940 LINE (22,160)-(29,148),3
11950 LINE (29,148)-(22,145),3
11960 LINE (22,145)-(22,160),3
11970 LINE (22,160)-(29,148),3
11980 LINE (29,148)-(22,145),3
11990 LINE (22,145)-(22,160),3
12000 LINE (22,160)-(29,148),3
12010 LINE (29,148)-(22,145),3
12020 LINE (22,145)-(22,160),3
12030 LINE (22,160)-(29,148),3
12040 LINE (29,148)-(22,145),3
12050 LINE (22,145)-(22,160),3
12060 LINE (22,160)-(29,148),3
12070 LINE (29,148)-(22,145),3
12080 LINE (22,145)-(22,160),3
12090 LINE (22,160)-(29,148),3
12100 LINE (29,148)-(22,145),3
12110 LINE (22,145)-(22,160),3
12120 LINE (22,160)-(29,148),3
12130 LINE (29,148)-(22,145),3
12140 LINE (22,145)-(22,160),3
12150 LINE (22,160)-(29,148),3
12160 LINE (29,148)-(22,145),3
12170 LINE (22,145)-(22,160),3
12180 LINE (22,160)-(29,148),3
12190 LINE (29,148)-(22,145),3
12200 LINE (22,145)-(22,160),3
12210 LINE (22,160)-(29,148),3
12220 LINE (29,148)-(22,145),3
12230 LINE (22,145)-(22,160),3
12240 LINE (22,160)-(29,148),3
12250 LINE (29,148)-(22,145),3
12260 LINE (22,145)-(22,160),3
12270 LINE (22,160)-(29,148),3
12280 LINE (29,148)-(22,145),3
12290 LINE (22,145)-(22,160),3
12300 LINE (22,160)-(29,148),3
12310 LINE (29,148)-(22,145),3
12320 LINE (22,145)-(22,160),3
12330 LINE (22,160)-(29,148),3
12340 LINE (29,148)-(22,145),3
12350 LINE (22,145)-(22,160),3
12360 LINE (22,160)-(29,148),3
12370 LINE (29,148)-(22,145),3
12380 LINE (22,145)-(22,160),3
```

Continued

KORS-ELF *Continued*

IBM PC & IBM PCjr

```
2690 PRINT "Your score was ";SC
2700 PRINT "Press to play again, N to
2710 PRINT "quit" B$="Y"
2720 IF A$="n" THEN CLS:PRINT
2730 IF A$="y" THEN RUN ELSE 2
2740
2750
2760
2770
2780
2790
2800
2810
2820
2830
2840
2850
2860
2870
2880
2890
2900
2910
2920
2930
2940
2950
2960
2970
```

HCM

KORS-ELF

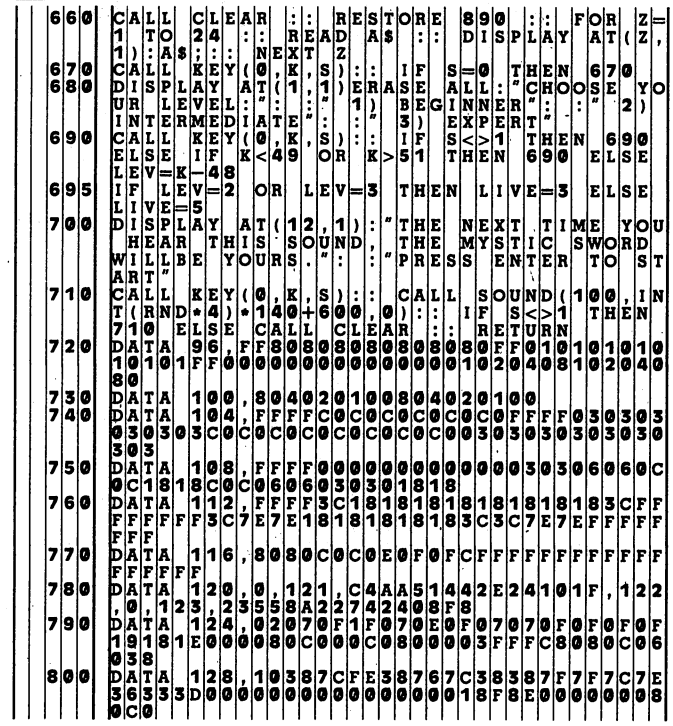
TI-99/4A

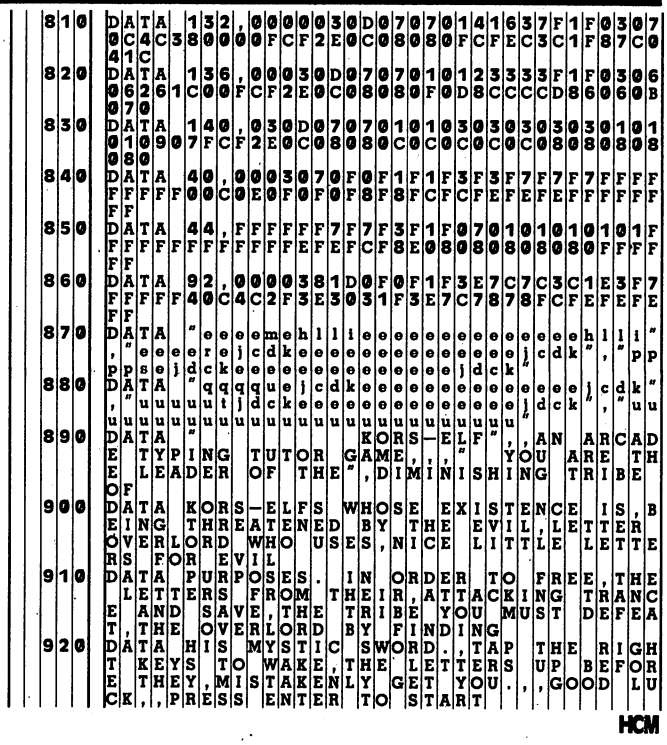
```
430 ALL SOUND(-100,1
440 GOSUB 550:INT(
450 GOTO 550:RUN ELSE 2
460 CALL CHARSET :: CALL
470 CALL CHAR(33,"01
480 CALL SPRITE(#1,136,2,72,#21,32,1
490 DISPLAY AT(1,1):"P
500 DISPLAY AT(12,1):"ERASE
510 DISPLAY AT(23,1):"PRESS EN
520 DISPLAY AT(12,1):"YOU
530 DISPLAY AT(12,1):"ERASE
540
550 CALL SPRITE(#4,124,14,105,24,
560 FOR Z=1 TO 13 :: STEP 2
570 TO 21 :: NEXT
580 SPRITE(#9,40,2,IN
590 #6,92,7,17
600
610 CALL HCHAR(14
620 CALL HCHAR(1
630 DISPLAY AT(24,8
640 DISPLAY
650 CALL KEY(0,K,S)::: IF S=0 THEN 650
```

Continued

KORS-ELF *Continued*

TI-99/4A

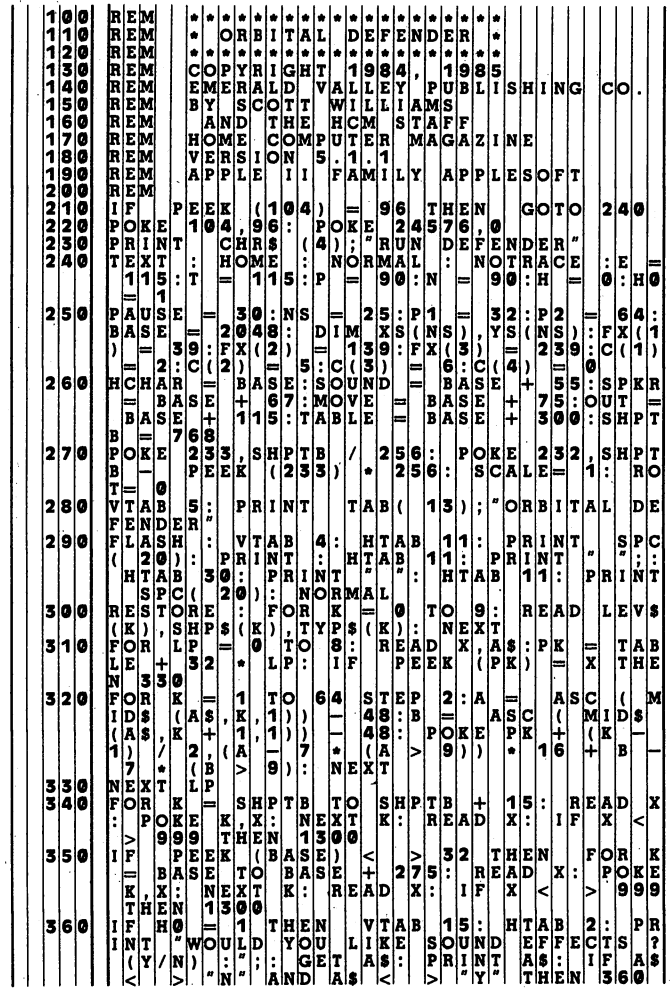


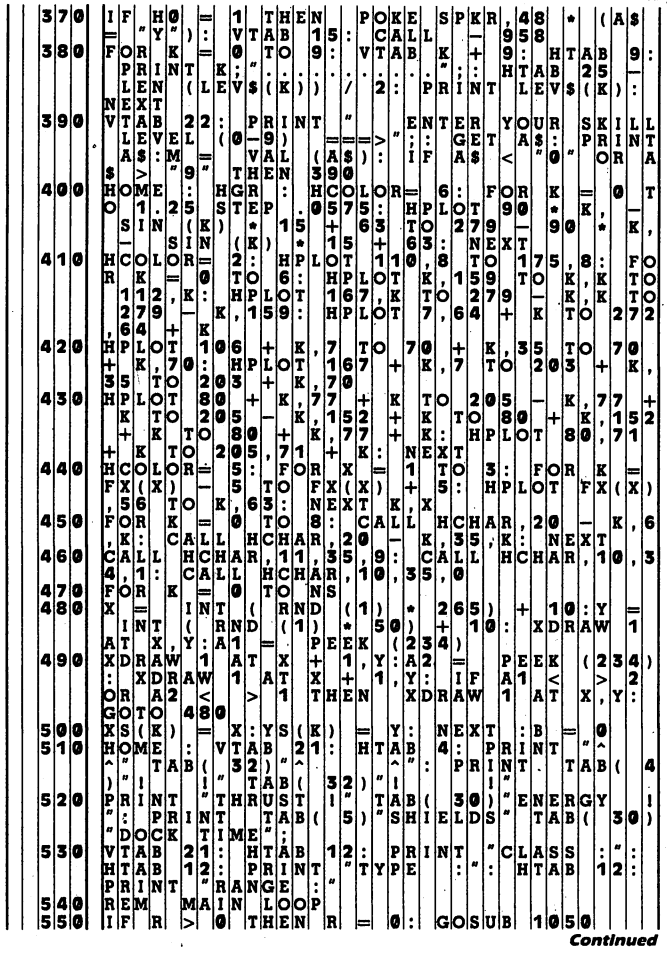


HCM

ORBITAL DEFENDER

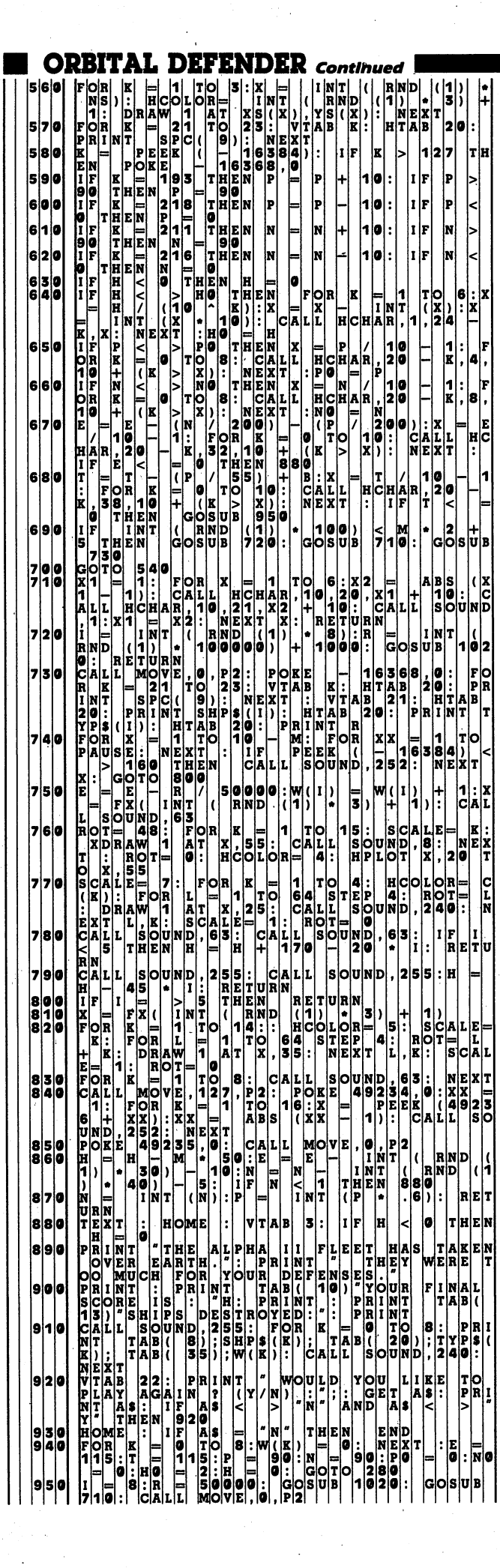
APPLE // Family





Continued

ORBITAL DEFENDER *Continued*



ORBITAL DEFENDER

```
100 REM **********************
110 REM * ORBITAL DEFENDER *
120 REM **********************
130 REM COPYRIGHT 1984,1985
140 REM EMERALD VALLEY PUBLISHING CO.
150 REM BY SCOTT WILLIAMS
          AND THE HCM STAFF
160 REM HOME COMPUTER MAGAZINE
170 REM VERSION 5.1.1
180 REM C-64 BASIC
190 REM DISPLAY TITLE SCREEN
200 POKE 53281,.:POKE 53280,.:POKE 53272,21:POKE 657,128:POKE 56,157:CLR
210 T$="ORBITAL DEFENDER"
220 PRINT"{SHIFT CLR}{CTRL YEL}{10 CRSR DOWN}        ** "T$" **"
230 PRINT"{CTRL GRN}{12 CRSR DOWN}   PRESS{SPACE}RETURN{SPACE}TO{SPACE}CONTINUE{CTRL YEL}":X=12:Y=10:S=1
240 GET K$:GOSUB1560:S=ABS(S-1):PRINT"{CTRL RVSON}";:IF S=1 THEN PRINT"{CTRL RVSOFF}";
250 FOR I=1 TO 100:NEXT:PRINTT$:IF K$="" THEN240
260 IF ASC(K$)<>13 THEN240
270 REM INTITIALIZE PROGRAM
280 PRINT"{SHIFT CLR}{CTRL YEL}{8 CRSR DOWN}     PLEASE WAIT WHILE I PREPARE YOUR"
290 PRINT"{CRSR DOWN}               SPACESHIP"
300 R=RND(-1):E=110:T=E:P=90:N=90:DIM ES$(1,9):B$="         "
310 V1(0)=54276:V1(1)=85:V2(0)=54283:V2(1)=129:V3(0)=54290:V3(1)=129
320 POKE 2040,11:POKE 2041,13:POKE 2042,14:POKE 2043,14:POKE 2044,15
330 POKE 53271,8:POKE 53276,49:POKE 53277,8:POKE 53285,0:POKE 53286,6
340 POKE 53287,2:POKE 53288,2:POKE 53289,2:POKE 53290,2:POKE 53291,0
350 POKE 53248,172:POKE 53249,174:G(0)=53256:F(0)=53250
360 F(1)=65:F(2)=172:F(3)=25:POKE 53251,75:POKE 53253,75:POKE 53255,65
370 G(1)=64:G(2)=171:G(3)=24:POKE 53257,92
380 POKE 2045,11:POKE 53258,172:POKE 53259,75:POKE 53292,2
390 FOR I=1 TO 9:READ SC$(I),ST$(I):NEXT
400 FOR I=0 TO 575:READ D:POKE 40192+I,D:NEXT
410 FOR I=0 TO 191:READ D:POKE 832+I,D:NEXT
420 FOR I=0 TO 24:READ D:POKE 54272+I,D:NEXT
430 FOR I=7 TO 9:ES$(0,I)="{CTRL RVSON}{CTRL BLU} {CTRL GRN}  {CTRL BLU} {CTRL GRN}  {CTRL BLU} {CTRL GRN}  {CTRL BLU}"
440 ES$(1,I)="{CTRL RVSON}{CTRL BLU} {CTRL CYN}  {CTRL BLU} {CTRL CYN}  {CTRL BLU} {CTRL CYN}  {CTRL BLU}"
450 ES$(0,I-6)="{CTRL RVSON}{CTRL BLU}{CTRL CYN}  {CTRL BLU} {CTRL CYN}  {CTRL BLU} {CTRL CYN}  {CTRL BLU}"
460 ES$(1,I-6)="{CTRL RVSON}{CTRL BLU}{CTRL GRN}  {CTRL BLU} {CTRL GRN}  {CTRL BLU} {CTRL GRN}  {CTRL BLU}":NEXT
470 ES$(0,4)="{CTRL RVSON}{CTRL BLU} {CTRL CYN}{3 CMDR Y}{CTRL BLU}{CTRL CYN}{CTRL BLU}{CTRL CYN}{2 CMDR T}{CTRL BLU}"
480 ES$(0,5)="{CTRL RVSON}{CTRL BLU} {CTRL CYN}{3 CMDR Y}{3 CMDR T}{CMDR Y}{SHIFT P}{CTRL GRN} {CTRL BLU}{CTRL CYN}{SHIFT O}{CMDR T} {6 CMDR T}{CTRL BLU}{CTRL CYN}{3 CMDR T}{2 CMDR Y}{SHIFT P}{CTRL GRN}{3 CMDR P}{2 CMDR @}{CTRL BLU}"
490 ES$(0,6)="{CTRL RVSON}{CTRL BLU} {CTRL GRN} {CTRL BLU}{CTRL GRN}{12 CMDR @}{CTRL BLU}{CTRL GRN}{6 CMDR @}{CTRL BLU}"
500 ES$(1,6)="{CTRL RVSON}{CTRL BLU} {CTRL CYN} {CTRL BLU}{CTRL CYN}{3 CMDR @}{4 CMDR P}{4 CMDR @}{CTRL BLU}{CTRL CYN} {CTRL BLU}"
510 ES$(1,5)="{CTRL RVSON}{CTRL BLU} {CTRL CYN}{5 CMDR @}{3 CMDR P}{CTRL GRN}{SHIFT O}{CTRL BLU}{CTRL GRN}{4 CMDR T}{3 CMDR Y}{CTRL BLU}{CTRL GRN}{CMDR Y}{SHIFT P}{CTRL CYN}{CMDR @}{CTRL BLU}"
520 ES$(1,4)="{CTRL RVSON}{CTRL BLU} {CTRL GRN}{2 CMDR Y}{3 CMDR T} {CTRL BLU}{CTRL GRN} {CTRL BLU}{CTRL GRN}{CMDR T}{CTRL CYN}{SHIFT L}{2 CMDR P}{CTRL GRN}{SHIFT O}{SHIFT P}{CTRL CYN}{2 CMDR P}{CTRL BLU}"
530 REM DISPLAY LEVEL OPTION SCREEN
540 POKE 198,.
550 PRINT"{SHIFT CLR}{CTRL GRN}      SKILL LEVEL":PRINT"{2 CRSR DOWN}{CTRL YEL}        0 - ROOKIE CADET"
560 PRINT"{CRSR DOWN}        1 - SPACE COOK'S ASSISTANT":PRINT"{CRSR DOWN}        2 - SPACE COOK"
570 PRINT"{CRSR DOWN}        3 - RADIO OPERATOR":PRINT"{CRSR DOWN}        4 - GUNNERY OFFICER"
580 PRINT"{CRSR DOWN}        5 - NAVIGATOR'S ASSISTANT":PRINT"{CRSR DOWN}        6 - NAVIGATOR"
590 PRINT"{CRSR DOWN}        7 - HELMSMAN":PRINT"{CRSR DOWN}        8 - CAPTAIN"
600 PRINT"{CRSR DOWN}        9 - COMMANDER":PRINT"{2 CRSR DOWN}{CTRL GRN}YOUR CHOICE? ";
610 GET K$:IF K$="" THEN610
620 IF ASC(K$)<48 OR ASC(K$)>57 THEN610
630 PRINTK$;:M=VAL(K$)
640 REM SCREEN DISPLAY ROUTINE
650 PRINT"{SHIFT CLR}{CTRL BLU}{CTRL RVSON}{SHIFT £}          {CMDR *}";
660 PRINT"{CTRL RVSON} {CTRL RVSOFF} {CTRL RVSON} {CTRL RVSOFF} {CTRL RVSON} {CTRL RVSOFF} {CTRL RVSON}";
670 PRINT" {CTRL RVSOFF} {CTRL RVSON} {CTRL RVSOFF} {CTRL RVSON} {CTRL RVSOFF} {CTRL RVSON}";
680 PRINT" {CTRL RVSOFF} {CTRL RVSON} {CTRL RVSOFF} {CTRL RVSON} {CTRL RVSOFF} {CTRL RVSON}";
690 PRINT" {CTRL RVSOFF} {CTRL RVSON} {CTRL RVSOFF} {CTRL RVSON} {CTRL RVSOFF} {CTRL RVSON}";
700 PRINT" {CTRL GRN} {CTRL BLU}{CTRL GRN} {CTRL BLU}{CTRL GRN} {CTRL BLU}";
710 PRINT" {CTRL GRN} {CTRL BLU}{CTRL GRN} {CTRL BLU}{CTRL GRN} {CTRL BLU}";
720 PRINT" {CTRL GRN} {CTRL BLU}{CTRL GRN} {CTRL BLU}{CTRL GRN} {CTRL BLU}";
730 PRINT"            ";
740 PRINT"{CTRL RVSOFF}{CTRL YEL}{CMDR A}{6 SHIFT C}{CMDR S}{CTRL RED}{CMDR R}{CTRL YEL}{CTRL BLU}{CTRL RVSON}{CTRL WHT}{CTRL BLU}{CTRL RVSOFF}{CTRL YEL}{CMDR A}{6 SHIFT C}{CMDR S}"
750 PRINT"{CTRL RVSOFF}{SHIFT B}THRUST{CMDR Z}{2 SHIFT C}{CMDR S}{CTRL RED}{SHIFT B}{CTRL YEL}{CTRL RVSON}{CTRL BLU}{CTRL WHT}{CTRL BLU}{CTRL RVSOFF}{CTRL YEL}{SHIFT B}CLASS:{CMDR Z}{2 SHIFT C}{CMDR S}"
760 PRINT"{SHIFT B}{CTRL RED}{CTRL RVSON}{CTRL YEL}{CTRL RVSOFF}{SHIFT B}{CTRL RED}{SHIFT B}{CTRL YEL}{CTRL RVSON}{CTRL BLU} WARNING↑ {CTRL RVSOFF}{CTRL YEL}{SHIFT B}{SHIFT B}"
770 PRINT"{SHIFT B}012345678{SHIFT B}{CTRL RED}{SHIFT B}{CTRL YEL}{CMDR Z}{9 SHIFT C}{CMDR X}"
780 PRINT"{CMDR Z}{9 SHIFT C}{CMDR X}{CTRL RED}{SHIFT B}{CTRL YEL}{CTRL RVSON}{CTRL BLU}{SHIFT £}{CMDR *}{CTRL RVSOFF}{CTRL RED}{CMDR Q}{9 SHIFT C}{CMDR W}"
790 PRINT"{CTRL YEL}{CMDR A}{7 SHIFT C}{CMDR S}{CTRL RED}{SHIFT +}{CTRL YEL}{CTRL RVSON}{CTRL BLU}{CMDR PUR}{CTRL BLU}{CTRL RVSOFF}{CTRL YEL}{CMDR A}{5 SHIFT C}{CMDR S}"
800 PRINT"{SHIFT B}SHIELDS{CMDR Z}{SHIFT C}{CMDR S}{CTRL RED}{SHIFT B}{CTRL YEL}{CTRL RVSON}{CTRL BLU}{CMDR PUR}{CTRL BLU}{CTRL RVSOFF}{CTRL YEL}{SHIFT B}TYPE:{CMDR Z}{3 SHIFT C}{CMDR S}"
810 PRINT"{SHIFT B}{CTRL RVSON}{CTRL RED}{CTRL RVSOFF}{CTRL YEL}{SHIFT B}{CTRL RED}{SHIFT B}{CTRL YEL}{CTRL RVSON}{CTRL BLU}{CMDR PUR}{CTRL BLU}{CTRL RVSOFF}{CTRL YEL}{SHIFT B}{SHIFT B}"
820 PRINT"{SHIFT B}012345678{SHIFT B}{CTRL RED}{SHIFT B}{CTRL YEL}{CTRL RVSON}{CTRL BLU}{CMDR PUR}{CTRL BLU}{CTRL RVSOFF}{CTRL YEL}{CMDR Z}{9 SHIFT C}{CMDR X}"
```

Continued

```
830 PRINT"{CMDR Z}{9 SHIFT C}{CMDR X}{CTRL RED}{SHIFT B}{CTRL YEL}{CTRL RVS ON}{CTRL BLU}{CMDR PUR}{CTRL BLU}{CTRL RVS OFF}{CTRL RED}{CMDR Q}{9 SHIFT C}{CMDR W}"
840 PRINT"{CTRL YEL}{CMDR A}{9 SHIFT C}{CMDR S}{CTRL RED}{CMDR E}{CTRL YEL} {CTRL RVS ON}{CTRL BLU}{CMDR PUR} {CTRL BLU}{CTRL RVS OFF}{CTRL YEL}{CMDR A}{6 SHIFT C}{CMDR S}"
850 PRINT"{SHIFT B}DOCK TIME{CMDR Z}{SHIFT C}{CMDR S}{CTRL RVS ON}{CTRL BLU} RADAR↑ {CTRL RVS OFF}{CTRL YEL} {SHIFT B}ENERGY{CMDR Z}{4 SHIFT C}{CMDR S}";
860 PRINT"{SHIFT B}{CTRL RVS ON}{CTRL RED} {CTRL RVS OFF}{CTRL YEL}{SHIFT B}{CMDR A}{12 SHIFT C}{CMDR S}{SHIFT B}{CTRL RVS ON}{CTRL RED}{CTRL RVS OFF}{CTRL YEL}{SHIFT B}";
870 PRINT"{SHIFT B}0123456789M{2 SHIFT B}RANGE: {2 SHIFT B}0123456789M{SHIFT B}";
880 PRINT"{CMDR Z}{11 SHIFT C}{CMDR X}{CMDR Z}{12 SHIFT C}{CMDR X}{CMDR Z}{11 SHIFT C}{CMDR X}";:PRINT"{CTRL GRN}SCORE:";
890 REM MAIN CONTROL LOOP
900 POKE V2(0),V2(1)
910 POKE 54279,P*2.5:IF INT(RND(.)*100)<5+M*2 THEN GOSUB1170:GOSUB1240
920 GOSUB1060:IF H<0 THEN H=0
930 IF INT(RND(.)*30)=25 THEN GOSUB1130
940 GOSUB1580:X=7:Y=24:GOSUB1560:PRINT"{CTRL GRN}"H"{SHIFT CRSR LEFT}";
950 E=E-(N/200)-(P/200):IF E<=0 THEN1650
960 X=28:Y=21:GOSUB1560
970 PRINT"{CTRL RVS ON}{CTRL RED}"LEFT$(B$,INT(E/10+.5))"{CTRL RVS OFF}"LEFT$(B$,INT((110-E)/10+.5))
980 D=P/10+.5
990 X=1:Y=11:GOSUB1560:PRINT"{CTRL RVS ON}{CTRL RED}"LEFT$(B$,D)"{CTRL RVS OFF}"LEFT$(B$,9-INT(D))
1000 D=N/10+.5
1010 Y=16:GOSUB1560:PRINT"{CTRL RVS ON}{CTRL RED}"LEFT$(B$,D)"{CTRL RVS OFF}"LEFT$(B$,9-INT(D))
1020 T=T-(P/55)+B:Y=21:GOSUB1560
1030 PRINT"{CTRL RVS ON}{CTRL RED}"LEFT$(B$,INT(T/10+.5))"{CTRL RVS OFF}"LEFT$(B$,INT((116-T)/10+.5))
1040 IF T<0 THEN GOSUB1780
1050 GOTO910
1060 REM SCROLL STARS
1070 FOR I=2 TO 4:POKE 781,I-1:SYS 59888:POKE 172,PEEK(60656+I)
1080 POKE 780,PEEK(217+I):SYS 59848:NEXT:Y=4:X=0:GOSUB1560
1090 PRINT"{CTRL RVS ON}{CTRL BLU} {CTRL RVS OFF}      {CTRL RVS ON} {CTRL RVS OFF}      {CTRL RVS ON}{CTRL RVS OFF}      {CTRL RVS ON} {CTRL RVS OFF}";
1100 FOR I=1 TO 5:X=INT(RND(.)*38+1):IF X=12 OR X=27 THEN1120
1110 GOSUB1560:POKE 646,RND(.)*7+1:PRINT"."
1120 NEXT:PRINT"{CTRL YEL}":RETURN
1130 REM SCROLL THE EARTH
1140 L=ABS(L-1)
1150 FOR D=1 TO 7:X=0:Y=5:GOSUB1560:FOR J=D+2 TO D STEP-1
1160 PRINTES$(L,J);:NEXT:GOSUB1060:NEXT:RETURN
1170 REM RADAR PICKED UP A SHIP
1180 S=INT(RND(.)*8+1)
1190 FOR D=0 TO 42 STEP 21:POKE V1(0),V1(1):X=16:Y=9:GOSUB1560
1200 PRINT"{CTRL RVS ON}{CTRL RED}{8 CMDR *}{CRSR DOWN}{8 SHIFT CRSR LEFT}{8 CMDR *}{CTRL RVS OFF}":GOSUB1060:FOR J=1 TO 75:NEXT
1210 POKE V1(0),V1(1)-1:POKE 198,0
1220 X=16:Y=9:GOSUB1560:PRINT"{CTRL RVS ON}{CTRL WHT}        {CRSR DOWN}{8 SHIFT CRSR LEFT}        {CTRL RVS OFF}"
1230 GOSUB1420:NEXT:RETURN
1240 R=INT(RND(.)*100000)+1000:R$=MID$(STR$(R),2)
1250 X=20:Y=22:GOSUB1560:PRINT"{CTRL YEL}"R$:POKE 53269,PEEK(53269)OR1
1260 X=28:Y=11:GOSUB1560:PRINTSC$(S):Y=16:GOSUB1560:PRINTST$(S)
1270 REM SCAN KEYBOARD TO FIRE - FIRE CANNON
1280 D=0:FOR J=1 TO 10-M:POKE V1(0),V2(1)
1290 GET K$:IF K$<>" " THEN1390
1300 D=1:O=INT(RND(.)*3+1)
1310 POKE 53264,0:POKE G(0),G(O):IF O=3 THEN POKE 53264,30
1320 POKE 53291,5+L*14:POKE 53269,PEEK(53269)OR16:POKE F(0),F(O)
1330 POKE F(0)+2,F(O):POKE 53291,2:POKE V3(0),V3(1)
1340 POKE 53269,PEEK(53269)OR2:POKE 53269,PEEK(53269)OR4:POKE V1(0),V2(1)-1
1350 GOSUB1060:POKE 53291,5+L*14:FOR I=1 TO 200:NEXT
1360 E=E-R/50000:W(S)=W(S)+1:IF S<6 THEN H=H+170-20*S:GOTO1380
1370 H=H-S*45
1380 J=10-M:POKE V3(0),V3(1)-1:POKE 53269,1:IF S=9 THEN J=J+1
1390 POKE V1(0),V2(1)-1:FOR DE=1 TO 50:NEXT:NEXT:IF S<6 AND D=0 THEN1450
1400 X=28:Y=11:GOSUB1560:PRINT"        ":Y=16:GOSUB1560:PRINT"        "
1410 X=20:Y=22:GOSUB1560:PRINT"        ":POKE 53269,0:RETURN
1420 REM STORE PROPER SHIP DATA
1430 O=(S-1)*64
1440 FOR I=0+D TO 21+D:POKE 704+I,PEEK(40192+I+O):NEXT:RETURN
1450 REM THE ALIEN SHOOTS BACK
1460 O=INT(RND(.)*3+1):POKE 53264,0:IF O=3 THEN POKE 53264,14
1470 POKE F(0),F(O):POKE F(0)+2,F(O):POKE F(0)+4,F(O)-10
1480 POKE 53269,PEEK(53269)OR2:FOR I=1 TO 300:NEXT
1490 POKE 53269,PEEK(53269)OR4:GOSUB1060:POKE 53269,PEEK(53269)OR8
1500 FOR I=1 TO 300:NEXT:H=H-M*50:E=E-INT(RND(.)*30)-10:N=N-INT(RND(.)*40)-5
1510 POKE V3(0),V3(1):POKE 53280,2:POKE 53281,2:POKE 53265,PEEK(53265)AND239
1520 FOR I=1 TO 300:NEXT
1530 POKE 53265,PEEK(53265)OR16:POKE 53281,0:POKE 53280,0:POKE 53269,0
1540 POKE V3(0),V3(1)-1:IF N<1 THEN1650
1550 N=INT(N):P=INT(P*.6):GOTO1400
1560 REM PLACE CURSOR AT X,Y
1570 POKE 781,Y:POKE 782,X:POKE 783,0:SYS 65520:RETURN
1580 REM READ KEYBOARD FOR CONTROL OF THRUST AND SHIELDS
1590 GET K$
1600 IF K$="S" THEN N=N+10:IF N>90 THEN N=90
1610 IF K$="{SHIFT S}" THEN N=N-10:IF N<0 THEN N=0
1620 IF K$="T" THEN P=P+10:IF P>90 THEN P=90
1630 IF K$="{SHIFT T}" THEN P=P-10:IF P<0 THEN P=0
1640 RETURN
1650 REM END OF GAME
1660 POKE 53269,0:POKE V2(0),V2(1)-1:PRINT"{SHIFT CLR}":IF H<0 THEN H=0
1670 PRINT"{CTRL YEL}THE ALPHA II FLEET HAS TAKEN OVER THE"
1680 PRINT"{CRSR DOWN}EARTH!! THEY WERE TOO MUCH FOR YOUR"
1690 PRINT"{CRSR DOWN}           EARTH DEFENSES."
1700 PRINT"{3 CRSR DOWN}YOUR FINAL SCORE IS:"H
1710 PRINT"{CRSR DOWN}SHIPS DESTROYED...{CRSR DOWN}"
1720 FOR I=1 TO 9:PRINTW(I),"←",ST$(I),SC$(I):NEXT
1730 FOR I=1 TO 9:W(I)=0:NEXT
1740 PRINT"{2 CRSR DOWN}WOULD YOU LIKE TO PLAY AGAIN?  (Y/N)";
1750 GET K$:IF K$<>"N" AND K$<>"Y" THEN1750
1760 IF K$="Y" THEN L=0:P=90:N=P:T=110:E=T:H=L:GOTO530
1770 PRINT"{SHIFT CLR}BYE":END
1780 REM DOCK WITH THE EARTH BASE
1790 S=9:GOSUB1190:R=50000:R$="50000"
1800 X=20:Y=22:GOSUB1560:PRINT"{CTRL YEL}"R$:POKE 53269,PEEK(53269)OR1
1810 X=28:Y=11:GOSUB1560:PRINTSC$(S):Y=16:GOSUB1560:PRINTST$(S)
1820 FOR J=1 TO 11-M:POKE V1(0),V2(1)
1830 GET K$:IF K$<>" " THEN1850
1840 T=110:B=B+.25:GOTO1300
1850 POKE V1(0),V2(1)-1:GOSUB1060:NEXT:POKE 53269,PEEK(53269)OR32
1860 FOR I=75 TO 92:POKE 53259,I:FOR O=1 TO 50:NEXT:NEXT
1870 IF W(6)+W(7)+W(8)+W(9)<=6 THEN1900
1880 FOR I=1 TO 20:POKE 54296,15:FOR J=1 TO 3:NEXT:POKE 54296,0:FOR O=1 TO 3
1890 NEXT:NEXT:T=110:POKE 54296,15:GOTO1930
1900 H=H+500:E=110:T=E:P=90:N=P
1910 FOR I=1 TO 3:POKE V3(0),V2(1):FOR O=1 TO 200:NEXT:POKE V3(0),V2(1)-1
1920 FOR O=1 TO 200:NEXT:NEXT
1930 POKE 53269,0:POKE 53259,75:GOTO1400
1940 REM DATA AND GRAPHICS FOR THE SHIPS
1950 DATA BASE,ALPHA II,FIGHTER,ALPHA II,FIGHTER,ALPHA II
```

PROGRAM LISTING

Continued

ORBITAL DEFENDER *Continued*


```
1960 DATA FIGHTER,SANDIAN,FIGHTER,SANDIA
N,FIGHTER,EARTH
1970 DATA CARGO,EARTH,PASSENGER,EARTH,BA
SE,EARTH
1980 DATA000,000,000,000,000,000,000,020
1990 DATA000,001,085,064,007,255,208,031
2000 DATA255,244,031,127,244,125,159,253
2010 DATA125,159,253,127,127,253,127,253
2020 DATA253,127,255,253,085,085,085,031
2030 DATA255,244,023,119,116,007,255,208
2040 DATA001,085,064,000,020,000,000,000
2050 DATA000,000,000,000,000,000,000,000
2060 DATA000,000,000,000,000,000,000,000
2070 DATA000,000,000,000,000,000,000,000
2080 DATA000,000,000,000,000,000,000,000
2090 DATA020,000,000,029,000,000,023,064
2100 DATA000,173,213,064,171,127,208,005
2110 DATA085,084,000,000,000,000,000,000
2120 DATA000,000,000,000,000,000,000,000
2130 DATA000,000,000,000,000,000,000,014
2140 DATA000,000,000,000,000,000,000,000
2150 DATA000,000,000,000,000,000,000,000
2160 DATA004,004,000,016,080,000,005,208
2170 DATA000,007,064,085,093,064,005,084
2180 DATA068,106,084,016,001,148,000,004
2190 DATA148,000,000,132,000,000,068,000
2200 DATA000,004,000,000,000,000,000,000
2210 DATA000,000,000,000,000,000,000,000
2220 DATA000,000,000,000,016,000,000,016
2230 DATA000,000,068,000,001,001,000,001
2240 DATA205,000,001,253,000,001,253,000
2250 DATA000,168,000,000,100,000,000,100
2260 DATA000,000,100,000,000,100,000,000
2270 DATA100,000,001,101,000,001,101,000
2280 DATA004,168,064,018,034,016,024,032
2290 DATA144,026,154,144,002,138,000,014
2300 DATA000,000,000,000,000,000,000,016
2310 DATA000,000,016,000,004,016,064,004
2320 DATA084,064,004,084,064,005,117,064
2330 DATA005,253,064,071,255,068,069,085
2340 DATA068,084,084,084,081,017,020,068
2350 DATA084,068,081,001,020,084,000,084
2360 DATA160,000,040,160,000,040,160,000
2370 DATA040,032,000,032,000,000,000,000
2380 DATA000,000,000,000,000,000,000,016
2390 DATA000,000,016,000,000,116,000,000
2400 DATA048,000,000,168,000,000,168,000
2410 DATA000,116,000,000,220,000,000,116
2420 DATA000,001,253,000,001,205,000,037
2430 DATA221,096,037,205,096,033,153,032
2440 DATA001,137,000,000,136,000,000,136
2450 DATA000,000,000,000,000,000,000,000
2460 DATA000,000,000,000,080,000,000,080
2470 DATA000,001,244,000,001,244,000,007
2480 DATA253,000,007,093,000,007,253,000
2490 DATA007,093,000,087,093,080,103,253
2500 DATA144,103,093,144,103,253,144,103
2510 DATA093,144,103,093,144,103,093,144
2520 DATA103,253,144,103,093,144,103,253
2530 DATA144,085,165,080,080,160,080,000
2540 DATA000,000,000,000,000,000,000,000
2550 DATA000,000,000,000,000,000,000,000
2560 DATA000,000,000,000,000,064,000,000
2570 DATA064,000,000,080,000,000,069,084
2580 DATA000,064,001,000,076,204,080,106
2590 DATA170,164,085,085,085,000,080,000
2600 DATA085,084,000,000,000,000,000,000
2610 DATA000,000,000,000,000,000,000,000
2620 DATA000,000,000,000,000,000,000,000
2630 DATA000,000,020,000,001,085,064,005
2640 DATA255,080,021,085,084,090,170,165
2650 DATA085,085,085,000,020,000,000,020
2660 DATA000,021,085,084,000,020,000,000
2670 DATA020,000,085,085,085,090,170,165
2680 DATA023,255,212,005,085,080,005,065
2690 DATA080,004,065,016,016,000,004,255
2700 DATA000,000,000,000,000,000,000,000
2710 DATA000,000,000,000,000,000,000,000
2720 DATA000,000,000,000,000,000,000,000
2730 DATA000,036,000,000,184,000,000,126
2740 DATA000,000,252,000,000,058,000,000
2750 DATA072,000,000,000,000,000,000,000
2760 DATA000,000,000,000,000,000,000,000
2770 DATA000,000,000,000,000,000,000,000
2780 DATA000,000,000,000,000,000,000,000
2790 DATA000,000,000,000,000,000,000,000
2800 DATA064,000,000,134,000,002,184,000
2810 DATA004,254,192,005,255,032,003,255
2820 DATA128,003,255,128,009,255,064,006
2830 DATA254,064,000,058,128,000,194,000
2840 DATA000,004,000,000,000,000,000,000
2850 DATA000,000,000,000,000,000,000,000
2860 DATA000,008,000,000,008,000,000,008
2870 DATA000,000,008,000,000,008,000,000
2880 DATA008,000,000,008,000,000,008,000
2890 DATA000,008,000,000,008,000,000,008
2900 DATA000,000,012,000,000,012,000,000
2910 DATA012,000,000,012,000,000,063,000
2920 DATA000,012,000,000,063,000,000,063
2930 DATA000,000,243,192,015,204,252,016
2940 REM SOUND DATA
2950 DATA 247, 9, 64, 53, 84, 0
2960 DATA 240, 90, 1, 160, 71, 128
2970 DATA 0, 106, 167, 6, 160, 136
2980 DATA 128, 0, 250, 0, 0, 0,15
```

HCM

ORBITAL DEFENDER


```
100 '  ******************
110 '  * Orbital Defender *
120 '  ******************
130 '  COPYRIGHT 1985
140 '  EMERALD VALLEY PUBLISHING CO.
150 '  BY SCOTT WILLIAMS
160 '  AND THE HCM STAFF
170 '  HOME COMPUTER MAGAZINE
180 '  VERSION 5.1.1
190 '  IBM PCjr WITH CARTRIDGE BASIC
       FROM DOS 2.1 or
200 '  IBM PC WITH BASICA and
210 '  COLOR/GRAPHICS ADAPTER and
220 '  COLOR MONITOR
230 '
240 SCREEN 1:COLOR 0,0:KEY OFF:RANDOMIZ
E VAL(MID$(TIME$,7)):ENERGY=110:DOC
KTIME=ENERGY:SCORE=0:THRUST=80:SHIE
LDS=80:RANGE=0:INIT=1:A=INIT:DOCK.D
ELAY=1
250 K=308:DIM TITLE%(334),AFIGHTER1%(K)
,EBASE%(K),ECARGO%(K),AFIGHTER2%(K)
,SFIGHTER1%(K),EFIGHTER%(K),SFIGHTE
R2%(K),ABASE%(K),EPASSENGER%(K),WAR
NING%(709),TEMP%(274),BLAST%(104),B
1%(49),B2%(49),B3%(49),B4%(49)
260 DATA "Alpha II","Base","Alpha II","
Fighter","Alpha II","Fighter","Sand
ian","Fighter","Sandian","Fighter",
"Earth","Fighter","Earth","Cargo","
Earth","Passenger","Earth","Base"
270 REM Define Graphics Data
280 CRAFT$(1)="bm+7,-2frfrfrfr2dl2dglgl
glghlhlhlhul2ur2erererebd3d4bm+2,-1
unubr2r0bl6nundbl2r0bl5u3e4lg4d6f4r
h4u2bm+18,-1u3h4rf4d6g4le4u2bm-9,5f
dfrfrdfd3g3l6h3u3enr9urereuebd6ndbd
3nl5nr5r3nu2l6u2"
290 CRAFT$(2)="r5urdr5br2fgul12r3nd5br2
nd5br2f6nl16uer2f3g3l2hunl16g5dbl3n
u5bl2nu5bl3rr11ufgbl2l5dlyl5bm+16,-
7r0bu2l0bm-19,2du4nh2ld4rg2"
300 CRAFT$(3)="bm+1,-1g4e2f2bfbrgbrbrre
r3f3d3g3l3h3u3elbr8ehbrbee2f2h4bm-5
,7lgdfreul2drubm+2,4rfnlr3e3uhlgdr2
uld2g3lr2bl111l2h3u3rfl3dr3gf2r4u"
310 CRAFT$(4)="bm+7,0f2d2fr3nu2f2ru2d4f
d2l20u2eu4d2re2nu2r3eu2e2bd3fd3ghu3
bm-2,2d3f2r2e2u4bm-1,8l4r2d2nlnrdf2
dl4ue2"
320 CRAFT$(5)="bm+7,0fdghuebm+2,4rfrfdr
nu3lglglgd2f3glh2g2lhe3u2hlhlhlnu3r
uererbm+2,1f2l4e2d"
330 CRAFT$(6)="bm-3,-2frfrf13dglh13uheb
m+1,5d2fd3fd2fd2fnd2eu2e8r2enr2hl2h
l2hl4ul2bm+5,8f4"
340 CRAFT$(7)="bm+18,16a2fdfd4gdgbm+3,-
5r3nd3u3lr11fd4gl12r6d4rnu4nl10r5e2
d6h2l13glhur20d2r2u4l2d2a0"
350 CRAFT$(8)="bd2bl5r3f5r14frf2g2l21u2
huhuhu2bm+10,7r0br2r0br2r0br2r0br2r
0br2r0bm-7,2ng4br2g4r10l16hld4rer16
ebm-16,-7h2ef2g"
360 CRAFT$(9)="bm+5,+1r5dl5r2nd18rd18f4
re2h2lg4lh4lg2f2re4u9nl9nr10dnl9r10
f3u7g3bm-19,0h3d7e3bm+3,-6r2dl2bd8r
2dl2bm+11,-10r2dl2bd8r2dl2"
370 CLS
380 GOSUB 460 : 'Put Up a Banner Screen
390 GOSUB 1220 : 'Get Skill Level
400 GOSUB 670 : 'Cockpit Detail
410 GOSUB 800 : 'Update Cockpit Control
s
420 GOSUB 1030 : 'Generate Horizon
430 PVTHRUST=1:PVENERGY=1:PVSHIELDS=1:P
VDOCK=1:GOSUB 800
440 GOSUB 1320: 'Main Loop
450 END
460 FOR J = 1 TO 9 : READ CLASS$(J),TYP
E$(J) : NEXT J
470 FOR K = 1 TO 4 :LOCATE 1,(K-1)*10+1
 : PRINT CLASS$(K): LOCATE 2,(K-1)*
10+1:PRINT TYPE$(K) : PRESET ((K-1)
*80+18,30) : DRAW "c1"+CRAFT$(K) :
LINE ((K-1)*80+10,20)-((K-1)*80+41,
60),3,B : NEXT K
480 FOR K = 5 TO 8 :LOCATE 10,(K-5)*10+
1 : PRINT CLASS$(K): LOCATE 11,(K-5
)*10+1:PRINT TYPE$(K) : PRESET ((K-
5)*80+18,100) : DRAW "c1"+CRAFT$(K)
: LINE ((K-5)*80+10,90)-((K-5)*80+
41,130),3,B : NEXT K
490 LOCATE 19,1 : PRINT CLASS$(9) : LOC
ATE 20,1 : PRINT TYPE$(9) : PRESET
(18,170) : DRAW "c1"+CRAFT$(9) : LI
NE (10,165)-(41,199),3,B
```

Continued

Continued

ORBITAL DEFENDER *Continued*


```
1260 KS=INKEYS:IF KS="" THEN 1260
     ENDIF THRUST=THRUST-10*(THRUST<800):GOTO
1270 IF THRUST=CHR$(80)+CHR$(80) THEN THRUST=
     THRUST+10*(THRUST>0):GOTO 1280
1280 IF KS=CHR$(80)+CHR$(77) THEN SHIELDS=
     SHIELDS+10*(SHIELDS<80):GOSUB 1800
1290 IF KS=CHR$(75) THEN SHIELDS=SHIELDS-
     10*(SHIELDS>0):GOSUB 1800
1300 PEEK(1052):RETURN
1310 'Plot Score
1320 IF SCORE < 0 THEN SCORE = 0
1330 ENERGY = ENERGY - (THRUST/400)
1340 IF ENERGY <= 0 AND TIME < 1740 THEN
     GOSUB 1480
1350 TIME = TIME + 1
1360 IF TIME=2500 THEN GOSUB 1110
1370 IF TIME<5+LEVEL*200 THEN GOTO 1360
1380 LEVEL=LEVEL+1
1390 'Supply Ship
1400 'Picked Up Ship
1410 FOR K=1 TO 50: NEXT K: SHIP=SHIP+1:
     GOTO 1360
1420 'Ship Destroyed
1430 IF SHIP>0 THEN SCORE=SCORE-450:
     SHIP=SHIP-1: GOTO 1360
1440 'Energy Ran Out
1450 'Alien Ship Destroyed
1460 IF SHIP=0 THEN 1360
1470 'Level Up
1480 'Ship Destroyed
1490 'Alien Ship Destroyed
1500 'Plot Ship
1510 'Erase Ship
1520 'Sound When Destroying Earth Ship
1530 'Sound When Destroying Alien Ship
1540 'Play Last
1550 'Play Blast
1560 'Play Blast
1570 'Play Blast
1580 'Play Blast
1590 'Play Blast
1600 'Play Blast
1610 'Play Blast
1620 'Play Blast
1630 'Play Blast
1640 'Play Blast
1650 'Play Blast
1660 'Play Blast
1670 'Play Blast
1680 'Play Blast
1690 'Play Blast
1700 'Play Blast
1710 'Play Blast
1720 'Play Blast
1730 'Play Blast
1740 'Play Blast
1750 'Play Blast
1760 'Play Blast
1770 'Play Blast
1780 'Play Blast
1790 'Play Blast
1800 'Play Blast
1810 'Play Blast
1820 'Play Blast
```

HCM

ORBITAL DEFENDER


```
100  REM ***********DEFENDER*********
110  REM ***ORBITAL DEFENDER*****
120  REM ***********************
130  REM **COPYRIGHT 1984, 1985
140  REM COMPUTER PUBLISHING CO.
150  REM BY THOMAS WILLIAMS
160  REM HOME COMPUTER MAGAZINE
170  REM VERSION 5.1.1
180  REM VT BASIC OR EXTENDED BASIC
190  REM *********************
200  CALL CLEAR
210  PRINT :::"ORBITAL DEFENDER":::::
     PRESS ENTER TO CONTINUE"
220  GOSUB 1200
230  CALL CLEAR
240  PRINT "KEY BOARD OR JOYSTICK
     (K/J)";
250  CALL KEY(0,K,S)
260  IF S=0 THEN 250
270  IF K<>74 AND K<>75 THEN 250
280  J=0
290  IF K<>74 THEN 320
300  J=1
310  CALL CLEAR
320  RANDOMIZE
330  DIM W(9)
340  READ E,T,P,N,H
350  DATA 110,110,90,90,0
360  RESTORE 3400
370  GOSUB 3120
380  GOSUB 3120
390  IF (K<48)+(K>57) THEN 390
400  M=K-48
410  CALL CLEAR
420  CALL SCREEN(5)
430  CALL COLOR(Z,5,5)
440  FOR Z=1 TO 14
450  CALL COLOR(Z,2,2)
460  NEXT Z
470  RESTORE 3430
480  FOR Z=1 TO 19
490  READ CHAR(A,A$)
500  CALL CHAR(A,A$)
510  NEXT Z
520  RESTORE 3530
530  RESTORE 3150
540  FOR Z=1 TO 14
550  READ CHAR(A,B,C,D)
560  CALL CHAR(A,B,C,D)
570  NEXT Z
580  FOR Z=1 TO 9
590  READ CHAR(A,B,C,D)
600  CALL CHAR(A,B,C,D)
610  NEXT Z
620  CALL COLOR(13,2,2)
630  FOR Z=1 TO 12
640  READ COLOR(Z,A,B)
650  CALL COLOR(Z,A,B)
660  NEXT Z
670  FOR Z=1 TO 30
680  IF INT(RND*6+5)=INT(RND*26+4)
     THEN 690 ELSE CALL CHAR(X,Y,G)
690  IF G<>HCHAR(X,Y,35) THEN 690
700  CALL CHAR(X,Y,35)
710  NEXT Z
720  RESTORE 3200
730  FOR Z=1 TO 32
740  READ B
750  IF B=0 THEN 770
760  GOSUB 3200
770  IF INT(RND*100)>4+M*2 THEN 800
780  GOSUB 970
790  GOSUB 1060
800  IF HCHAR>=0 THEN 820
810  H=0
820  GOSUB 1980
830  E=E-(N/200)-(P/200)
```

Continued

```
840 IF E<=0 THEN 2810
850 CALL HCHAR(22,20,104,E/10)
860 CALL HCHAR(22,20+INT(E/10+.5),112,11-INT(E/10+.5))
870 T=T-(P/90)+B
880 CALL HCHAR(22,3,104,T/10)
890 CALL HCHAR(22,3+INT(T/10+.5),112,11-INT(T/10+.5))
900 IF T>0 THEN 920
910 GOSUB 2260
920 CALL HCHAR(14,3,104,P/10)
930 CALL HCHAR(14,3+INT(P/10+.5),112,9-INT(P/10+.5))
940 CALL HCHAR(18,3,104,N/10)
950 CALL HCHAR(18,3+INT(N/10+.5),112,9-INT(N/10+.5))
960 GOTO 760
970 FOR Z=1 TO 3
980 CALL COLOR(9,7,5)
990 CALL SOUND(-200,350,3,440,3,480,3,-2,0)
1000 CALL COLOR(9,3,5)
1010 FOR A=1 TO 10
1020 NEXT A
1030 GOSUB 3200
1040 NEXT Z
1050 RETURN
1060 A=INT(RND*9)+1
1070 IF A=6 THEN 1060
1080 ON A GOSUB 1800,1820,1840,1860,1880,1900,1920,1940,1960
1090 GOSUB 3200
1100 R=INT(RND*10000+1000)
1110 READ A$,B$,C$,D$,E$,F$
1120 CALL CHAR(128,A$)
1130 CALL CHAR(129,B$)
1140 CALL CHAR(130,C$)
1150 CALL CHAR(131,D$)
1160 CALL COLOR(13,15,2)
1170 FOR Z=1 TO 10-M
1180 CALL KEY(1,K1,S1)
1190 CALL KEY(0,K,S)
1200 CALL SOUND(-1,660,0)
1210 IF ((S=1)*(K=32))+((S1=1)*(K1=18))THEN 1280
1220 NEXT Z
1230 GOSUB 3200
1240 GOSUB 3220
1250 IF A<6 THEN 1530
1260 CALL COLOR(13,2,2)
1270 RETURN
1280 D=INT(RND*3+1)
1290 D=D*10-4
1300 CALL COLOR(12,10,2)
1310 FOR Z=3 TO 0 STEP -1
1320 CALL SOUND(-100,110,30,110,30,5000*Z+5000,30,-4,0)
1330 CALL HCHAR(7+Z,D,120+Z)
1340 NEXT Z
1350 CALL SOUND(-500,110,3,110,30,3000,30,-8,0)
1360 GOSUB 3200
1370 GOSUB 3220
1380 E=E-R/50000
1390 W(A)=W(A)+1
1400 CALL VCHAR(7,D,33,4)
1410 IF A<6 THEN 1490
1420 H=H-A*45
1430 IF H>=0 THEN 1450
1440 H=0
1450 CALL COLOR(13,2,2)
1460 CALL SOUND(-1000,110,0)
1470 GOSUB 3050
1480 RETURN
1490 H=H+170-20*A
1500 CALL COLOR(13,2,2)
1510 GOSUB 3050
1520 RETURN
1530 RESTORE 3470
1540 D=INT(RND*3+1)
1550 D=D*10-4
1560 CALL CHAR(136,"00")
1570 CALL HCHAR(7,D,136)
1580 CALL COLOR(14,7,2)
1590 FOR Z=1 TO 6
1600 READ A$
1610 CALL CHAR(136,A$)
1620 CALL SOUND(450,110,30,110,30,6500-(Z*1000),30,-8,30-Z*5)
1630 NEXT Z
1640 CALL HCHAR(7,D,33)
1650 CALL COLOR(13,2,2)
1660 FOR Z=1 TO 10
1670 CALL SOUND(-100,-6,0)
1680 CALL SCREEN(7)
1690 CALL COLOR(1,10,10)
1700 CALL SOUND(-100,-7,0)
1710 CALL SCREEN(5)
1720 CALL COLOR(1,2,8)
1730 NEXT Z
1740 H=H-M*50
1750 E=E-INT(RND*30)-10
1760 N=N-INT(RND*40)-5
1770 IF N<1 THEN 2810
1780 P=INT(P*.6)
1790 RETURN
1800 RESTORE 3590
1810 RETURN
1820 RESTORE 3600
1830 RETURN
1840 RESTORE 3610
1850 RETURN
1860 RESTORE 3620
1870 RETURN
1880 RESTORE 3630
1890 RETURN
1900 RESTORE 3640
1910 RETURN
1920 RESTORE 3650
1930 RETURN
1940 RESTORE 3660
1950 RETURN
1960 RESTORE 3670
1970 RETURN
1980 IF J=0 THEN 2110
1990 CALL JOYST(1,X,Y)
2000 N=N+Y*2.5
2010 P=P+X*2.5
2020 IF P<=90 THEN 2040
2030 P=90
2040 IF P>=0 THEN 2060
2050 P=0
2060 IF N<=90 THEN 2080
2070 N=90
2080 IF N>=0 THEN 2100
2090 N=0
2100 RETURN
2110 CALL KEY(0,K,S)
2120 IF S=0 THEN 2250
2130 IF K<>69 THEN 2160
2140 P=P+10
2150 GOTO 2020
2160 IF K<>88 THEN 2190
2170 P=P-10
2180 GOTO 2040
2190 IF K<>68 THEN 2220
2200 N=N+10
2210 GOTO 2060
2220 IF K<>83 THEN 2250
2230 N=N-10
2240 GOTO 2080
2250 RETURN
2260 RESTORE 3640
2270 GOSUB 970
2280 CALL COLOR(13,2,2)
2290 FOR Z=0 TO 3
2300 READ A$
2310 CALL CHAR(128+Z,A$)
2320 NEXT Z
2330 READ E$,F$
2340 CALL COLOR(13,15,2)
2350 FOR Z=1 TO 10-M
2360 CALL KEY(1,K1,S1)
2370 CALL KEY(0,K,S)
2380 CALL SOUND(1,660,0)
2390 IF ((S=1)*(K=32))+((S1=1)*(K1=18))THEN 2770
2400 IF W(6)+W(7)+W(8)+W(9)<=6 THEN 2410
2410 NEXT Z
2420 IF W(6)+W(7)+W(8)+W(9)<=6 THEN 2490
2430 FOR Z=1 TO 3
2440 FOR ZZ=1600 TO 800 STEP -25
2450 CALL SOUND(-100,ZZ,0)
2460 NEXT ZZ
2470 NEXT Z
2480 GOTO 2710
2490 CALL HCHAR(5,16,128)
2500 CALL HCHAR(6,16,129)
2510 CALL HCHAR(5,17,130)
2520 CALL HCHAR(6,17,131)
2530 FOR Z=330 TO 800 STEP 10
2540 CALL SOUND(-100,Z,0)
2550 NEXT Z
2560 GOSUB 3200
2570 GOSUB 3220
2580 H=H+500
2590 GOSUB 3050
2600 E=110
2610 P=90
2620 N=90
2630 T=110
2640 FOR Z=1 TO 20
2650 CALL SOUND(1,1200,0)
2660 NEXT Z
2670 CALL HCHAR(5,16,33,2)
2680 CALL HCHAR(6,16,33,2)
2690 CALL COLOR(13,2,2)
2700 RETURN
2710 T=110
2720 FOR Z=1 TO 20
2730 CALL SOUND(-200,-2,Z)
2740 NEXT Z
2750 CALL COLOR(13,2,2)
2760 RETURN
2770 T=110
2780 H=H-230
2790 A=6
2800 GOTO 1280
2810 CALL CLEAR
2820 CALL SCREEN(8)
2830 PRINT "THE ALPHA II FLEET HAS TAKEN OVER THE EARTH":"THEY WERE TOO MUCH FOR YOUR DEFENSES"::
2840 IF H>0 THEN 2860
2850 H=0
```

Continued

PROGRAM LISTING

ORBITAL DEFENDER *Continued*

TI-99/4A

```
2860 PRINT "YOUR FINAL SCORE IS:";H::"SH
     IPS DESTROYED:"
2870 RESTORE 3590
2880 FOR Z=1 TO 9
2890 READ A$,B$,C$,D$,E$,F$
2900 PRINT F$;TAB(10);E$;TAB(22);W(Z)
2910 NEXT Z
2920 PRINT ::"PLAY AGAIN   Y   N"
2930 CALL KEY(0,K,S)
2940 IF S=0 THEN 2930
2950 IF K=78 THEN 3030
2960 IF K<>89 THEN 2930
2970 FOR Z=1 TO 9
2980 W(Z)=0
2990 NEXT Z
3000 RESTORE 360
3010 CALL CLEAR
3020 GOTO 350
3030 CALL CLEAR
3040 END
3050 CALL HCHAR(1,17,32,5)
3060 A$=STR$(H)
3070 X=1
3080 Y=17
3090 GOSUB 3360
3100 RETURN
3110 GOTO 3110
3120 CALL KEY(0,K,S)
3130 IF S=0 THEN 3120
3140 RETURN
3150 FOR Z=1 TO 23
3160 READ A$
3170 PRINT A$
3180 NEXT Z
3190 RETURN
3200 CALL SOUND(-4250,110,30-P/4,220,30-
     P/4,3500,30,-8,30-P/4)
3210 RETURN
3220 X=14
3230 Y=22
3240 A$=E$
3250 CALL HCHAR(14,22,32,9)
3260 GOSUB 3360
3270 X=17
3280 A$=F$
3290 CALL HCHAR(17,22,32,9)
3300 GOSUB 3360
3310 X=20
3320 A$=STR$(R)
3330 CALL HCHAR(20,22,32,9)
3340 GOSUB 3360
3350 RETURN
3360 FOR Z=0 TO LEN(A$)-1
3370 CALL HCHAR(X,Y+Z,ASC(SEG$(A$,Z+1,1)
     ))
3380 NEXT Z
3390 RETURN
3400 DATA "        SKILL LEVEL",,,0 ROOK
     IE CADET,1 SPACE COOK'S ASSISTANT,2
     SPACE COOK,3 RADIO OPERATOR
3410 DATA 4 GUNNERY OFFICER,5 NAVIGATOR'
     S ASSISTANT,6 NAVIGATOR,7 HELMSMAN,
     8 CAPTAIN
3420 DATA 9 COMMANDER,,,,PLACE ALPHA LOC
     K DOWN,,,,,YOUR CHOICE?
3430 DATA 33,FFFFFFFFFFFFFFFF,40,010103
     03070F0F,41,1F1F3F3F7F7FFFFF,42,F0
     F0E0E0C0C08080,43,FFFFFEFEFCFCF8F8
3440 DATA 44,FFFF7F7F3F3F1F1F,45,0F0F070
     703030101,46,8080C0C0E0E0F0F0,47,F8
     F8FCFCFEFEFFFF,96,0000FFFFFFFFFFFF
3450 DATA 97,FFFFFFFFFFFFFFFF,104,FF0000
     00000000FF,107,FFFFFFFFFFFFFFFF,35,
     FFFFFEFFFFFFFFFF,61,008894A2A2A2948
     8
3460 DATA 120,0808080808080808,121,1C1C1
     C1C1C1C1C1C,122,3E3E3E3E3E3E3E3E,12
     3,7F7F7F7F7F7F7F7F
3470 DATA 00000010,00000018,0000001818,0
     000383838,00183C7E7E3C18,3C7EFFFFFF
     FF7E3C
3480 DATA 2,3,107,28,11,3,107,12,11,19,1
     07,12,14,3,104,9,18,3,104,9,22,3,10
     4,11,22,20,104,11
3490 DATA 12,12,107,3,12,19,107,3,11,15,
     96,4,12,15,97,4,21,12,107,10,1,1,10
     7,10,1,23,107,10
3500 DATA 3,3,107,9,2,30,107,9,13,12,107
     ,8,13,21,107,8,1,31,107,96,16,16,12
     8,1,17,16,129,1,16,17,130,1
3510 DATA 17,17,131,1
3520 DATA 2,8,2,5,2,8,2,8,2,8,2,8,2,8,2,
     8,3,5,5,7,2,2,8,3
3530 DATA "        SCORE:",,"!!!!!!!!!*)
     !!!!!!/-!!!!!!!!",",!!!!!!!!+(!!!!!!
     !!.,!!!!!!!"
3540 DATA "!!!!!!!!*)!!!!!!!!/-!!!!!!!",
     "!!!!!!!+(!!!!!!!!!!.,!!!!!!",","!!!!
     !!!*)!!!!!!!!!!/-!!!!!!"
3550 DATA "!!!!!!!k!!!!!!!!!!!!k!!!!!!",
     "!!!!!!!k!!!!!!!!!!!!k!!!!!!",","!!!!
     !!!k!!!!!!!!!!!!k!!!!!!"
3560 DATA ,,"THRUST:   !!!!!!!! CLASS:",
     "          !!!!!!!!",","012345678 !!!
     !!!!!",","          !!!!!!!! TYPE:"
3570 DATA "SHIELDS:  !!!!!!!!",","
       !!!!!!!!!",","012345678 !!!!!!! RAN
     GE:",","          !!!!!!!!"
3580 DATA "DOCK TIME:         ENERGY:",,
     "0123456789=      0123456789="
3590 DATA 438EB1FFB18E4301,0205083F44955
     13F,C2718DFF8D71C280,40A010FC465512
     FC,BASE,ALPHA II
3600 DATA 081020508C0A0502,070E1C0EC7C12
     21C,100804 0A3150A040,E07038 70E38344
     38,FIGHTER A,ALPHA II
3610 DATA 00317F292824A2FF,E0FFA2242829 7
     F31,0080C08000038C41A,251AC438 0080C0
     80,FIGHTER B,ALPHA II
3620 DATA 01010302828499E3,B01E02020C112
     23C,000080808242328E,1AF080806010 88
     78,FIGHTER A,SANDIAN
3630 DATA 0204F884432C30C0,412214083412 0
     100,21468EFC08294A8C,10112A2444649 4
     8C,FIGHTER B,SANDIAN
3640 DATA 839DE1FFFFE19D83,010131 4A8484 4
     A31,C1B987FFFF87B9C1,80808C522121 52
     8C,BASE,EARTH
3650 DATA 8040302916090814,121111503013 1
     408,001008FE01028464,144C249444FC03
     02,FIGHTER,EARTH
3660 DATA 20333F33200106 1F,3F4040808040 4
     03F,F0FCFEFC609008FC,FD0202040402 02
     FD,CARGO,EARTH
3670 DATA 0060584 4F809580,807F1830BFE0B
     F,00000000FC065101,0EF03060F80EF8,P
     ASSENGER,EARTH
```

HCM

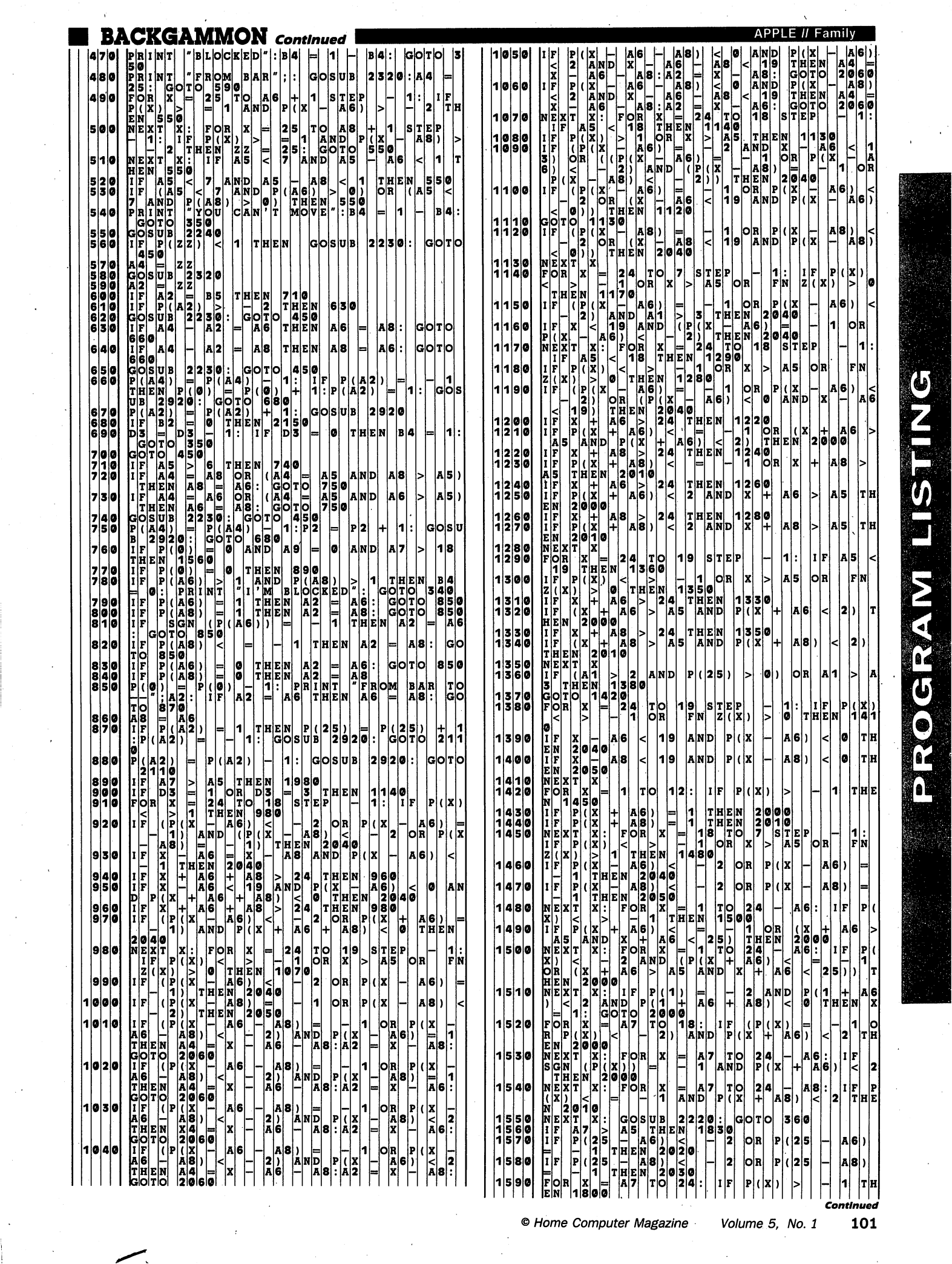
BACKGAMMON

APPLE // Family

```
100 REM    ***************
110 REM    *  BACKGAMMON  *
120 REM    ***************
130 REM    COPYRIGHT 1984,1985
140 REM    EMERALD VALLEY PUBLISHING CO.
150 REM    BY DENNIS WEBBER & THE HCM STA
    FF
160 REM    HOME COMPUTER MAGAZINE
170 REM    VERSION 5.1.1
180 REM    APPLE II FAMILY APPLESOFT
190 REM
200 IF PEEK (103) = 1 AND PEEK (104)
    = 64 THEN 230
210 POKE 103,1: POKE 104,64: POKE 16384
    ,0: POKE 16385,0: POKE 16386,0
220 PRINT CHR$ (4);"RUN GAMMON"
230 DEF FN Z(X) = (P(X + A6) > 0) + (P
    (X + A8) > 0)
240 TEXT : HOME : VTAB 12: HTAB 15: PRI
    NT "BACKGAMMON": VTAB 23: HTAB 15:
    PRINT "PLEASE WAIT"
250 DEF FN D(A) = INT ( RND (1) * 6)
    + 1
260 GOSUB 2400
270 GOSUB 2920
280 INPUT "NAME ?";NA$: PRINT
290 IF LEN (NA$) > 8 THEN PRINT "8 CH
    ARACTERS MAXIMUM": GOTO 280
300 D1 = FN D(0):D2 = FN D(0): PRINT
    NA$;" ROLLED ";D1: PRINT "I ROLLED
    ";D2: FOR TD = 1 TO 2000: NEXT
310 IF D1 < D2 THEN PRINT "I GO FIRST"
    :B4 = 1: FOR TD = 1 TO 2000: NEXT :
      GOTO 410
320 IF D2 < D1 THEN PRINT "YOU GO FIRS
    T":B4 = 0: FOR TD = 1 TO 2000: NEXT
      : GOTO 410
330 GOTO 300
340 PRINT "YOUR TURN ";NA$:A6 = D1:A8 =
      D2:B4 = 0: FOR TD = 1 TO 500: NEXT
      : GOTO 360
350 PRINT "MY TURN":A6 = D1:A8 = D2:B4
    = 1: FOR TD = 1 TO 2000: NEXT
360 IF B4 = 0 THEN C$ = NA$: GOTO 380
370 GOTO 400
380 PRINT "ROLL ";C$
390 IF A13 THEN GET A$
400 D1 = FN D(0):D2 = FN D(0): IF D1
    < D2 THEN A8 = D1:A6 = D2: GOTO 420
410 A6 = D1:A8 = D2
420 GOSUB 3270
430 D3 = 2: IF A6 = A8 THEN D3 = 4
440 ON B4 + 1 GOTO 450,760
450 IF P(25) = 0 THEN 490
460 IF P(25 - A6) > - 2 OR P(25 - A8)
    > - 2 THEN 480
```

Continued

BACKGAMMON *Continued*



PROGRAM LISTING

Continued

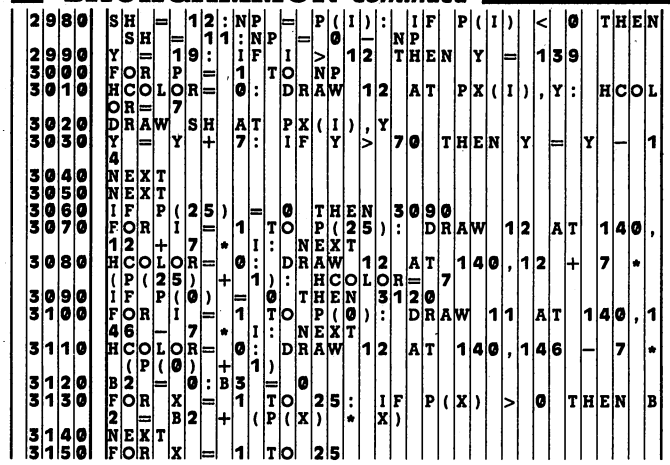
PROGRAM LISTING

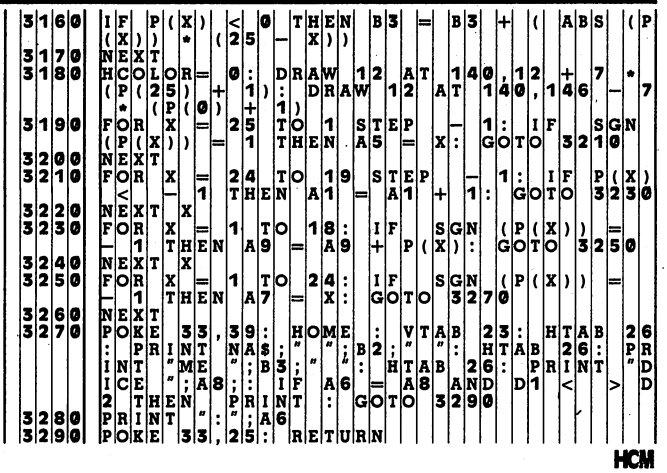
```
1600
1610
1620
1630
1640
1650
1660
1670
1680
1690
1700
1710
1720
1730
1740
1750
1760
1770
1780
1790
1800
1810
1820
1830
1840
1850
1860
1870
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000
2010
2020
2030
2040
2050
2060
2070
2080
2090
2100
2110
2120
2130
2140
2150
2160
2170
2180
2190
2200
2210
2220
2230
2240
2250
2260
2270
2280
2290
2300
2310
2320
2330
2340
2350
2360
2370
2380
2390
2400
2410
2420
2430
2440
2450
2460
2470
2480
2490
2500
2510
2520
2530
2540
2550
2560
2570
2580
2590
2600
2610
2620
2630
2640
2650
2660
2670
2680
2690
2700
2710
2720
2730
2740
2750
2760
2770
2780
2790
2800
2810
2820
2830
2840
2850
2860
2870
2880
2890
2900
2910
2920
2930
2940
2950
2960
2970
```

Continued

BACKGAMMON *Continued*

APPLE // Family

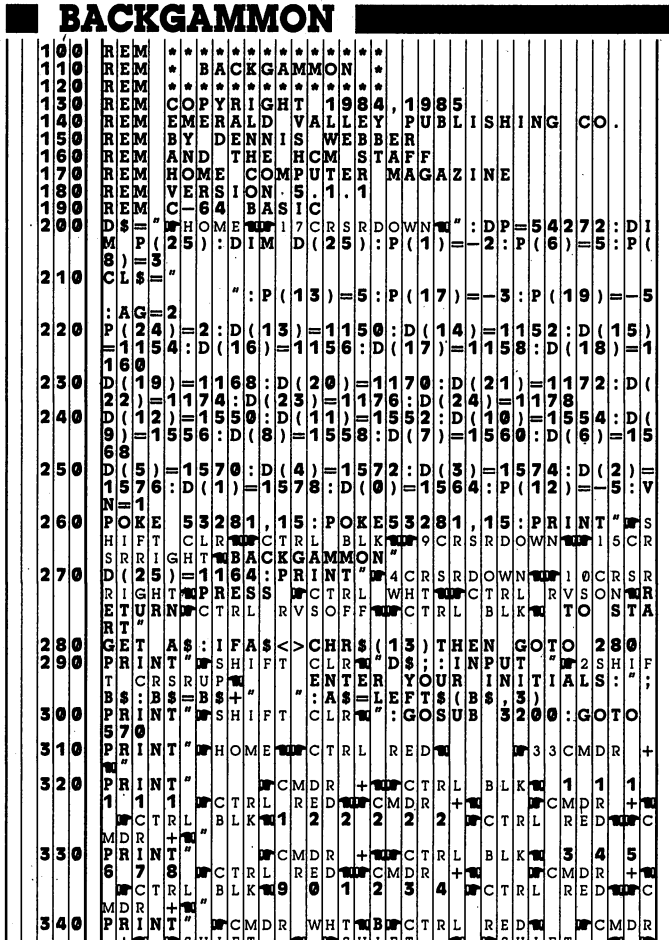


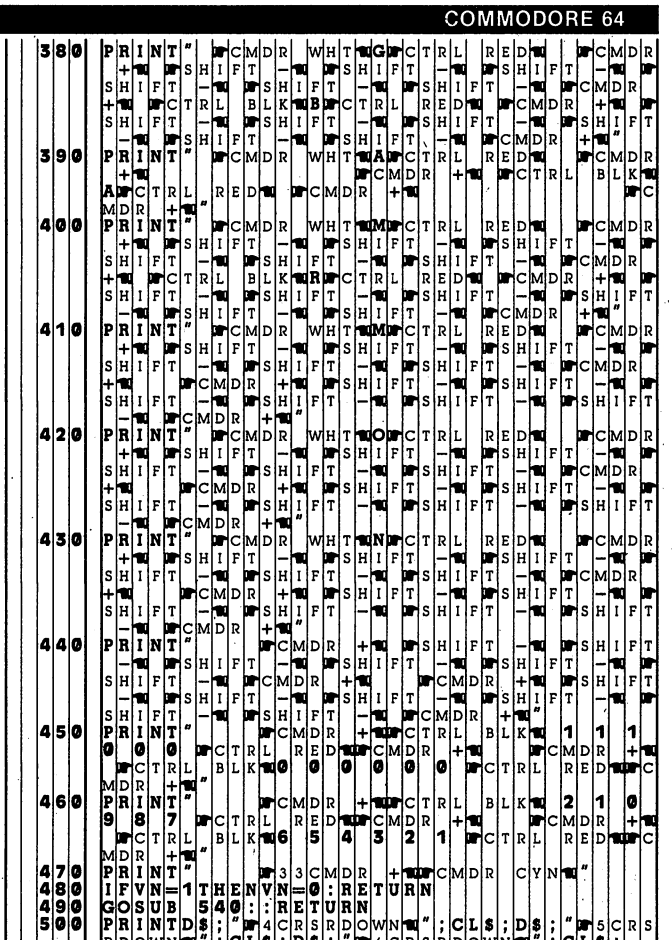


HCM

BACKGAMMON

COMMODORE 64





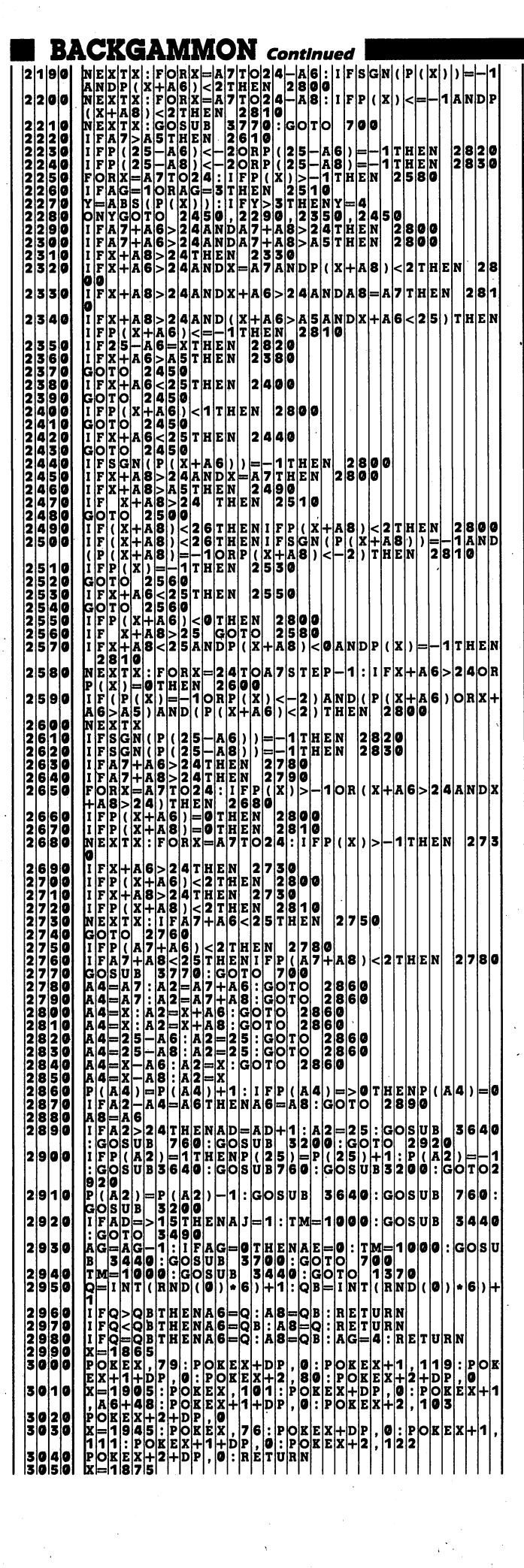
Continued

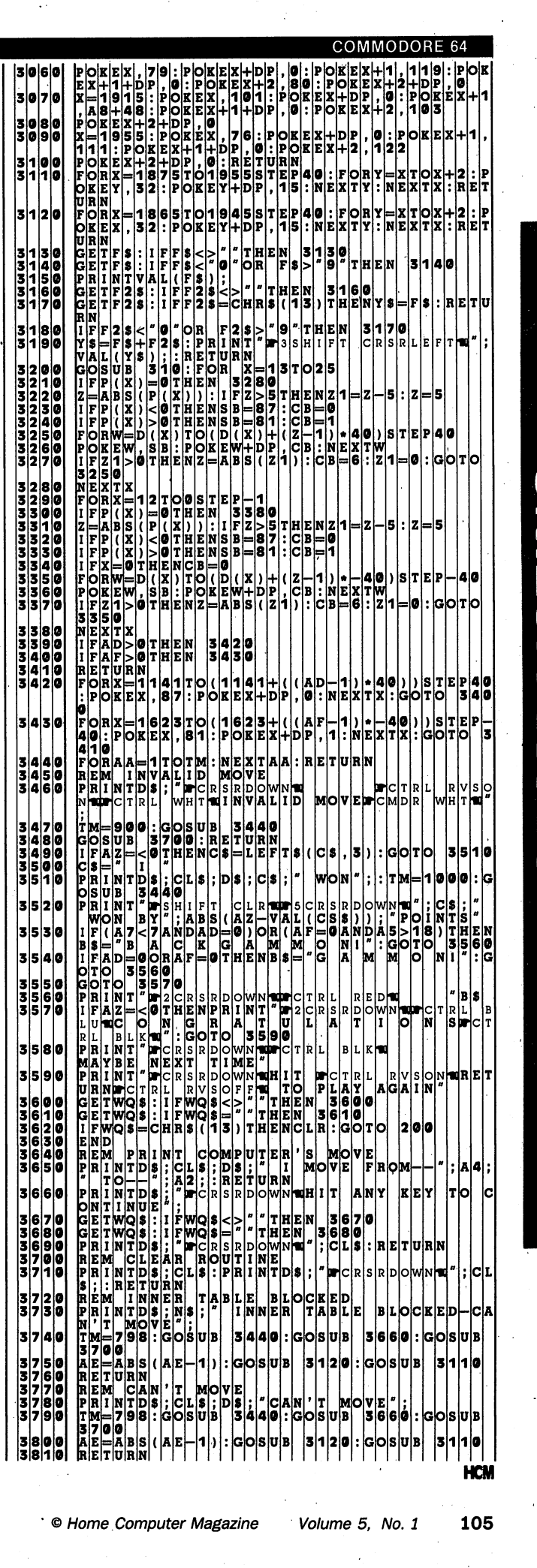
PROGRAM LISTING

BACKGAMMON *Continued*


```
 610  REM PRINTD$;"■4CRSRDOWN■";CL$;D$;"■
      6CRSRDOWN■";CL$;
 620  IFQ<>QBTHEN 660
 630  PRINT"";:PRINTD$;"  DOUBLES - NO ONE W
      INS!";:PRINTD$;:"■CRSRDOWN■DICE WILL BE RE-
      ROLLED";
 640  TM=10000:GOSUB 3440::GOSUB 3700::GOT
      O 580
 660  IFQ>QBTHENPRINTD$"I'M FIRST";:TM=15
      00:GOSUB3440:GOSUB3660:GOSUB500:GOT
      O360
 670  PRINTD$A$" GOES FIRST";:TM=1500:GOS
      UB 3440:GOSUB 3660:GOSUB 500:GOTO 9
      60
 680  PRINTD$;:PRINT" "A$" WILL MOVE FROM
      ";:GOSUB 3130:RETURN
 690  PRINT" TO";:GOSUB 3130::RETURN
 700  IFA=0THENC$=A$+"'S":GOTO 720
 710  C$="MY"
 720  GOSUB"  ":3120:GOSUB 3700:PRINTD$;C$;"
      ROLL";:AG=2
 730  TM=10000:GOSUB 3440
 740  IFAE<>0THEN 750
 750  ONAE+1GOTO 960,1360
 760  AZ=0:AC=0
 770  FORX=1TO25
 780  IFP(X)>0THENAZ=AZ+(P(X)*X):GOTO 800
 790  IFP(X)<0THENAC=AC+ABS(P(X))*(25-X)
 800  NEXTX
 810  GOSUB 540
 820  A7=0:A9=0:A1=0:A5=0
 830  FORX=25TO1STEP-1
 840  IFSGN(P(X))=1THENA5=X:GOTO 860
 850  NEXTX
 860  FORX=24TO19STEP-1
 870  IFP(X)<-1THENA1=A1+1:GOTO 890
 880  NEXTX
 890  FORX=1TO18
 900  IFSGN(P(X))=-1THENA9=A9+P(X):GOTO 9
      20
 910  NEXTX
 920  FORX=1TO24
 930  IFSGN(P(X))=-1THENA7=X:RETURN
 940  NEXTX
 950  RETURN
 960  GOSUB 2950
 970  GOSUB 2990:IFAG>1THENGOSUB 3050
 980  IFAG<>1THENGOSUB 3110
 990  IFP(25)=0THEN 1030
1000  IF(P(25)=-1OR(P(25)=-2)THEN 1090
1010  N$="BLACK":GOSUB 3720:GOTO 700
1020  PRINT";:A4=25:GOSUB3130:A2=VAL(Y$):GOTO
      1140
1030  FORX=25TOA6+1STEP-1:IFP(X)>=1ANDP(X
      -A6)>-2THEN 1090
1040  NEXTX:FORX=25TOA8+1STEP-1:IFP(X)>=1
      ANDP(X-A8)>-2THEN 1090
1050  NEXTX:IFA5<7ANDA5-A6<1THEN 1090
1060  IFA5<7ANDA5-A8<1THEN 1090
1070  IF(A5<7ANDP(A6)>0)OR(A5<7ANDP(A8)>0
      )THEN 1090
1080  GOSUB 3770:GOTO 700
1090  GOSUB 3700:GOSUB 6800:A4=VAL(Y$)
1100  IFA4<10RA4>24THENGOSUB 3450:GOTO 10
      90
1110  IFP(A4)<1THENGOSUB 3450:GOTO 1090
1120  GOSUB 690:A2=VAL(Y$)
1130  IFA2=0THEN 1310
1140  IFA2<10RA2>24THEN 1160
1150  IFP(A2)>-2THEN 1190
1160  GOSUB 3450
1170  IFP(25)>0THEN 1020
1180  GOTO 1090
1190  IFA4-A2=A6THENA6=A8:GOTO 1240
1200  IFA4-A2=A8THENA8=A6:GOTO 1240
1210  GOSUB 3450
1220  IFP(25)>0THENGOTO 1020
1230  GOTO 1090
1240  P(A4)=P(A4)-1
1250  IFP(A2)=-1THENP(0)=P(0)+1:P(A2)=1:G
      OSUB 760::GOSUB 3200::GOTO 1270
1260  P(A2)=P(A2)+1:GOSUB 760:GOSUB 3200
1270  IFAF=>15THENAJ=2:GOTO 3490
1280  AG=AG-1:IFAG=0THENAE=1:GOTO 700
1290  GOSUB 760
1300  GOTO 970
1310  IFA5>6THEN 1340
1320  IFA4=A8OR(A4=A5ANDA8>A5)THENA8=A6:G
      OTO 1350
1330  IFA4=A6OR(A4=A5ANDA6>A5)THENA6=A8:G
      OTO 1350
1340  GOSUB 3450:GOTO 1090
1350  P(A4)=P(A4)-1:AF=AF+1:GOSUB 760:GOS
      UB 3200:GOTO 1270
1360  GOSUB 2950
1370  GOSUB 2990:IFAG>1THENGOSUB 3050
1380  IFAG<>1THENGOSUB 3110
1390  IFP(0)=0ANDA9=0ANDA7>18THEN 2220
1400  IFP(0)=0THEN 1540
1410  IFP(A6)>1ANDP(A8)>1THENN$="WHITE":G
      OSUB 3720:GOTO 700
1420  IFP(A6)=1THENA2=A6:GOTO 1480
1430  IFP(A8)=1THENA2=A8:GOTO 1480
1440  IFSGN(P(A6))=-1THENA2=A6:GOTO 1480
1450  IFP(A8)<=-1THENA2=A8:GOTO 1480
1460  IFP(A6)=0THENA2=A6:GOTO 1480
1470  IFP(A8)=0THENA2=A8
1480  P(0)=P(0)-1
1490  PRINTD$;CL$;:PRINTD$;"I MOVE FROM B
      AR TO";A2;
1500  IFA2=A6THENA6=A8:GOTO 1520
1510  A8=A6
1520  IFP(A2)=1THENP(25)=P(25)+1:P(A2)=-1
      :GOSUB 760:GOSUB 3200::GOTO 2920
1530  P(A2)=P(A2)-1:GOSUB 760:GOSUB 3200:
      GOTO 2920
1540  IFA7>A5THEN 2780
1550  IFAG=1ORAG=3THEN 1800
1560  FORX=24TO18STEP-1:IFP(X)<>1THEN 163
      0
1570  IF(P(X-A6)<-2ORP(X-A6)=-1)AND(P(X-A
      8)<-2ORP(X-A8)=-1)THEN 2840
1580  IFX-A6=X-A8ANDP(X-A6)<-1THEN 2840
1590  IFX+A6<A8>24THEN 1610
1600  IFX-A6<19ANDP(X-A6)<0ANDP(X+A6+A8)<
      0THEN 2840
1610  IFX+A8>24THEN 1630
1620  IFP(X-A6)<-2ORP(X+A6)=-1)ANDP(X+A6
      +A8)<0THEN 2840
1630  NEXTX:FORX=24TO19STEP-1:IFP(X)<>-1O
      RX>A5THEN 1740
1640  IFP(X-A6)<-2ORP(X-A6)=-1)THEN 2840
1650  IFP(P(X-A8)=-1ORP(X-A8)<-2)THEN 2850
1660  P(X-A6)=1THENA2=X-A8:GOTO 1710
1670  IFP(P(X-A6-A8)=-1ORP(X-A6-A8)<-2)AND
      P(X-A8)=1THENA2=X-A6:GOTO 1710
1680  IFP(P(X-A6-A8)=-1ORP(X-A6-A8)<-2)AND
      P(X-A8)<2THENA2=X-A6:GOTO 1710
1690  IFP(X-A6-A8)=-1ORP(X-A6-A8)<-2)AND
      P(X-A6)<2THENA2=X-A8:GOTO 1710
1700  GOTO 1720
1710  A4=X-A6-A8:GOTO 2860
1720  IFP(X-A6-A8)<0ANDP(X-A6)<2ANDX-A6-A
      8<19THENA4=X-A6-A8:A2=X-A8:GOTO 286
      0
1730  IFP(X-A6-A8)<0ANDP(X-A8)<2ANDX-A6-A
      8<19THENA4=X-A6-A8:A2=X-A6:GOTO 286
      0
1740  NEXTX:FORX=24TO18STEP-1:IFA5<18THEN
      1800
1750  IFP(X)0RX>A5THEN 1790
1760  IFT=P(X-A6):FY=P(X-A8):X6=X-A6:X8=X-
      A8
1770  IF(FT=-2ANDX6<13)OR(FT=-1ORFT<-2)AN
      D(FY=-1ORFY<-2)THEN 2840
1780  IF(FT=-1ORFT<-2OR(X6<19ANDFT<0))AND
      (FY=-1ORFY<-2OR(X8<19ANDFY<0))THEN
      2840
1790  NEXTX
1800  FORX=24TO7STEP-1:IFP(X)<>1ORX>A5THE
      N 1830
1810  IF(P(X-A6)=-1ORP(X-A6)<-2)ANDA1>3TH
      EN 2840
1820  IFX<19ANDP(X-A6)=-1ORP(X-A6)<-2)TH
      EN 2840
1830  NEXTX:FORX=24TO18STEP-1:IFA5<18THEN
      1950
1840  IFP(X)<>-1ORX>A5THEN 1940
1850  IFP(P(X-A6)=-1ORP(X-A6)<-2)OR(P(X-A6
      )<0ANDX-A6<19)THEN 2840
1860  IFX+A6>24THEN 1880
1870  IFP(X+A6)<=-1OR(X+A6>A5ANDP(X+A6)<2
      )THEN 2800
1880  IFX+A8>24THEN 1900
1890  IFP(X+A8)<=-1ORX+A8>A5THEN 2810
1900  IFX+A6>24THEN 1920
1910  IFP(X+A6)<2ANDX+A6>A5THEN 2800
1920  IFX+A8>24THEN 1940
1930  IFP(X+A8)<2ANDX+A8>A5THEN 2810
1940  NEXTX
1950  FORX=24TO19STEP-1:IFA5<19THEN 2020
1960  IFP(X)<>-1ORX>A5THEN 2010
1970  IFX+A6>24THEN 1990
1980  IFP(X+A6>A5ANDP(X+A6)<2)THEN 2800
1990  IFX+A8>24THEN 2010
2000  IFP(X+A8>A5ANDP(X+A8)<2)THEN 2810
2010  NEXTX
2020  IFT(A1>2ANDP(25)>0)ORA1>3THEN 2040
2030  GOTO 2080
2040  FORX=24TO19STEP-1:IFP(X)<>1THEN 207
      0
2050  IFX-A6<19ANDP(X-A6)<0THEN 2840
2060  IFX-A8<19ANDP(X-A8)<0THEN 2850
2070  NEXTX
2080  FORX=1TO12:IFP(X)>-1THEN 2110
2090  IFP(X+A6)=1THEN 2800
2100  IFP(X+A8)=1THEN 2810
2110  NEXTX:FORX=18TO7STEP-1:IFP(X)<>-1OR
      X>A5THEN 2140
2120  IFP(X-A6)<-2ORP(X-A6)=-1THEN 2840
2130  IFP(X-A8)<-2ORP(X-A8)=-1THEN 2850
2140  NEXTX:FORX=1TO24-A6:IFP(X)<>-1THEN
      2160
2150  IFP(X+A6)<=-1OR(X+A6>A5ANDX+A6<25)T
      HEN 2800
2160  NEXTX:FORX=1TO24-A6:IFP(X)<-2AND(P(
      X+A6)<=-1OR(X+A6>A5ANDX+A6<25))THEN
      2800
2170  NEXTX:IFP(1)=-2ANDP(1+A6)<2ANDP(1+A
      6+A8)<0THENX=1:GOTO 2800
2180  FORX=A7TO18:IFP(X)=-1ORP(X)<-2)AND
      P(X+A6)<2THEN 2800
```

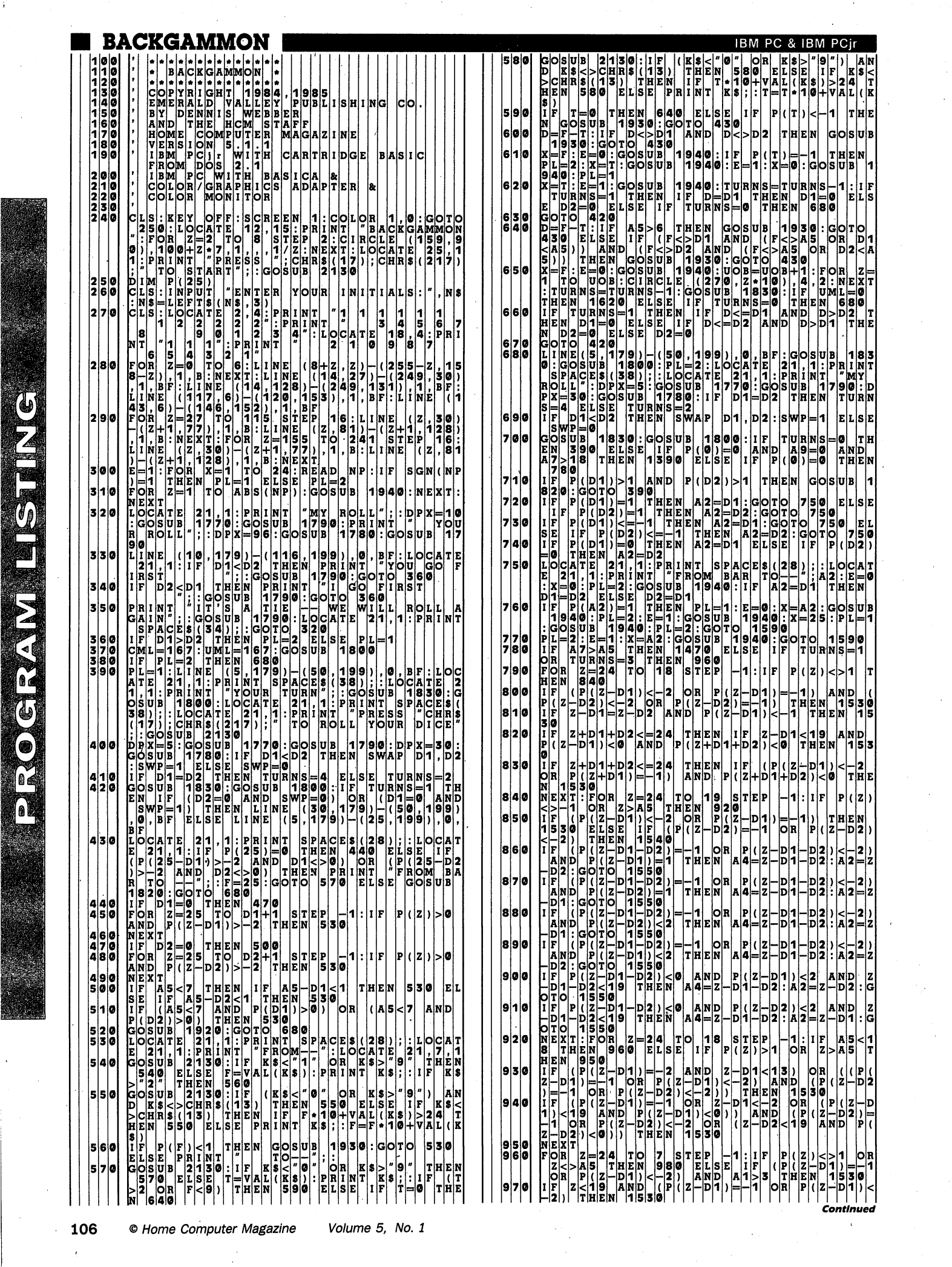
Continued





BACKGAMMON

IBM PC & IBM PCjr



Continued

Continued

BACKGAMMON *Continued*

IBM PC & IBM PCjr

```
1870 NEXT Z:A1=0 TO 18:IF SGN(P(Z))=-1 THEN
1880 FOR A9=-1:G=0 TO 1900
1890 NEXT Z:A1=0:A9=0 TO 24:IF SGN(P(Z))=-1 THEN
1900 FOR Z=R7 TO URN
1910 NEXT Z:A1=0:A7=1 TO 3:IF LOCATE 21,1:PRINT SPACE$(20);
1920 FOR Z=1 TO 7:LOCATE 21,Z:PRINT SPACE$(20);:NEXT Z:RETURN
1930 FOR Z=1 TO 7:LOCATE 21,Z:PRINT INT(P(Z));:NEXT Z:RETURN
1940 IF X<0 THEN 1960 ELSE ZP=1:PRINT "Player 1 ";:RETURN
1950 PRINT "Player 2 ";:RETURN
1960 IF X<6 THEN 1980 ELSE ZP=2:PRINT "Player 2 ";:RETURN
1970 PRINT "Player 1 ";:RETURN
1980
```

```
1990 PY=24-OFFY-((ABS(P(X)))=1)*8):IF E=1
2000 IF X=0 THEN ZP=1 ELSE 2040+((6-
2010 IF 48 THEN OF=0 ELSE OF=48
2020 IF * 16) THEN IF ABS(P(X))>6 THEN OF=
2030 E=1:OF=0 ELSE IF ABS(P(X))>5
2040 E=1:OF=0 ELSE ZP=1
2050 E=25-(P(X))+3),0,-1):IF E=0 THEN ZP=1
2060 IF (X)=P(X)-1 THEN ZP=2
2070 IF P(X)=P(X)+1 ELSE P(X)=
2080 IF P(X)=0 THEN ZP=2
2090 IF X=0 THEN ZP=1
2100 IF X=25 THEN ZP=2
2110 IF X THEN
2120 RETURN
2130 INPUT "WHILE K$="":K$=INKEY$:WEND:RE
2140 DATA 0,0,-2,0,0,0,0,0,5,0,3,0,0,0,-5,5,0,
```

HCM

BACKGAMMON

TI-99/4A

```
100 ! ** BACKGAMMON **
110 ! ** BY **
120 ! ** JEFFREY R. BROWN **
130 ! ** COPYRIGHT 1984 **
140 ! EMERALD ISLE SOFTWARE
150 ! BY MERIDIAN MICROCOMPUTER
160 ! HOME VERSION
170 ! VERSION 1.0
180 ! TSP
190 DIM P$(25),A$(25),B$(25),C$(25),D$(25),X$(25),Y$(25),Z$(25)
200 FOR I=1 TO 25::P$(I)=" ":NEXT I::P$(1)="2":P$(6)="5":P$(8)="3":P$(12)="5":P$(13)="5":P$(17)="3":P$(19)="5":P$(24)="2"
210 A$="A":B$="B":C$="C":D$="D":X$="X":Y$="Y":Z$="Z"
220 CALL CLEAR::CALL SCREEN(4)::CALL COLOR(2,2,16,3,2,16,4,2,16,5,2,16,6,2,16,7,2,16,8,2,16,9,2,16,10,2,16,11,2,16,12,2,16,13,2,16,14,2,16,15,2,16,16,2,16)
230 CALL CHAR(128,"FFFFFFFFFFFFFFFF",129,"FFFFFFFFFFFFFFFF",130,"FFFFFFFFFFFFFFFF",131,"FFFFFFFFFFFFFFFF",132,"FFFFFFFFFFFFFFFF",133,"FFFFFFFFFFFFFFFF",134,"FFFFFFFFFFFFFFFF",135,"FFFFFFFFFFFFFFFF",136,"FFFFFFFFFFFFFFFF",137,"FFFFFFFFFFFFFFFF",138,"FFFFFFFFFFFFFFFF",139,"FFFFFFFFFFFFFFFF",140,"FFFFFFFFFFFFFFFF",141,"FFFFFFFFFFFFFFFF",142,"FFFFFFFFFFFFFFFF",143,"FFFFFFFFFFFFFFFF")
240 CALL CHAR(33,"1818181818001800",34,"1818FF4242FF1800",35,"1818FF4242FF1800",36,"1818FF4242FF1800",37,"1818FF4242FF1800",38,"1818FF4242FF1800",39,"1818FF4242FF1800",40,"1818FF4242FF1800",41,"1818FF4242FF1800",42,"1818FF4242FF1800",43,"1818FF4242FF1800",44,"1818FF4242FF1800",45,"1818FF4242FF1800",46,"1818FF4242FF1800",47,"1818FF4242FF1800",48,"1818FF4242FF1800")
250 CALL CHAR(49,"1818FF4242FF1800",50,"1818FF4242FF1800",51,"1818FF4242FF1800",52,"1818FF4242FF1800",53,"1818FF4242FF1800",54,"1818FF4242FF1800",55,"1818FF4242FF1800",56,"1818FF4242FF1800",57,"1818FF4242FF1800",58,"1818FF4242FF1800",59,"1818FF4242FF1800",60,"1818FF4242FF1800",61,"1818FF4242FF1800",62,"1818FF4242FF1800",63,"1818FF4242FF1800",64,"1818FF4242FF1800")
260 CALL CHAR(65,"1818FF4242FF1800",66,"1818FF4242FF1800",67,"1818FF4242FF1800",68,"1818FF4242FF1800",69,"1818FF4242FF1800",70,"1818FF4242FF1800",71,"1818FF4242FF1800",72,"1818FF4242FF1800",73,"1818FF4242FF1800",74,"1818FF4242FF1800",75,"1818FF4242FF1800",76,"1818FF4242FF1800",77,"1818FF4242FF1800",78,"1818FF4242FF1800",79,"1818FF4242FF1800",80,"1818FF4242FF1800")
270 CALL CHAR(81,"1818FF4242FF1800",82,"1818FF4242FF1800",83,"1818FF4242FF1800",84,"1818FF4242FF1800",85,"1818FF4242FF1800",86,"1818FF4242FF1800",87,"1818FF4242FF1800",88,"1818FF4242FF1800",89,"1818FF4242FF1800",90,"1818FF4242FF1800",91,"1818FF4242FF1800",92,"1818FF4242FF1800",93,"1818FF4242FF1800",94,"1818FF4242FF1800",95,"1818FF4242FF1800",96,"1818FF4242FF1800")
280 CALL CHAR(97,"1818FF4242FF1800",98,"1818FF4242FF1800",99,"1818FF4242FF1800",100,"1818FF4242FF1800",101,"1818FF4242FF1800",102,"1818FF4242FF1800",103,"1818FF4242FF1800",104,"1818FF4242FF1800",105,"1818FF4242FF1800",106,"1818FF4242FF1800",107,"1818FF4242FF1800",108,"1818FF4242FF1800",109,"1818FF4242FF1800",110,"1818FF4242FF1800",111,"1818FF4242FF1800",112,"1818FF4242FF1800")
290 CALL CHAR(113,"1818FF4242FF1800",114,"1818FF4242FF1800",115,"1818FF4242FF1800",116,"1818FF4242FF1800",117,"1818FF4242FF1800",118,"1818FF4242FF1800",119,"1818FF4242FF1800",120,"1818FF4242FF1800",121,"1818FF4242FF1800",122,"1818FF4242FF1800",123,"1818FF4242FF1800",124,"1818FF4242FF1800",125,"1818FF4242FF1800",126,"1818FF4242FF1800",127,"1818FF4242FF1800",128,"1818FF4242FF1800")
300 CALL HCHAR(8,16,66,2):CALL HCHAR(9,16,66,2):CALL HCHAR(10,16,66,2):CALL HCHAR(11,16,66,2):CALL HCHAR(12,16,66,2):CALL HCHAR(13,16,66,2):CALL HCHAR(14,16,66,2):CALL HCHAR(15,16,66,2):CALL HCHAR(16,16,66,2):CALL HCHAR(17,16,66,2)
310 CALL HCHAR(8,15,67,2):CALL HCHAR(9,15,67,2):CALL HCHAR(10,15,67,2):CALL HCHAR(11,15,67,2):CALL HCHAR(12,15,67,2):CALL HCHAR(13,15,67,2):CALL HCHAR(14,15,67,2):CALL HCHAR(15,15,67,2):CALL HCHAR(16,15,67,2):CALL HCHAR(17,15,67,2)
320 CALL HCHAR(8,14,68,2):CALL HCHAR(9,14,68,2):CALL HCHAR(10,14,68,2):CALL HCHAR(11,14,68,2):CALL HCHAR(12,14,68,2):CALL HCHAR(13,14,68,2):CALL HCHAR(14,14,68,2):CALL HCHAR(15,14,68,2):CALL HCHAR(16,14,68,2):CALL HCHAR(17,14,68,2)
330 CALL HCHAR(8,13,69,2):CALL HCHAR(9,13,69,2):CALL HCHAR(10,13,69,2):CALL HCHAR(11,13,69,2):CALL HCHAR(12,13,69,2):CALL HCHAR(13,13,69,2):CALL HCHAR(14,13,69,2):CALL HCHAR(15,13,69,2):CALL HCHAR(16,13,69,2):CALL HCHAR(17,13,69,2)
340 CALL HCHAR(8,12,70,2):CALL HCHAR(9,12,70,2):CALL HCHAR(10,12,70,2):CALL HCHAR(11,12,70,2):CALL HCHAR(12,12,70,2):CALL HCHAR(13,12,70,2):CALL HCHAR(14,12,70,2):CALL HCHAR(15,12,70,2):CALL HCHAR(16,12,70,2):CALL HCHAR(17,12,70,2)
350 CALL HCHAR(8,11,71,2):CALL HCHAR(9,11,71,2):CALL HCHAR(10,11,71,2):CALL HCHAR(11,11,71,2):CALL HCHAR(12,11,71,2):CALL HCHAR(13,11,71,2):CALL HCHAR(14,11,71,2):CALL HCHAR(15,11,71,2):CALL HCHAR(16,11,71,2):CALL HCHAR(17,11,71,2)
360 DISPLAY AT(19,0)BEEP:"                ROLL ":"&C$
370 IF A13=1 THEN 380 ELSE CALL KEY(0,K,S
380 RANDOMIZE::A8=MIN(
390 GOTO 650
400 SUB GAMMENT
410 CALL VCHAR(13,16,C,P(25)):::CALL SO:
420 FOR Y=2 TO 12 STEP 10 :: CALL VCHAR(Y,16,B,P(0)):: NEXT Y :: CALL VCHAR(
430 FOR Y=9 TO 1 STEP -1 :: CALL HCHAR(Y,X,32,12):: NEXT X
440 IF X>12 THEN 480 ELSE PT=SGN(SG
450 IF X=0 THEN 510 ELSE NP=MIN(ABS(P(X),5)):: CALL VCHAR(16-NP,PX(X),96+PT
460 IF ABS(P(X))>5 THEN NP=MIN(ABS(P(X),5)):: CALL VCHAR(16-NP,PX(X),97+PT*8
470 CALL HCHAR(4,PX(X),E,5)::PT=SGN(SG
480 IF X=0 THEN 510 ELSE NP=MIN(ABS(P(X),5)):: CALL VCHAR(4,PX(X),96+PT
490 IF ABS(P(X))>5 THEN NP=MIN(ABS(P(X),5)):: CALL VCHAR(16-NP,PX(X),97+PT*8
500 NEXT X
510 FOR X=1 TO 25::IF P(X)=0 THEN 530 ELSE IF P(X)<0 THEN A11=1 ELSE A11=0
520 A10=A10+(P(X)*X):NEXT X:SIZE(@):IF A11=1 THEN DISPLAY AT(24,A12)SIZE(
530 NEXT X:IF A14=1 THEN DISPLAY AT(22,2)
540 IF A12=1 THEN DISPLAY AT(24,A12)SIZE(
550 DISPLAY AT(22,25):A10::DISPLAY AT(24,26):A11:ABS(25*P(0)))
560 FOR X=25 TO 0 STEP -1 :: IF P(X)<0 THEN A5=A5+P(X):: NEXT X
570 GOTO 580 ELSE IF P(X)<-1 THEN A5=A5+P(X):: NEXT X
580 IF A5=15 THEN 600 ELSE 590
590 PRINT @0,"Player 1 Wins":::GOTO 620
600 PRINT @0,"Player 2 Wins"
610 FOR X=1 TO 24::IF SGN(P(X))=-1 THEN A7=A7+P(X):: NEXT X
620 IF A7=-15 THEN 640 ELSE 630
630 PRINT @0,"Player 1 Wins":::GOTO 650
640 PRINT @0,"Player 2 Wins"
650 DISPLAY AT(22,22)SIZE(3):CHR$(111+A
    IF A15>@ THEN CALL HCHAR(22,26
    ,111+A8)
```

Continued

BACKGAMMON *Continued*

Continued

BACKGAMMON *Continued*


```
1500 IF X+A6<25 THEN IF SGN(P(X+A6))=-@
     THEN X+A6>24 AND X=A7 THEN 1690
1510 IF X+A8>24 AND X+A8<26 THEN I
1520 IF P(X+A8)<26 THEN I
     F (X+A8)<26 THEN IF P(X+A8)=-2)THEN
1530 IF (X+A8)<26 THEN IF SGN(P(X+A8))=-
     @ OR P(X+A8)=-@ THEN 1700
1540 IF P(X)=@ THEN IF X+A6<25 THEN IF
     X+A8<25 AND P(X)=-@ THEN 1700
1550 NEXT X :: FOR X=24 TO A7 STEP -@
     :: IF P(X)>-@ OR P(X+A6)=0 AND P(X+A6)<2)THEN 1690
1560 NEXT X :: FOR X=24 TO P(X)=0 AND P(X)>-2)THEN 1570
1570 NEXT X
1580 IF SGN(P(25-A6))=-@ THEN 1710 ELSE
     IF SGN(P(25-A8))=-@ THEN 1720
1590 IF A7+A6>24 THEN 1680 ELSE IF A7+A8
     >24 THEN 1690
1600 FOR X=A7 TO 24 STEP -1 :: IF P(X)>-@ OR (X+A
     6)>24 OR P(X+A6)=0 THEN 1620
1610 IF P(X+A6)=0 THEN 1690 ELSE IF P(X+
     A8)=0 THEN 1700 ELSE NEXT X
1620 FOR X=A7 TO 24 STEP -1 :: IF P(X)
     >-@ THEN 1640 ELSE IF P(X+A6)=0 THEN 1690
     ELSE IF P(X+A8)=0 THEN 1700 ELSE NEXT X
1630 IF X+A6<25 THEN 1690 ELSE IF X+A8<
     25 THEN 1640 ELSE IF P(X+A8)<2 THEN 1700
1640 NEXT X :: IF A7+A6<25 THEN IF P(A7+
     A6)<2 THEN 1670
1650 IF A7+A8<25 THEN IF P(A7+A8)<2 THEN
     1670
1660 CALL CNT(A13)::GOTO 3500
1670 A4=A7+A6 :: GOTO 1750
1680 A4=A7+A8 :: GOTO 1750
1690 A4=X+A6 :: GOTO 1750
1700 A4=X+A8 :: GOTO 1750
1710 A4=25-A6 :: A2=25 :: GOTO 1750
1720 A4=25-A8 :: A2=25 :: GOTO 1750
1730 A4=X-A6 :: A2=X :: GOTO 1750
1740 A4=X-A8 :: A2=X
1750 P(A4)=P(A4)+@ :: IF A2-A4=A6 THEN A
1760 IF A2>24 THEN A12=A12+@ :: A2=A16
     :: GOSUB 1790 :: GOTO 1
1770 IF P(A2)=@ THEN P(25)=P(25)+@ :: P(
     A2)=-@ :: GOSUB 1790 :: P(A2)=P(A2)-@ :: G
     OTO 1780
1780 IF A11=0 THEN A20=@ :: IF A15=A15-@
     :: GOTO 1800 ELSE A13=0 :: GOTO 3500
1790 DISPLAY AT(19,4)BEEP::"FROM--"&STR$(
     A4)&"TO-"&STR$(A2)::RETURN
1800 IF A7<7 AND A12<0)OR(A14=0 AND A5>
     18) OR A14=0 THEN B$="G A M
     E OVER"::GOTO 3500
1810 IF A20=@ THEN CALL SOUND(2000,110,3
     0,825,30,-4,0)
1820 DISPLAY AT(19,3):"WINNER IS "
     &CHR$(88-X)&"!" :: "PLAY AGAIN?" :: P
     OSITION(R,C)::ACCEPT AT(R,C)SIZE(1)B
     EEP:A$ :: IF A$="Y" OR A$="y" THEN 1
     830 ELSE 3500
1830 IF S=0 THEN 1830 ELSE RESTORE :: GO
     TO 2
1840 DATA 29,27,25,23,21,19,13,11,9,7,5,
     3,5,9,7,9,11,13,19,21,23,25,27,29
1850 SUB DISPLAY(X):: FOR I=@ TO 6 :: DI
     SPLAY AT(19,(17)):"INVALID MOVE" :: CA
     LL HCHAR(19,1,32,32)::NEXT I :: SUBEND
1860 SUB A3(X):: FOR A3=@ TO X :: NEXT A
     3 :: SUBEND
1870 SUB CNT(A13):: FOR X=@ TO 9 :: DISPC
     LAY AT(19,14)BEEP:"CAN'T MOVE" :: CALL HCHAR(19,1,32,32)
     :: NEXT X :: A13=ABS(A13-1)):: SUBEND
1880 SUB BLK(N$,A13)::DISPLAY AT(19,@)BEEP:N$&"IN
     NER TABLE BLOCKED" :: CALL A3(798)
```

HCM

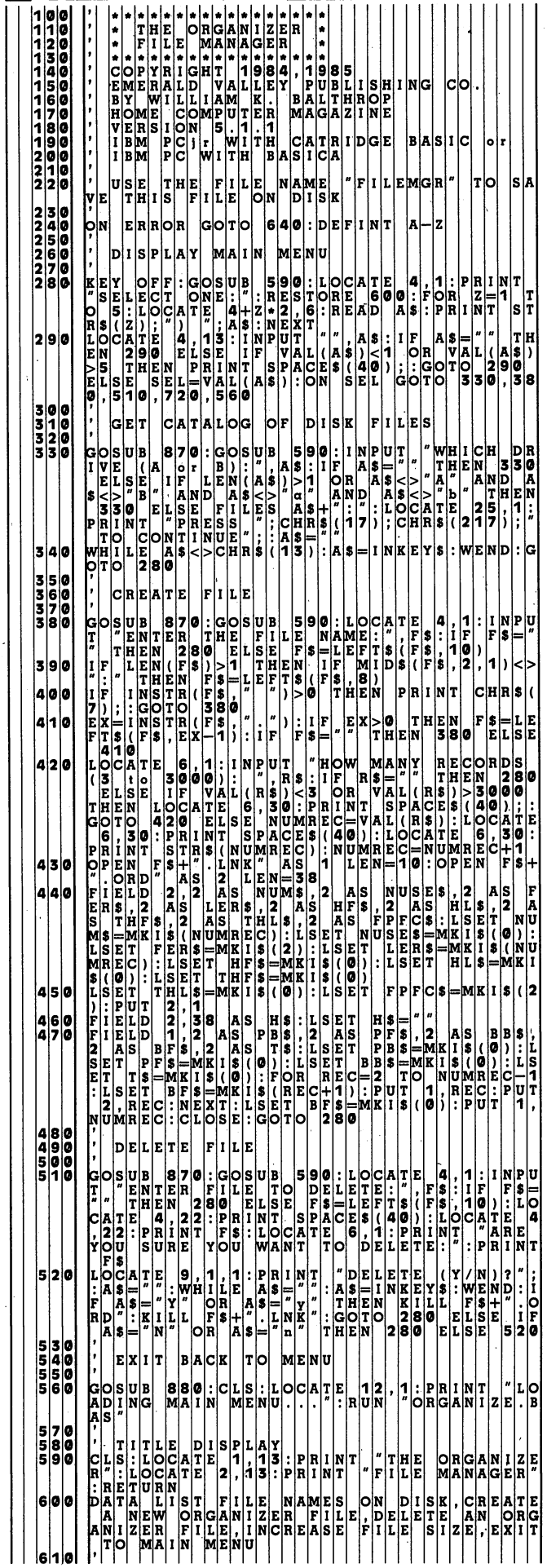
The Organizer for the IBM PC & IBM PCjr

MAIN MENU


```
100 '
110 '          ***********
120 '          * THE ORGANIZER *
130 '          ***********
140 '          MAIN MENU
150 '          ***********************
160 '          * COPYRIGHT 1984, 1985 *
170 '          * BY WILLIAM PALTHROPE *
180 '          * HOME COMPUTER MAGAZINE *
190 '          * IBM PCjr WITH 1.1 CARTRIDGE BASIC or *
200 '          * IBM PC WITH BASIC *
210 '          ***********************
220 '
230 '          USE THE FILE NAME "ORGANIZE"
240 '          TO SAVE THIS PROGRAM ON DISK
250 ON ERROR GOTO 390
260 '
270 '
280 '          DISPLAY MAIN MENU
290 KEY OFF
300 CLS :: DIM SC$(20,1)::LOCATE 1,13:PRINT
    "MAIN MENU":LOCATE 4,1:PRINT "SELECT
    ONE:":GOSUB 510 :: GOSUB 500 :: IF VAL(A$
    )<1 OR VAL(A$)>4 THEN 310
310 ON VAL(A$) GOSUB 330,340,320,350
320 CLS :: LOCATE 12,1:PRINT "PLACE YOUR
    DISK IN DRIVE A:" :: GOTO 440
330 CLS :: LOCATE 12,1:PRINT "PRINTING
    DIRECTORY" :: CHAIN MERGE "DIR",510,ALL
340 CLS :: LOCATE 12,1:PRINT "LOADING
    ORGANIZER" :: CHAIN MERGE "ORGANIZE",510,
    ALL
350 SUN
360 '
370 '          ERROR ROUTINE
380 '
390 SOUND 110,30:CLS :: LOCATE 12,1:IF ERR
    =53 THEN PRINT "DISK NOT FOUND" ELSE IF
    ERR=71 THEN PRINT "DISK NOT IN DRIVE A:"
    ELSE PRINT "ERROR #";ERR;" IN LINE";ERL
400 IF ERR=72 THEN PRINT "CHECK YOUR DISK
    DRIVE" ELSE IF ERR=68 THEN PRINT "I CAN'T ACCESS
    DISK" ELSE PRINT "ERROR #";ERR
410 IF ERR=68 THEN PRINT "DEVICE ERROR"
    ELSE IF ERR=71 THEN PRINT "DISK NOT IN
    DRIVE" ELSE PRINT "ERROR #";ERR
420 IF ERR=68 THEN PRINT "I CAN'T ACCESS
    DISK" ELSE IF ERR=71 THEN PRINT "DISK NOT
    IN DRIVE" ELSE PRINT "ERROR #";ERR
430 PRINT "ERROR #";ERR;" IN LINE";ERL
440 LOCATE 20,1:PRINT "PRESS ";CHR$(17)
    ;"ENTER";CHR$(16);" TO CONTINUE";
450 K$=INKEY$ :: IF K$="" THEN 450 ELSE RU
N
460
470 ROUTINE TO DISPLAY
480 PROGRAM OPTIONS
490
500 FOR Z=1 TO SC :: LOCATE 4+Z*2,6:PRINT
    STR$(Z);SC$(Z,0):NEXT :: RETURN
510 RESTORE 560
520 READ PROMPT$,FILENAME$:IF PROMPT$="*
    " THEN RETURN
530 SC=SC+1:SC$(SC,0)=PROMPT$:SC$(SC,1)
    =FILENAME$:GOTO 520
540 '
550 MENU DATA ---- MENU SELECTION,PROGR
    AM NAME
560 DATA OUTLINE,EDITOR,OUTLINE
570 DATA REPORTS
580 DATA FILEMGR
590 DATA QUIT,*
600 DATA *
610 KEY 1,"LIST"+CHR$(13):KEY 2,"RUN"+CH
    R$(13):KEY 3,"LOAD"+CHR$(34):KEY 4,"SAVE"+CHR$(34)
    :KEY 5,"CONT"+CHR$(13):"+CHR$(34)+"LPT1:"+CHR$(34)
    +"TRON"+CHR$(13):KEY 7,"TROFF"+CHR$(13):KEY 8,"KEY
    LIST"+CHR$(13):KEY 9,"KEY OFF"+CHR$(13):KEY 10,"SCR
    EEN 0,0,0"+CHR$(13)):KEY ON:END
```

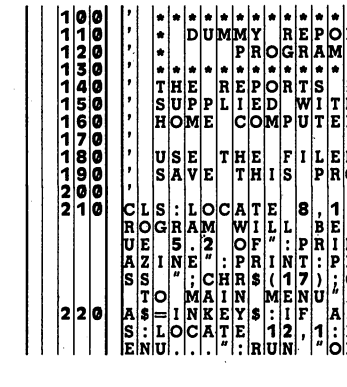
HCM

FILE MANAGER



HCM

REPORTS



HCM

PROGRAM LISTING

OUTLINE EDITOR


```
100 '  ********** ORGANIZER **********
110 '  *  THE OUTLINE EDITOR  *
120 '  *  COPYRIGHT 1984, 1985  *
130 '  *  BY MERLIN VALLEY PUBLISHING CO.  *
140 '  *  VERSION 5.1 WITH CARTRIDGE BASIC or  *
150 '  *  IBM COLOR/GRAPHICS ADAPTER  *
160 '  **********************************
170 '
180 '
190 '
200 '
210 '
220 '
230 '
240 '
250 '
260 '
270 ON ERROR GOTO 37500:DEF INT A-Z:DIM SC$(21),SZ,CH$(21),Z
280 KEY 15,"LIST"+CHR$(34)+"CRT:"+CHR$(34)+CHR$(13):KEY 16,"AUTO "+CHR$(13):KEY 17,"SAVE "+CHR$(34):KEY 18,"LOAD "+CHR$(34):KEY 19,"FILES "+CHR$(34):KEY 20,"RUN "+CHR$(34)+CHR$(13):KEY 21,"LIST "+CHR$(34):KEY 22,"NEW"+CHR$(13):CX=2
290 '
300 '
310 '
320 '  ********** LOCATE 12,1:PRINT "TO EXIT — PRESS ESC" **********
330 '  ********** PRINT "WHEN PROMPTED FOR FILE NAME" **********
340 '  ********** PRINT "ORGANIZER FILE." **********
350 '  ********** PRINT "TO EXIT — PRESS ESC" **********
360 '  ********** PRINT "WITH NO FILE NAME" **********
370 '  ********** PRINT "OUTLINE PLEASE WAIT..." **********
380 '  ********** PRINT "LOADING" **********
390 '  ********** READING RECORD #" **********
400 '
410 '
420 '  ********** OUTLINE EDITOR — MAIN CONTROL LOOP **********
430 '
440 '
450 '
460 '
470 '
480 '
490 '
500 '
510 '  ********** EDIT LINE **********
520 '
530 '
540 '
550 '  ********** TEXT EDITOR CONTROL LOOP **********
560 '
570 '
580 '
590 '
600 '  ********** TEXT EDIT — LOAD SCREEN **********
610 '
620 '
630 '
640 '
650 '
660 '
670 '
680 '  ********** TEXT EDIT — SAVE SCREEN **********
690 '
700 '
710 '
720 '
730 '  ********** GRAB & COPY LINE ROUTINE **********
740 '
750 '
760 '
770 '
780 '
790 '
800 '
810 '
820 '
830 '
840 '
850 '
860 '
870 '
880 '
890 '
900 '
910 '
920 '
930 '
940 '
950 '
960 '
970 '
980 '
990 '
1000 '
1010 '
1020 '
1030 '
1040 '
1050 '
1060 '
1070 '
1080 '
1090 '
1100 '
1110 '
1120 '
1130 '
1140 '
1150 '
1160 '
1170 '  ********** INSERT LINE ROUTINE **********
```

Continued

```
1180  IF MK$=SC$ THEN 1210 ELSE IF MK$=LC$ THEN 1220 ELSE IF MK$=CP$ THEN 1230 ELSE IF MK$=SP$ THEN 1240 ELSE IF MK$=BS$ THEN 1250 ELSE IF MK$=CR$ THEN 1260 ELSE IF MK$=DL$ THEN 1270 ELSE IF MK$=RT$ THEN 1280 ELSE IF MK$=TB$ THEN 1290 ELSE IF MK$=ES$ THEN 1300 ELSE IF MK$=UP$ THEN 1310 ELSE IF MK$=DN$ THEN 1320 ELSE IF MK$=LT$ THEN 1330 ELSE IF MK$=HM$ THEN 1340 ELSE IF MK$=EL$ THEN 1350 ELSE IF MK$=SF$ THEN 1360 ELSE IF MK$=SB$ THEN 1370 ELSE IF MK$=IN$ THEN 1380 ELSE IF MK$=DE$ THEN 1390 ELSE IF MK$=CF$ THEN 1400 ELSE IF MK$=CB$ THEN 1410 ELSE IF MK$=CL$ THEN 1420 ELSE IF MK$=CE$ THEN 1430 ELSE IF MK$=CC$ THEN 1440 ELSE IF MK$=CV$ THEN 1450 ELSE IF MK$=CX$ THEN 1460 ELSE IF MK$=CM$ THEN 1470 ELSE IF MK$=CA$ THEN 1480 ELSE IF MK$=CD$ THEN 1490 ELSE IF MK$=CT$ THEN 1500 ELSE IF MK$=CN$ THEN 1510 ELSE IF MK$=CS$ THEN 1520 ELSE IF MK$=CI$ THEN 1530 ELSE IF MK$=CO$ THEN 1540 ELSE IF MK$=CW$ THEN 1550 ELSE IF MK$=CU$ THEN 1560 ELSE IF MK$=CG$ THEN 1570 ELSE IF MK$=CH$ THEN 1580
1190  MK$=SC$: GOSUB 2940: RETURN
1200  MK$=LC$: GOSUB 2940: RETURN
1210  MK$=CP$: GOSUB 2940: RETURN
1220  MK$=SP$: GOSUB 2940: RETURN
1230  MK$=BS$: GOSUB 2940: RETURN
1240  MK$=CR$: GOSUB 2940: RETURN
1250  MK$=DL$: GOSUB 2940: RETURN
1260  MK$=RT$: GOSUB 2940: RETURN
1270  MK$=TB$: GOSUB 2940: RETURN
1280  MK$=ES$: GOSUB 2940: RETURN
1290  MK$=UP$: GOSUB 2940: RETURN
1300  MK$=DN$: GOSUB 2940: RETURN
1310  MK$=LT$: GOSUB 2940: RETURN
1320  MK$=HM$: GOSUB 2940: RETURN
1330  MK$=EL$: GOSUB 2940: RETURN
1340  MK$=SF$: GOSUB 2940: RETURN
1350  MK$=SB$: GOSUB 2940: RETURN
1360  MK$=IN$: GOSUB 2940: RETURN
1370  MK$=DE$: GOSUB 2940: RETURN
1380  MK$=CF$: GOSUB 2940: RETURN
1390  MK$=CB$: GOSUB 2940: RETURN
1400  MK$=CL$: GOSUB 2940: RETURN
1410  MK$=CE$: GOSUB 2940: RETURN
1420  MK$=CC$: GOSUB 2940: RETURN
1430  MK$=CV$: GOSUB 2940: RETURN
1440  MK$=CX$: GOSUB 2940: RETURN
1450  MK$=CM$: GOSUB 2940: RETURN
1460  MK$=CA$: GOSUB 2940: RETURN
1470  MK$=CD$: GOSUB 2940: RETURN
1480  MK$=CT$: GOSUB 2940: RETURN
1490  MK$=CN$: GOSUB 2940: RETURN
1500  MK$=CS$: GOSUB 2940: RETURN
1510  MK$=CI$: GOSUB 2940: RETURN
1520  MK$=CO$: GOSUB 2940: RETURN
1530  MK$=CW$: GOSUB 2940: RETURN
1540  MK$=CU$: GOSUB 2940: RETURN
1550  MK$=CG$: GOSUB 2940: RETURN
1560  MK$=CH$: GOSUB 2940: RETURN
1570  MK$=CR$: GOSUB 2940: RETURN
1580  RETURN
1590  '
1600  '
1610  '
1620  '
1630  '
1640  '
1650  '
1660  '
1670  '
1680  '
1690  '
1700  '
1710  '
1720  '
1730  '
1740  '
1750  '
1760  '
1770  '
1780  '
1790  '
1800  '
1810  '
1820  '
1830  '
1840  '
1850  '
1860  '
1870  '
1880  '
1890  '
1900  '
1910  '
1920  '
1930  '
1940  '
1950  '
1960  '
1970  '
```

Continued

OUTLINE EDITOR *Continued*

IBM PC & IBM PCjr

```
1980 CLS:LOCATE 7,7:PRINT "OUTLINE EDITOR":LOCATE 9,9:PRINT "Copyright (C) 1985":LOCATE 11,11:PRINT "by Ric
     hard Otis":LOCATE 13,13:PRINT "Version 2.1":LOCATE 15,15:PRINT "IBM PC & PCjr":LOCATE 17,17:PRINT "Home
     Computer Magazine":LOCATE 19,19:PRINT "Greensboro, NC":LOCATE 21,21:PRINT "(919) 275-9809":DEFINT A-Z:
     DIM A$(1),B$(1),C$(1),D$(1),E$(1),F$(1),G$(1),H$(1),I$(1),J$(1),K$(1),L$(1),M$(1),N$(1),O$(1),P$(1),Q$
     (1),R$(1),S$(1),T$(1),U$(1),V$(1),W$(1),X$(1),Y$(1),Z$(1):DEFSTR A-Z:DEFINT I-N:DEFSNG O-Y:DIM SP(1),
     LEVEL(1),LINK(1),LEN(1),CHILD(1),PARENT(1),NEXT(1),PREV(1),LINE(1),LEVEL$(1),LINK$(1),LEN$(1),CHILD$(1
     ),PARENT$(1),NEXT$(1),PREV$(1),LINE$(1):DEFSTR A-Z:DEFINT I-N:DEFSNG O-Y:DIM SP(1),LEVEL(1),LINK(1),L
     EN(1),CHILD(1),PARENT(1),NEXT(1),PREV(1),LINE(1),LEVEL$(1),LINK$(1),LEN$(1),CHILD$(1),PARENT$(1),NEXT$
     (1),PREV$(1),LINE$(1)
1990 DEFINT A-Z:DEFSNG O-Y:DIM SP(1),LEVEL(1),LINK(1),LEN(1),CHILD(1),PARENT(1),NEXT(1),PREV(1),LINE(1),L
     EVEL$(1),LINK$(1),LEN$(1),CHILD$(1),PARENT$(1),NEXT$(1),PREV$(1),LINE$(1):DEFSTR A-Z:DEFINT I-N:DEFSNG
     O-Y:DIM SP(1),LEVEL(1),LINK(1),LEN(1),CHILD(1),PARENT(1),NEXT(1),PREV(1),LINE(1),LEVEL$(1),LINK$(1),LEN
     $(1),CHILD$(1),PARENT$(1),NEXT$(1),PREV$(1),LINE$(1)
2000 ON ERROR GOTO 19000
2010 GOSUB 2020,2110,2170:GOSUB 2900:GOTO 2010
2020 PRINT:PRINT "Enter name of file to be loaded (ENTER for new file)":INPUT F$:IF F$="" THEN RETURN
2030 IF INSTR(F$,".")=0 THEN F$=F$+".OUT"
2040 OPEN "I",1,F$:PRINT "Loading file...":INPUT #1,N:FOR I=1 TO N:INPUT #1,LEVEL(I),LINK(I),LEN(I),CHILD
     (I),PARENT(I),NEXT(I),PREV(I),LINE(I):NEXT I:FOR I=1 TO N:INPUT #1,LEVEL$(I),LINK$(I),LEN$(I),CHILD$(
     I),PARENT$(I),NEXT$(I),PREV$(I),LINE$(I):NEXT I:CLOSE 1:RETURN
2050 IF RECC=SUBC THEN SUBC=SUBC+1:IF SUBC>20 THEN SUBC=20:GOSUB 20260:SUBC=SUBC-1:GOTO 20250 ELSE RETURN
2060 IF SUBC<1 THEN SUBC=1:GOSUB 20260:SUBC=SUBC+1:GOTO 20250 ELSE RETURN
2070 IF KL=1 THEN KL=0:GOSUB 20260:KL=1:GOTO 20250 ELSE RETURN
2080 IF RECC=1 THEN RETURN ELSE RECC=RECC-1:GOSUB 20260:RECC=RECC+1:GOTO 20250
2090 IF RECC=NUMC THEN RETURN ELSE RECC=RECC+1:GOSUB 20260:RECC=RECC-1:GOTO 20250
2100 RECC=1:GOSUB 20260:GOTO 20250
2110 PRINT "ENTER COMMAND (? FOR HELP)":INPUT A$:IF A$="" THEN RETURN
2120 ON INSTR("?ABCDEFGHIJKLMNOPQRSTUVWXYZ",LEFT$(A$,1)) GOSUB 2130,2140,2150,2160,2170,2180,2190,2200,2
     210,2220,2230,2240,2250,2260,2270,2280,2290,2300,2310,2320,2330,2340,2350,2360
2130 PRINT "? - Help":PRINT "A - Add line":PRINT "B - Beginning of file":PRINT "C - Change line":PRINT "D
     - Delete line":PRINT "E - End of file":PRINT "F - Find string":PRINT "G - Go to line":PRINT "H - Help"
     :PRINT "I - Insert line":PRINT "J - Join lines":PRINT "K - Kill file":PRINT "L - List file":PRINT "M
     - Move line":PRINT "N - Number lines":PRINT "O - Output file":PRINT "P - Print file":PRINT "Q - Quit"
     :PRINT "R - Read file":PRINT "S - Save file":PRINT "T - Type line":PRINT "U - Undelete line":PRINT "V
     - View file":PRINT "W - Write file":PRINT "X - Exit":PRINT "Y - Yank lines":PRINT "Z - Zap lines":RE
     TURN
2140 GOSUB 2030:IF LINK(SP,1)>0 THEN GOSUB 2030:GOTO 2140 ELSE RETURN
2150 LINK(SP,0)=0:LEVEL=1:LINK(SP,1)=0:LEN=0:LINE="":LEVEL$="1":LINK$="0":LEN$="0":LINE$="":RETURN
2160 IF LINK(SP,1)=0 THEN RETURN ELSE SP=LINK(SP,1):GOTO 2160
2170 IF PREV(SP)=0 THEN RETURN ELSE SP=PREV(SP):GOTO 2170
2180 PRINT "Enter string to find":INPUT F$:IF F$="" THEN RETURN
2190 IF INSTR(LINE(SP),F$)>0 THEN RETURN ELSE IF NEXT(SP)=0 THEN RETURN ELSE SP=NEXT(SP):GOTO 2190
2200 PRINT "Enter line number":INPUT N:IF N<1 OR N>NUMC THEN RETURN ELSE SP=1:FOR I=1 TO N-1:SP=NEXT(SP)
     :NEXT I:RETURN
2210 RETURN
2220 '
2230 '       FORMAT TEXT FUNCTION
2240 '
2250 A$="":B$="":C$="":D$="":E$="":F$="":G$="":H$="":I$="":J$="":K$="":L$="":M$="":N$="":O$="":P$="":Q$
     ="":R$="":S$="":T$="":U$="":V$="":W$="":X$="":Y$="":Z$="":BR=0:GR=0:DN=0:GOSUB 2510:BR=1:GOSUB 2510:
     GR=1:GOSUB 2510:DN=1:GOSUB 2510:IF LEN=0 THEN RETURN
2260 IF LEFT$(LINE,1)=" " THEN A$=" ":LINE=MID$(LINE,2):LEN=LEN-1:GOTO 2260
2270 IF LEN<=70 THEN A$=A$+LINE:LINE="":LEN=0:GOTO 2290
2280 FOR I=70 TO 1 STEP -1:IF MID$(LINE,I,1)=" " THEN A$=A$+LEFT$(LINE,I-1):LINE=MID$(LINE,I+1):LEN=LEN
     -I:GOTO 2290 ELSE NEXT I:A$=A$+LEFT$(LINE,70):LINE=MID$(LINE,71):LEN=LEN-70
2290 PRINT A$:GOTO 2260
2300 '
2310 '
2320 '
2330 '
2340 '
2350 '
2360 '
2370 IF LEN(B$)+LEN(LINE)<255 THEN B$=B$+" "+LINE ELSE B$=MID$(B$,1,255):LINE=" "+LINE:GOSUB 2900:GOTO
     2370
2380 RETURN
2390 '
2400 '
2410 '
2420 '
2430 '
2440 '
2450 '
2460 '
2470 '
2480 '
2490 '
2500 '
2510 '
2520 '
2530 '
2540 '
2550 '
2560 '
2570 '
2580 '
2590 '
2600 '
2610 '
2620 '
2630 '
2640 '
2650 '
2660 '
2670 '
2680 '
2690 '
2700 IF F="" THEN F="*.*"
2710 FILES F:PRINT:PRINT "Enter name of file to be loaded (ENTER for new file)":INPUT F$:IF F$="" THEN RE
     TURN
2720 IF INSTR(F$,".")=0 THEN F$=F$+".OUT"
2730 OPEN "I",1,F$:PRINT "Loading file...":INPUT #1,N:FOR I=1 TO N:INPUT #1,LEVEL(I),LINK(I),LEN(I),CHILD
     (I),PARENT(I),NEXT(I),PREV(I),LINE(I):NEXT I:FOR I=1 TO N:INPUT #1,LEVEL$(I),LINK$(I),LEN$(I),CHILD$(
     I),PARENT$(I),NEXT$(I),PREV$(I),LINE$(I):NEXT I:CLOSE 1:RETURN
2740 '
2750 '
2760 IF LEN(B$)+LEN(LINE)<255 THEN B$=B$+" "+LINE ELSE B$=MID$(B$,1,255):LINE=" "+LINE:GOSUB 2900:GOTO
     2760
2770 '
2780 '
2790 '
```

Continued

OUTLINE EDITOR *Continued*

```
2800 IF LCP=1 THEN ENCR$=CHR$(Z):ELSE GOSUB 7000:
2810 FOR LCP=1 TO 61 STEP 2:IF LCP=61 THEN ENCR$=ENCR$+CHR$(Z):ELSE ENCR$=ENCR$+" ":
2820 NEXT:RETURN
2830 '
2840 '
2850 '
2860 CLS:LOCATE 1,1,1:LCP=1:PRINT;:PRINT"USE ^ TO MOVE UP, V TO MOVE DOWN, < TO MOVE LEFT, > TO MOVE RIGHT";:
2870 PRINT;:PRINT"USE SPACE TO ADD A LINE, D TO DELETE A LINE, E TO EDIT A LINE, I TO INSERT A LINE, Q TO QUIT";:
2880 PRINT;:PRINT"USE HOME TO GO TO TOP, END TO GO TO BOTTOM, PGUP TO GO UP 20 LINES, PGDN TO GO DOWN 20 LINES";:
2890 PRINT;:PRINT"USE TAB TO GO TO NEXT TAB, SHIFT TAB TO GO TO PREVIOUS TAB";:
2900 GOSUB 7600:LSET H$=HT$:PUT 2,REC:RETURN
2910 '
2920 '
2930 '
2940 '
2950 '
2960 '
2970 '
2980 '
2990 '
3000 '
3010 '
3020 '
3030 '
3040 '
3050 '
3060 '
3070 '
3080 '
3090 '
3100 '
3110 '
3120 '
3130 '
3140 '
3150 '
3160 '
3170 '
3180 '
3190 '
3200 '
3210 '
3220 IF LCP=10 THEN ZCR=1:GOSUB 7000:LCP=9:
3230 IF LCP=9 THEN ZCR=2:GOSUB 7000:LCP=8:
3240 IF LCP=8 THEN ZCR=3:GOSUB 7000:LCP=7:
3250 IF LCP=7 THEN ZCR=4:GOSUB 7000:LCP=6:
3260 IF LCP=6 THEN ZCR=5:GOSUB 7000:LCP=5:
3270 IF LCP=5 THEN ZCR=6:GOSUB 7000:LCP=4:
3280 IF LCP=4 THEN ZCR=7:GOSUB 7000:LCP=3:
3290 IF LCP=3 THEN ZCR=8:GOSUB 7000:LCP=2:
3300 IF LCP=2 THEN ZCR=9:GOSUB 7000:LCP=1:
3310 IF LCP=1 THEN ZCR=10:GOSUB 7000:
3320 RETURN
3330 '
3340 '
3350 '
3360 '
3370 '
3380 '
3390 '
3400 '
3410 '
3420 '
3430 '
3440 '
3450 '
3460 '
3470 '
3480 '
3490 '
3500 '
3510 '
3520 '
3530 '
3540 '
3550 '
3560 '
3570 '
3580 '
3590 '
3600 '
3610 '
3620 '
3630 '
3640 '
3650 '
3660 '
3670 '
3680 '
3690 '
3700 '
3710 '
3720 '
3730 '
3740 '
3750 '
3760 '
3770 '
3780 '
3790 '
3800 '
3810 IF LCP=1 THEN 3850 ELSE IF MD=1 THEN LCP=LCP-1:GOSUB 2910:GOTO 3810 ELSE IF LCP=10 THEN 3810 ELSE LCP=LCP+1:GOSUB 2910:
```

Continued

PROGRAM LISTING

OUTLINE EDITOR *Continued*

IBM PC & IBM PCjr

```
3820 LOCATE 8,1:PRINT "FILE IS FULL";:SO
     UND 110,10:PRINT:PRINT "YOU NEED TO
      EXPAND THE FILE IF YOU WISH TO CON
     TINUE"
3830 PRINT:PRINT "USE FILE MANAGER OPTIO
     N 4: ":PRINT " ** INCREASE FILE SIZE
     **":LOCATE 22,1:PRINT "PRESS ";CHR$
     (17);CHR$(217);" TO RETURN TO PROGR
     AM":PRINT:PRINT "PRESS [Esc] TO EXI
     T TO MAIN MENU"
3840 K$=INKEY$:IF K$="" THEN 3840 ELSE I
     F K$=CHR$(27) THEN RUN "ORGANIZE.BA
     S" ELSE IF K$=CHR$(13) THEN RESUME
     400
3850 RESTORE 3930:FOR Z=1 TO 11:READ A,A
     $:IF ERR=A THEN 3870
3860 NEXT:CLS:LOCATE 8,1:PRINT "HAVING T
     ROUBLE WITH LINE:";ERL:PRINT:PRINT
     "ERROR CODE";ERR:GOTO 3880
3870 CLS:LOCATE 8,1:PRINT A$:PRINT:PRINT
      "ERROR CODE:";ERR
3880 LOCATE 22,1:PRINT "PRESS ";CHR$(17)
     ;CHR$(217);" TO CONTINUE"
3890 K$=INKEY$:IF K$="" THEN 3890 ELSE C
     LEAR:GOTO 270
3900 '
3910 ' ERROR ROUTINE DATA
3920 '
3930 DATA 53,CAN'T LOCATE THAT FILE -- T
     RY AGAIN,61,DISK IS FULL -- MAKE RO
     OM BEFORE YOU CONTINUE TO ADD RE
     CORDS TO THE FILE,64,ILLEGAL FILE N
     AME -- DO NOT USE":PRINT "EXTENSION
     S. THIS PROGRAM CREATES IT'S OWN
     EXTENSIONS"
3940 DATA 67,TOO MANY FILES ON DISK -- O
     R BAD NAME,70,DISK IS WRITE PROTECT
     ED -- REMOVE WRITE PROTECT TAB OR U
     SE A DIFFERENT DISK,71,DISK IS NOT
     READY -- MAKE SURE DISK IS IN THE
     PROPER DRIVE,72,BAD DISK -- MEDIA E
     RROR -- CHANGE DISK
3950 DATA 75,PATH/FILE ACCESS ERROR -- T
     RY AGAIN,76,CAN'T FIND THAT PATH --
      TRY AGAIN,26,DEVICE IS NOT WORKING
      VERY WELL CHECK THE DEVICE
      FOR ANY PROBLEMS,52,PROBLEM WITH T
     HAT FILE -- TRY AGAIN-ff-
3960 IF MD=1 THEN LOCATE 24,1:COLOR 0,7:
     PRINT SPACE$(10);"-- OUTLINE EDITOR
     --";:COLOR 7,0:RETURN
3970 IF MD=2 THEN LOCATE 24,1:COLOR 0,7:
     PRINT SPACE$(12);"-- TEXT EDITOR --
     ";:COLOR 7,0:RETURN
3980 RETURN
3990 LOCATE 24,1:COLOR 0,7:PRINT SPACE$(
     12);"-- LINE EDITOR --";:
     COLOR 7,0:RETURN
4000 CLOSE:CLS:LOCATE 12,1:PRINT "PLACE
     PROGRAM DISK IN DRIVE 'A'":PRINT "P
     RESS ";CHR$(17);CHR$(217);" WHEN RE
     ADY"
4010 A$=INKEY$:IF A$="" THEN 4010 ELSE C
     LS:LOCATE 12,1:PRINT "LOADING MAIN
     MENU...":CLOSE:RUN "ORGANIZE.BAS"
```

HCM

The Organizer for the TI-99/4A

MAIN MENU

TI-99/4A

```
100 REM ****************
110 REM * THE ORGANIZER *
120 REM *    MAIN MENU   *
130 REM ****************
140 REM COPYRIGHT 1984,1985
150 REM EMERALD VALLEY PUBLISHING CO.
160 REM BY WILLIAM K. BALTHROP
170 REM HOME COMPUTER MAGAZINE
180 REM VERSION 5.1.1
190 REM TI EXTENDED BASIC
200 REM WITH 32K MEMORY EXPANSION
210 REM
220 REM USE "DSK1.ORGANIZE"
230 REM TO SAVE THIS PROGRAM
240 REM
250 FOR Z=1 TO 13 :: CALL COLOR(Z,16,5)
    :: NEXT Z :: CALL SCREEN(5)
260 DISPLAY AT(1,8)ERASE ALL:"THE ORGAN
    IZER":TAB(10);"MAIN MENU"
270 DISPLAY AT(5,1):"SELECT ONE:": :TAB
    (3);"1) OUTLINE EDITOR": :TAB(3);"2
    ) REPORTS": :TAB(3);"3) FILE MANAGE
    R": :TAB(3);"4) QUIT"
280 CALL KEY(0,K,S):: IF S=0 THEN 280 E
    LSE IF K<49 OR K>52 THEN 280 ELSE O
    N K-48 GOTO 290,310,330,350
290 DISPLAY AT(12,1)ERASE ALL:"LOADING
    OUTLINE EDITOR..." :: RUN "DSK1.OUT
    LINE"
300 GOTO 260
310 DISPLAY AT(12,1)ERASE ALL:"LOADING
    REPORTS GENERATOR...";:: RUN "DSK1.
    REPORTS"
320 GOTO 260
330 DISPLAY AT(12,1)ERASE ALL:"LOADING
    FILE MANAGER..." :: RUN "DSK1.FILEM
    GR"
340 GOTO 260
350 CALL CLEAR :: END
```

HCM

REPORTS

TI-99/4A

```
100 ! ****************
110 ! * DUMMY REPORTS *
120 ! *    PROGRAM     *
130 ! ****************
140 ! LOOK FOR REPORTS
150 ! PROGRAM IN VOLUME 5,
160 ! NO.2 OF
170 ! HOME COMPUTER MAGAZINE
180 !
190 ! USE "DSK1.REPORTS"
200 ! TO SAVE THIS PROGRAM
210 !
220 CALL CLEAR :: DISPLAY AT(8,1):"THE
    REPORTS PROGRAM WILL BE SUPPLIED IN
     VOLUME 5, NO.2 OF HOME COMPUTER M
    AGAZINE"
230 DISPLAY AT(14,1):"PRESS ENTER TO RE
    TURN TO THEMAIN MENU"
240 CALL KEY(0,K,S):: IF S=0 THEN 240 E
    LSE DISPLAY AT(12,1)ERASE ALL:"LOAD
    ING MAIN MENU..." :: RUN "DSK1.ORGA
    NIZE"
```

HCM

FILE MANAGER

TI-99/4A

```
100 REM ****************
110 REM * THE ORGANIZER *
120 REM * FILE MANAGER  *
130 REM ****************
140 REM COPYRIGHT 1984,1985
150 REM EMERALD VALLEY PUBLISHING CO.
160 REM BY WILLIAM K. BALTHROP
170 REM HOME COMPUTER MAGAZINE
180 REM VERSION 5.1.1
190 REM TI EXTENDED BASIC
200 REM WITH 32K MEMORY EXPANSION
210 REM AND DISK MEMORY SYSTEM
220 REM
230 REM USE "DSK1.FILEMGR"
240 REM TO SAVE THIS PROGRAM
250 REM
260 FOR Z=1 TO 13 :: CALL COLOR(Z,16,5)
    :: NEXT Z :: CALL SCREEN(5)
270 DEF STN(Q)=((ASC(SEG$(Z$,Q*2+1,1)))
    *256)+(ASC(SEG$(Z$,Q*2+2,1)))
280 DEF NTS$(Z)=CHR$(INT(Z/256))&CHR$((
    Z-INT(Z/256)*256))
290 CALL CLEAR :: GOSUB 730 :: DISPLAY
    AT(5,1):"SELECT ONE:": :"1) LIST
    FILE NAMES": :"2) CREATE NEW FILE
    "
300 DISPLAY AT(11,3):"3) DELETE A FILE"
    : :"4) INCREASE FILE SIZE": :"5
    ) EXIT TO MAIN MENU"
310 CALL KEY(0,K,S):: IF S=0 THEN 310 E
    LSE IF K<49 OR K>53 THEN 310 ELSE O
    N K-48 GOTO 320,440,560,600,720
320 GOSUB 730 :: DISPLAY AT(3,1):"WHICH
     DRIVE (1 2 OR 3):": :ACCEPT AT(3,
    24)SIZE(1)VALIDATE("123"):D$ :: IF
    D$="" THEN 290
330 D=VAL(D$):: OZ=1 :: OPEN #1:"DSK"&S
    TR$(D)&".",INTERNAL,RELATIVE,INPUT
340 INPUT #1:A$,A,B,C :: DISPLAY AT(3,1
    ):"DISK NAME: ";A$
350 FOR Z=OZ TO 19+OZ :: INPUT #1:A$,A,
    B,C :: IF A$="" THEN 410
360 FOR DB=1 TO (LEN(A$)-1):: IF SEG$(A
    $,DB,2)="_D" THEN 390
370 NEXT DB
380 GOTO 400
390 DISPLAY AT(Z+4-OZ,1):SEG$(A$,1,LEN(
    A$)-2):: DISPLAY AT(Z+4-OZ,12):"DAT
    A FILE";
400 NEXT Z
410 DISPLAY AT(24,1):"PRESS ENTER TO CO
    NTINUE";
420 CALL KEY(0,K,S):: IF S=0 THEN 420 E
    LSE IF A$<>"" THEN OZ=21 :: GOTO 35
    0
430 CLOSE #1 :: GOTO 290
440 GOSUB 730 :: DISPLAY AT(5,1):"ENTER
     FILE NAME:":"DSK1.FILE": :ACCEPT
    AT(6,4)SIZE(-10):F$ :: F$="DSK"&F$
450 DISPLAY AT(7,1):"HOW MANY RECORDS:"
    : :ACCEPT AT(7,18)SIZE(3)VALIDATE(
    DIGIT):NR
460 OPEN #1:SEG$(F$,1,5),INTERNAL,RELAT
    IVE,INPUT
470 INPUT #1:A$,A,B,C :: E$=SEG$(F$,6,L
    EN(F$)-5)&"_D"
```

Continued

FILE MANAGER *Continued*


```
480 INPUT #1:A$,A,B,C :: IF A$="" THEN CLOSE #1 :: GOTO 510 ELSE IF A$=E$ THEN 490 ELSE 480
490 DISPLAY AT(9,1):"FILE ALREADY EXISTS!" :: CALL SOUND(-1,440,0):: CLOSE #1
500 FOR OZ=1 TO 750 :: NEXT OZ :: GOTO 290
510 OPEN #1:F$&"_L",RELATIVE,INTERNAL,FIXED 11 :: OPEN #2:F$&"_D",RELATIVE,FIXED 27
520 A$=NTS$(NR)&NTS$(0)&NTS$(2)&NTS$(NR+1)&NTS$(0):: PRINT #1,REC 0:A$
530 A$=NTS$(0)&NTS$(0)&NTS$(0)&NTS$(2):: PRINT #1:A$ :: PRINT #2,REC 0:"" :: PRINT #2:""
540 FOR X=2 TO NR :: PRINT #1:RPT$(CHR$(0),6)&NTS$(X+1)&CHR$(0)&CHR$(0):: PRINT #2:"" :: NEXT X
550 PRINT #1:RPT$(CHR$(0),10):: PRINT #2:"" :: CLOSE #1 :: CLOSE #2 :: GOTO 290
560 GOSUB 730 :: DISPLAY AT(5,1):"DELETE WHICH FILE:":"DSK1.FILE" :: ACCEPT AT(6,4)SIZE(-10):F$ :: F$="DSK"&F$
570 DISPLAY AT(7,1):"SURE YOU WANT TO":"DELETE ";F$;" (Y/N)?N";:: ACCEPT AT(8,15+LEN(F$))SIZE(1)VALIDATE("YNyn"):OK$
580 IF OK$="N" OR OK$="n" THEN 290
590 DELETE F$&"_L" :: DELETE F$&"_D" :: GOTO 290
600 GOSUB 730 :: DISPLAY AT(5,1):"EXPAND WHICH FILE:":"DSK1.FILE" :: ACCEPT AT(6,4)SIZE(-10):F$ :: F$="DSK"&F$
610 OPEN #1:F$&"_L",RELATIVE,INTERNAL,FIXED 11 :: OPEN #2:F$&"_D",RELATIVE,FIXED 27
620 INPUT #1,REC 0:Z$ :: NR=STN(0):: NUSE=STN(1):: FEMP=STN(2):: LEMP=STN(3):: HFIR=STN(4)
630 DISPLAY AT(7,1):"THERE ARE";NR;" RECORDS": :"HOW MANY MORE DO YOU NEED"
640 ACCEPT AT(10,1)VALIDATE(DIGIT)SIZE(3):MR
650 IF MR+NR<501 THEN 670 ELSE DISPLAY AT(11,1)BEEP:"TOO MANY RECORDS"
660 FOR TD=1 TO 750 :: NEXT TD :: DISPLAY AT(11,1):"" :: GOTO 640
670 IF FEMP=0 THEN FEMP=NR+2 ELSE PRINT #1,REC LEMP:RPT$(CHR$(0),6)&NTS$(NR+2)&CHR$(0)&CHR$(0)
680 TR=NR+MR :: PRINT #1,REC 0:NTS$(TR)&NTS$(NUSE)&NTS$(FEMP)&NTS$(TR+1)&NTS$(HFIR)
690 FOR X=NR+2 TO TR :: PRINT #1,REC X:RPT$(CHR$(0),6)&NTS$(X+1)&CHR$(0)&CHR$(0):: PRINT #2,REC X:"" :: NEXT X
700 PRINT #1,REC TR+1:RPT$(CHR$(0),10):: PRINT #2,REC TR+1:""
710 CLOSE #1 :: CLOSE #2 :: GOTO 290
720 DISPLAY AT(12,1)ERASE ALL:"LOADING MAIN MENU..." :: RUN "DSK1.ORGANIZE"
730 DISPLAY AT(1,8)ERASE ALL:"THE ORGANIZER":TAB(8);"FILE MANAGER" :: RETURN
```

HCM

OUTLINE EDITOR

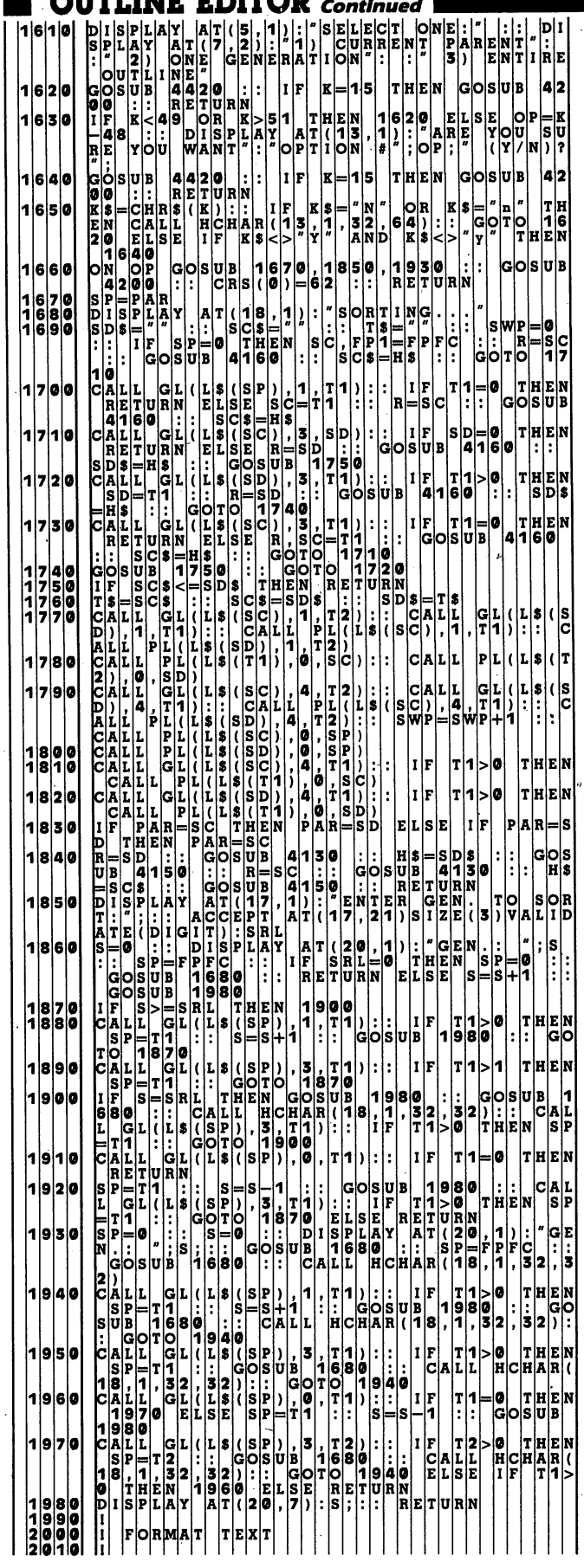

```
100 ! ******************
110 ! *  THE ORGANIZER    *
120 ! * OUTLINE EDITOR  *
130 ! ******************
140 ! COPYRIGHT 1984,1985
150 ! EMERALD VALLEY PUBLISHING CO.
160 ! BY WILLIAM K. BALTHROP
170 ! HOME COMPUTER MAGAZINE
180 ! VERSION 5.1.1
190 ! TI EXTENDED BASIC
200 ! WITH 32K MEMORY EXPANSION
210 ! AND DISK MEMORY SYSTEM
220 !
230 ! USE "DSK1.OUTLINE"
240 ! TO SAVE THIS PROGRAM
250 !
260 CALL INIT :: CALL LOAD(-31878,0):: CALL CLEAR
270 DIM L$(501),SCR$(21),SCR(21),MD$(2):: RESTORE 4680 :: READ MD$(0),MD$(1),MD$(2)
280 SL$="LVL### USE### FRE### R## C##" :: CRS(0)=95
290 DEF NTS$(Z)=CHR$(INT(Z/256))&CHR$(Z-(INT(Z/256)*256))
300 DEF STN(Q)=((ASC(SEG$(Z$,Q*2+1,1)))*256)+(ASC(SEG$(Z$,Q*2+2,1)))
310 FOR Z=1 TO 14 :: CALL COLOR(Z,16,5):: NEXT Z :: CALL SCREEN(5)
320 !
330 ! LOAD FILE LINKS
340 !
350 GOSUB 4050 :: DISPLAY AT(5,1):"ENTER FILE NAME:DSK1.FILE" :: ACCEPT AT(5,20)SIZE(-10):F$ :: IF F$="" THEN 3040
360 OPEN #1:"DSK"&F$&"_L",RELATIVE,INTERNAL,FIXED 11
370 OPEN #2:"DSK"&F$&"_D",RELATIVE,FIXED 27 :: F$=SEG$(F$,3,LEN(F$)-2)
380 DISPLAY AT(8,1):"READING #";:: GOSUB 4090 :: FOR R=2 TO NR+1 :: INPUT #1:L$(R):: DISPLAY AT(8,11):R-1;:: NEXT R
390 PAR=0 :: LCP=1 :: HD$=F$ :: CC,SFC=FPFC
400 !
410 ! OUTLINE EDITOR
420 ! MAIN CONTROL ROUTINE
430 !
440 MD=1 :: MDD=0 :: GOSUB 4200
450 IF SCR(LCP)=0 THEN CX=3 :: GOSUB 4460 :: MD,MDD=1 :: GOSUB 4390 :: GOSUB 3350 ELSE GOTO 470
460 H$=SCR$(LCP):: GOSUB 4150 :: MDD=0 :: GOSUB 4380
470 GOSUB 4380 :: CRS(0)=62 :: CRS(1)=32
480 GOSUB 4400 :: IF K<16 THEN ON K GOSUB 2880,510,510,510,510,2970,510,510,510,3150,3110,2390,3150,1600,3040 :: GOTO 450
490 IF K>176 AND K<184 THEN K=K-176 :: ON K GOSUB 550,860,510,510,510,2390,1030 :: GOTO 450 ELSE IF K=158 THEN GOSUB 3010 :: GOTO 450
500 IF K=60 THEN GOSUB 4620 :: GOTO 450 ELSE IF K=62 THEN GOSUB 4630 :: GOTO 450 ELSE 480
510 RETURN
520 !
530 ! DELETE LINE
540 !
550 DR=SCR(LCP):: CALL GL(L$(DR),1,CH):: CALL GL(L$(DR),2,PR):: CALL GL(L$(DR),3,FR):: CALL GL(L$(DR),4,TX)
560 IF MD=1 THEN IF CH>0 THEN GOSUB 800 :: GOTO 580
570 IF TX>0 THEN T1=TX :: CALL PL(L$(DR),4,0):: GOSUB 2760
580 IF PR<>0 OR FR<>0 OR MD<>1 OR PAR<>0 THEN 600 ELSE SCR(1)=0 :: GOSUB 820
590 L$(DR)=RPT$(CHR$(0),10):: R=DR :: GOSUB 4130 :: GOSUB 4110 :: IF CH>0 THEN RETURN ELSE NUSE=NUSE-1 :: RETURN
600 IF PR<>0 OR FR<>0 OR MD<>1 OR PAR=0 THEN 640 ELSE CALL PL(L$(PAR),1,0):: R=PAR :: GOSUB 4130 :: CC=PAR :: CALL GL(L$(PAR),0,PAR)
610 LEV=LEV-1 :: SFC=0 :: GOSUB 820 :: GOSUB 4110 :: CALL GL(L$(DR),4,T1)
620 IF T1>0 THEN TF=HFIR :: TL=HLST :: HFIR,HLST=DR :: GOSUB 2660 :: HFIR=TF :: HLST=TL
630 L$(DR)=RPT$(CHR$(0),10):: R=DR :: GOSUB 4130 :: NUSE=NUSE-1 :: GOSUB 4110 :: GOSUB 4200 :: RETURN
640 IF PR<>0 OR FR<>0 OR MD<>2 THEN 670 ELSE SCR(LCP)=0 :: SCR$(LCP)="" :: GOSUB 820
650 L$(D)=RPT$(CHR$(0),10):: R=DR :: GOSUB 4130 :: H$="" :: GOSUB 4150 :: NUSE=NUSE-1 :: GOSUB 4110 :: GOSUB 3200 :: GOSUB 4570
660 RETURN
670 IF PR>0 THEN CALL PL(L$(PR),3,FR):: R=PR :: GOSUB 4130 :: GOTO 760
680 IF PAR=0 AND MD=1 THEN FPFC,SFC,R=FR :: CALL PL(L$(FR),2,0):: GOSUB 4130 :: GOSUB 4110 :: GOTO 710
690 IF MD=1 THEN CALL PL(L$(PAR),1,FR):: SFC=FR :: R=PAR :: GOSUB 4130 :: CALL PL(L$(FR),2,0):: R=FR :: GOSUB 4130
700 IF MD=2 THEN CALL PL(L$(TP),4,FR):: R=TP :: GOSUB 4130 :: SFTR,R=FR :: CALL PL(L$(FR),2,0):: GOSUB 4130
710 GOSUB 820 :: NUSE=NUSE-1 :: GOSUB 4110 :: CALL PL(L$(DR),3,0):: R=DR :: GOSUB 4130
720 FOR Z=LCP TO 20 :: SCR(Z)=SCR(Z+1):: SCR$(Z)=SCR$(Z+1):: NEXT Z
```

Continued

OUTLINE EDITOR *Continued*

Continued

OUTLINE EDITOR *Continued*



Continued

OUTLINE EDITOR *Continued*

Continued

OUTLINE EDITOR *Continued*

```
3450 IF K=15 THEN RETURN
3460 IF K<15 THEN ON K GOSUB 2880,510,3560,3630,510,510,3760,3490,3500,3910,3950,2390,3480,510 :: GOTO 3360
3470 IF K>176 AND K<184 THEN K=K-176 :: ON K GOSUB 550,860,510,510,2020,2390,1030 :: GOTO 3360 ELSE 3360
3480 CALL HCHAR(LCP+2,CX+2,CRS(1)):: GOSUB 3910 :: CX=3 :: CALL GCHAR(LCP+2,CX+2,CRS(1)):: RETURN
3490 IF CX=3 THEN RETURN ELSE CALL HCHAR(LCP+2,CX+2,CRS(1)):: CX=CX-1 :: CALL GCHAR(LCP+2,CX+2,CRS(1)):: RETURN
3500 IF CX=28 THEN CALL HCHAR(LCP+2,CX+2,CRS(1)):: CX=3 :: GOSUB 3910 :: CALL GCHAR(LCP+2,CX+2,CRS(1)):: RETURN
3510 IF CX-1>LEN(SCR$(LCP))THEN SCR$(LCP)=SCR$(LCP)&CHR$(32)
3520 CALL HCHAR(LCP+2,CX+2,CRS(1)):: CX=CX+1 :: CALL GCHAR(LCP+2,CX+2,CRS(1)):: RETURN
3530 !
3540 ! DELETE CHARACTER
3550 !
3560 MU=0 :: IF CX>3 AND CX<29 THEN SCR$(LCP)=SEG$(SCR$(LCP),1,CX-3)&SEG$(SCR$(LCP),CX-1,26-(CX-2))&" " ELSE 3580
3570 DISPLAY AT(LCP+2,3):SCR$(LCP);:: CALL GCHAR(LCP+2,CX+2,CH):: CRS(1)=CH :: RETURN
3580 IF CX=3 THEN SCR$(LCP)=SEG$(SCR$(LCP),2,25)&" " :: GOTO 3570
3590 SCR$(LCP)=SEG$(SCR$(LCP),1,25)&" " :: GOTO 3570
3600 !
3610 ! INSERT CHARACTER
3620 !
3630 IF CX>3 AND CX<29 THEN SCR$(LCP)=SEG$(SCR$(LCP),1,CX-3)&" "&SEG$(SCR$(LCP),CX-2,26-(CX-3)):: GOTO 3660
3640 IF CX=3 THEN SCR$(LCP)=" "&SCR$(LCP):: GOTO 3660
3650 RETURN
3660 IF LEN(SCR$(LCP))<27 THEN 3720
3670 IF SEG$(SCR$(LCP),LEN(SCR$(LCP)),1)=" " THEN SCR$(LCP)=SEG$(SCR$(LCP),1,LEN(SCR$(LCP))-1):: GOTO 3720
3680 FOR Z=LEN(SCR$(LCP))-1 TO 1 STEP -1
3690 IF SEG$(SCR$(LCP),Z,1)=" " THEN WD$=SEG$(SCR$(LCP),Z+1,27-Z):: SCR$(LCP)=SEG$(SCR$(LCP),1,Z-1):: GOTO 3710
3700 NEXT Z :: WD$=SEG$(SCR$(LCP),27,1):: SCR$(LCP)=SEG$(SCR$(LCP),1,26)
3710 TCX=CX :: TLCP=LCP :: DISPLAY AT(LCP+2,3):SCR$(LCP):: GOSUB 3910 :: GOSUB 860 :: SCR$(LCP)=WD$ :: MU=1
3720 DISPLAY AT(LCP+2,3):SCR$(LCP);:: CRS(1)=32 :: IF MU=1 THEN CX=TCX :: LCP=TLCP :: RETURN ELSE RETURN
3730 !
3740 ! ERASE LINE
3750 !
3760 SCR$(LCP)="" :: CALL HCHAR(LCP+2,5,32,26):: CX=3 :: CRS(1)=32 :: RETURN
3770 !
3780 ! WORD WRAP
3790 !
3800 IF K=32 OR POS(SCR$(LCP),CHR$(32),1)=0 THEN 3880
3810 IF SEG$(SCR$(LCP),26,1)=" " THEN WD$=CHR$(K):: GOTO 3850
3820 FOR Z=LEN(SCR$(LCP))TO 1 STEP -1 :: IF SEG$(SCR$(LCP),Z,1)=" " THEN 3840
3830 NEXT Z :: WD$=CHR$(K):: GOTO 3850
3840 WD$=SEG$(SCR$(LCP),Z+1,LEN(SCR$(LCP))-Z)&CHR$(K):: SCR$(LCP)=SEG$(SCR$(LCP),1,Z-1)
3850 GOSUB 4030 :: GOSUB 3910 :: IF DCNR=1 THEN SCR$(LCP)=WD$ :: DISPLAY AT(LCP+2,3)SIZE(26):WD$ :: CX=LEN(WD$)+3 :: GOTO 3870
3860 GOSUB 950 :: SCR$(LCP)=WD$ :: DISPLAY AT(LCP+2,3)SIZE(26):SCR$(LCP);:: CX=LEN(WD$)+3
3870 GOSUB 4380 :: RETURN
3880 GOSUB 3910 :: CX=3 :: IF SCR$(LCP)<>"" THEN GOSUB 860
3890 IF K<>32 THEN SCR$(LCP)=CHR$(K):: DISPLAY AT(LCP+2,3)SIZE(26):SCR$(LCP);:: CX=4
3900 GOSUB 4390 :: RETURN
3910 IF LCP<21 THEN 3930 ELSE GOSUB 3290 :: LCP=10 :: SFTR=SCR(12)
3920 FOR Z=1 TO 21 :: SCR$(Z)="" :: SCR(Z)=0 :: NEXT Z :: GOSUB 3200
3930 CALL HCHAR(LCP+2,CX+2,CRS(1)):: LCP=LCP+1 :: IF SCR(LCP)=0 THEN GOSUB 4570 :: SCR(LCP)=R :: DCNR=1 ELSE DCNR=0
3940 CALL GCHAR(LCP+2,CX+2,CRS(1)):: RETURN
3950 IF LCP>1 THEN 3990 ELSE CALL GL(L$(SCR(1)),2,PR):: IF PR=0 THEN RETURN ELSE GOSUB 3290
3960 FOR Z=1 TO 11 :: CALL GL(L$(PR),2,TR):: IF TR=0 THEN 3980 ELSE PR=TR
3970 NEXT Z
3980 SFTR=PR :: SCR(1)=PR :: GOSUB 3200 :: LCP=1 :: RETURN
3990 CALL HCHAR(LCP+2,CX+2,CRS(1)):: LCP=LCP-1 :: CALL GCHAR(LCP+2,CX+2,CRS(1)):: RETURN
4000 !
4010 ! MISC SUBROUTINES
4020 !
4030 DISPLAY AT(LCP+2,3)SIZE(26):SCR$(LCP);:: RETURN
4040 ACCEPT AT(LCP+2,3)SIZE(-26):SCR$(LCP):: RETURN
4050 DISPLAY AT(1,8)ERASE ALL:"THE ORGANIZER":TAB(8);"OUTLINE EDITOR" :: RETURN
4060 !
4070 ! DISK OPERATIONS
4080 !
4090 INPUT #1,REC 0:Z$ :: NR=STN(0):: NUSE=STN(1):: FEMP=STN(2):: LEMP=STN(3):: HFIR=STN(4)
4100 INPUT #1,REC 1:Z$ :: HLST=STN(0):: THFIR=STN(1):: THLST=STN(2):: FPFC=STN(3):: RETURN
4110 A$=NTS$(NR)&NTS$(NUSE)&NTS$(FEMP)&NTS$(LEMP)&NTS$(HFIR):: PRINT #1,REC 0:A$
4120 A$=NTS$(HLST)&NTS$(THFIR)&NTS$(THLST)&NTS$(FPFC):: PRINT #1,REC 1:A$ :: RETURN
4130 PRINT #1,REC R:L$(R):: RETURN
4140 INPUT #1,REC R:L$(R):: RETURN
4150 PRINT #2,REC R:H$ :: RETURN
4160 INPUT #2,REC R:H$ :: RETURN
4170 !
4180 ! LOAD OUTLINE SCREEN
4190 !
4200 IF PAR>0 THEN R=PAR :: GOSUB 4160 :: HD$=H$ ELSE HD$=F$ :: LCP=1 :: IF NUSE=0 THEN Z=1 :: GOTO 4250
4210 IF SFC=0 THEN IF PAR=0 THEN SFC=FPFC ELSE CALL GL(L$(PAR),1,SFC)
4220 IF SFC=0 THEN Z=1 :: GOTO 4250
4230 R=SFC :: FOR Z=1 TO 21 :: GOSUB 4160 :: SCR(Z)=R :: SCR$(Z)=H$ :: CALL GL(L$(R),3,R):: IF R=0 THEN Z=Z+1 :: GOTO 4250
4240 NEXT Z :: GOTO 4260
4250 FOR Z=Z TO 21 :: SCR(Z)=0 :: SCR$(Z)="" :: NEXT Z
4260 GOSUB 4370 :: FOR Z=1 TO 21 :: IF SCR(Z)>0 THEN DISPLAY AT(Z+2,3)SIZE(26):SCR$(Z);
4270 NEXT Z :: FOR Z=1 TO 21 :: IF SCR(Z)=CC THEN LCP=Z :: DISPLAY AT(LCP+2,1)SIZE(1):">";:: GOTO 4290
4280 NEXT Z :: LCP=1 :: CC=1 :: DISPLAY AT(LCP+2,1)SIZE(1):">";
4290 FOR Z=1 TO 21 :: X=SCR(Z):: IF X=0 THEN RETURN
4300 CALL GL(L$(X),1,T1):: CALL GL(L$(X),4,T2):: IF T1>0 AND T2>0 THEN C$="*" :: GOTO 4330
4310 CALL GL(L$(X),1,T1):: IF T1>0 THEN C$="#" :: GOTO 4330
4320 CALL GL(L$(X),4,T1):: IF T1>0 THEN C$="&" ELSE C$=""
4330 DISPLAY AT(Z+2,2)SIZE(1):C$;:: NEXT Z :: RETURN
4340 !
4350 ! SET UP SCREEN
4360 !
4370 CALL CLEAR :: GOSUB 4380 :: DISPLAY AT(2,1)SIZE(28):"<";HD$;:: RETURN
4380 DISPLAY AT(24,1):MD$(MDD);
4390 DISPLAY AT(1,1)SIZE(28):USING SL$:LEV,NUSE,NR-NUSE,LCP,CX-2 :: RETURN
4400 CALL KEY(0,K,S):: IF S=0 THEN Q=ABS(Q-1):: CALL HCHAR(LCP+2,3,CRS(Q)):: GOTO 4400 ELSE CALL HCHAR(LCP+2,3,CRS(1))
4410 CALL SOUND(-1,880,0):: RETURN
4420 CALL KEY(0,K,S):: IF S=0 THEN 4420 ELSE RETURN
4430 !
4440 ! GET NEXT OUTLINE RECORD
4450 !
4460 IF FEMP=0 THEN 4660 ELSE R=FEMP :: NUSE=NUSE+1 :: CALL GL(L$(FEMP),3,FEMP):: SCR(LCP)=R
4470 IF FEMP=0 THEN LEMP=0 :: GOSUB 4110 ELSE GOSUB 4110
4480 L$(R)=RPT$(CHR$(0),10):: IF R=FPFC THEN GOSUB 4130 :: RETURN
4490 IF LCP>1 THEN PR=SCR(LCP-1)ELSE 4520
```

Continued

OUTLINE EDITOR *Continued*

TI-99/4A

```
4500 IF PAR=0 THEN CALL PL(LS(R),2,PR)::
     GOTO 4130 ::: CALL PL(LS(PR),3,R)::
     :: GOTO 4130 ::: R=PR :: GOSUB 4130
     :: R=TR :: RETURN
4510 CALL PL(LS(R),0,PAR)::: CALL PL(LS(R
     ),2,PR)::: GOSUB 4130 ::: CALL PL(LS(
     PR),3,R)::: GOSUB 4130 ::: R=PR :: GO
     SUB 4130 :: R=TR :: RETURN
4520 IF PAR=0 THEN R=LS(R)&SC$(CHR$(0),10
     ):: GOSUB 4130 :: GOTO 4110
4530 CALL PL(LS(R),0,PAR)::: GOSUB 4130 :
     :: R=TR :: RETURN
4540
4550 GET NEXT TEXT RECORD
4560
4570 IF FEMP=0 THEN 4660 ELSE R=FEMP :::
     NUSE=NUSE+1 :: CALL GL(LS(FEMP),3,F
     EMPS)::: IF FEMP=0 THEN LEMP=0
4580 SUB 4110 ::: LS(R)=R$ :: RETURN
4590
4600 IF FLAG=0 THEN CALL PL(LS(TP),4,R)::
     CALL PL(LS(R),0,TP)::: GOSUB 4130 ::
     R=TP :: RETURN
4610 PR=SCRN-1)::: CALL PL(LS(R),2,PR)
     ::: CALL PL(LS(PR),3,R)::: GOSUB 413
     0 :: R=PR :: RETURN
4620 IF PAR=0 THEN RETURN ELSE CC=PAR :::
     CALL GL(LS(PAR),0,PAR)::: LEV=LEV-1
     :: LCP=1 :: GOSUB 4200
```

```
4630 PAR=SCR(LCP)::: CALL GL(LS(PAR),1,CC
     )::: LEV=LEV+1 ::: SFC=0 :: LCP=1 ::
     GOSUB 4200 :: RETURN
4640 DATA 60,62,158,178,181,182
4650 REM OUT OF RECORDS MESSAGE
4660 DISPLAY AT(24,1):"OUT OF SPACE IN FI
     LE" :: CALL SOUND(-1,440,0)
4670 FOR Z=1 TO 500 :: NEXT Z :: GOTO 39
     0
4680 DATA ---OUTLINE EDITOR---,---LINE ED
     ITOR---,---TEXT EDITOR---
4690 SUB GL(LS,Y,Z)
4700 Z=(ASC(LS,Y*2+1,1))*256)+ASC(SEG$(LS
     ,Y*2+2,1))
4710 SUBEND
4720 SUB PL(LS,Y,Z)::: ON Y+1 GOTO 4730,4
     740,4750,4760,4770
4730 LS=CHR$(INT(Z/256))&CHR$(Z-(INT(Z/2
     56)*256))&SEG$(LS,3,8)::: SUBEXIT
4740 LS=SEG$(LS,1,2)&CHR$((INT(Z/256))&CH
     R$(Z-(INT(Z/256)*256)))&SEG$(LS,5,6)
     ::: SUBEXIT
4750 LS=SEG$(LS,1,4)&CHR$((INT(Z/256))&CH
     R$(Z-(INT(Z/256)*256)))&SEG$(LS,7,4)
     ::: SUBEXIT
4760 LS=SEG$(LS,1,6)&CHR$((INT(Z/256))&CH
     R$(Z-(INT(Z/256)*256)))&SEG$(LS,9,2)
     ::: SUBEXIT
4770 LS=SEG$(LS,1,8)&CHR$((INT(Z/256))&CH
     R$(Z-(INT(Z/256)*256))):::: SUBEXIT
4780 SUBEND
```

HCM

The Organizer for the APPLE *II* Family

MAIN MENU

APPLE *II* Family

```
100  REM ********************
110  REM * THE ORGANIZER *
120  REM *    MAIN MENU    *
130  REM ********************
140  REM COPYRIGHT 1984, 1985
150  REM EVERALL DIGITAL VALLEY PUBLISHING CO.
160  REM BY WILLIAM K. BALTHROP
170  REM AND THE HCM STAFF
180  REM VERSION 5.1.1
190  REM APPLESOFT II FAMILY APPLESOFT
200  REM ASAVE TO PEEK (48905) = 76 "ORGANIZE" AND
210  PEEK (48911) = 0 THEN PD = 1
220  IF PD THEN 240
230  PRINT CHR$ (4) "MAXFILES 1"
240  POSUB 3200
250  GOSUB 4600
260  GOSUB 4600
270  IF VAL (INS$)) = "QUIT" THEN HOME :
     MESSAGE : GOTO 300
280  GOSUB 7200
290  GOTO 250
300  END
310  REM INITIALIZATION
320  DIM SC$(10)
330  BLOAD CHR$ (7)
340  INS = ""
350  READ INS : IF INS = "*" THEN RETURN
360  SC = SC + 1
370  SC$(SC) = INS
380  GOTO 350
390  DATA "OUTLINE"
400  DATA "REPORTS"
410  DATA "FILEMGR"
420  DATA "QUIT"
430  DATA "*"
440  RETURN
450  REM MENU DISPLAY
460  HOME
470  HOME
480  VTAB 12
490  INVERSE
500  PRINT "THE ORGANIZER"
510  NORMAL
520  VTAB 15
530  PRINT "MAIN MENU"
540  FOR I = 1 TO SC
550  VTAB 2 + I : PRINT I ". " ; SC$(I)
560  PRINT
570  NEXT
580  RETURN
590  REM GET A NUMBER CHOICE
600  VTAB 27 : HTAB 12
610  GET INS : IF INS = "" THEN 610
620  IF INS < "1" AND INS < > "" THEN CHR$(
     48 + SC) : GOTO 640
```

```
630  PRINT INS; : GOTO 600
640  PRINT INS
650  VTAB 2 : VAL (INS) + 5 : HTAB 7
660  PRINT SE INS". "; SC$(VAL (INS))
670  NORMAL
680  FOR DI = 1 TO 1000 : NEXT
690  RETURN
700  REM BRANCH TO APPROPRIATE PROGRAM
710  BX$ = SC$(VAL (INS))
720  IF PD THEN GOSUB 770
730  HOME : VTAB 10 : HTAB 1 : PRINT "LOAD
     "; BX$;"PROGRAM" : RUN : BX$
740  PRINT CHR$ (4) "RUN ";BX$
750  RETURN
760  VTAB 1 : VTAB 22 : PRINT "PLACE DISK
     AND PRESS RETURN"
770  GOSUB 810 : IF KB < > (141) THEN 78
     0
780  PRINT CHR$ (4);"PREFIX,D1"
790  RETURN
800  KB = PEEK (-16384) : IF KB > 127
     THEN POKE -16368,0 : RETURN
810  GOTO 810
820  DR$ : GOTO 820
```

HCM

REPORTS

APPLE *II* Family

```
100  REM ********************
110  REM * DUMMY REPORTS *
120  REM *    PROGRAM     *
130  REM ********************
140  REM THE REPORTS PROGRAM WILL BE
150  REM SUPPLIED WITH VOL. 5 NO. 2 OF
160  REM HOME COMPUTER MAGAZINE
170  REM
180  REM SAVE THIS PROGRAM WITH THE FILE
     NAME "REPORTS"
190  REM
200  REM
210  HOME : VTAB 8: PRINT "THE REPORTS PR
     OGRAM WILL BE SUPPLIED WITH ISSUE 5.
     2 OF": PRINT "HOME COMPUTER MAGAZINE
     "
220  VTAB 14: PRINT "PRESS RETURN TO RET
     URN TO MAIN MENU"
230  GET AS: IF AS < > CHR$ (13) THEN
     230
240  PRINT CHR$ (4);"RUN ORGANIZE,D1"
     END
```

HCM

Continued

FILE MANAGER *Continued*


```
1900 IF I$ = "" THEN 1950
1910 IF LEN (I$) > 1 THEN I$ = LEFT$ (I$,1)
     IF ASC (I$) < ASC ("A") OR ASC (I$) > ASC ("Z") THEN 1900
1920 IF I$ = "D" OR I$ = "I" OR I$ = "L" OR I$ = "R" OR I$ = "S" OR I$ = "T" THEN 1940
1930 IF I$ = "Q" THEN END
     GOTO 1900
1940 ON I GOSUB 2000,2500,3000,4000,5000,6000,7000
1950 GOTO 1900
1960 REM
1970 REM
1980 REM
1990 REM
2000 REM
2010 REM
2020 REM
2030 REM
2040 REM
2050 REM
2060 REM
2070 REM
2080 REM
2090 REM
2100 REM
2110 REM
2120 REM
2130 REM
2140 REM
2150 REM
2160 REM
2170 REM
2180 REM
2190 REM
2200 REM
2210 REM
2220 REM
2230 REM
2240 REM
2250 REM
2260 REM
2270 REM
2280 REM
2290 REM
2300 REM
2310 REM
2320 REM
2330 REM
2340 REM
2350 REM
2360 REM
2370 REM
2380 REM
2390 REM
2400 REM
2410 REM
2420 REM
2430 REM
2440 REM
2450 REM
2460 REM
2470 REM
2480 REM
2490 REM
2500 REM
2510 REM
2520 REM
2530 REM
2540 REM
2550 REM
2560 REM
2570 REM
2580 REM
2590 REM
2600 REM
2610 REM
2620 REM
2630 REM
2640 REM
2650 REM
2660 REM
2670 REM
2680 REM
2690 REM
2700 REM
2710 REM
2720 REM
2730 REM
2740 REM
2750 REM
2760 REM
2770 REM
2780 REM
2790 REM
2800 REM
2810 REM
2820 REM
2830 REM
2840 REM
2850 REM
2860 REM
2870 REM
2880 REM
2890 REM
2900 REM
2910 REM
2920 REM
2930 REM
2940 REM
2950 REM
2960 REM
2970 REM
2980 REM
2990 REM
3000 REM
3010 REM
3020 REM
```

HCM

OUTLINE EDITOR


```
100 REM
110 REM
120 REM
130 REM
140 REM
150 REM
160 REM
170 REM
180 REM
190 REM
200 REM
210 REM
220 REM
230 REM
240 REM
250 REM
260 REM
270 REM
280 REM
290 REM
300 REM
310 REM
320 REM
330 REM
340 REM
350 REM
360 REM
370 REM
380 REM
390 REM
400 REM
410 REM
420 REM
430 REM
440 REM
450 REM
460 REM
470 REM
480 REM
490 REM
500 REM
510 REM
```

Continued

```
520 BL$ = CHR$ (7):A$ = "A":Z$ = "Z"
530 LF$ = CHR$ (8):RT$ = CHR$ (21):RB$ = CHR$ (13):ESC$ = CHR$ (27)
540 QZ$ = "0000"
550 DIM KB(2):KB(1) = - 16384:KB(2) = - 16368
560 ZQ = 0:ZL = 0:ZM = 0
570 DIM FP%(6)
580 DS% = 50:FR$ = "": FOR IT = 1 TO DS% - 5:FR$ = FR$ + "*": NEXT
590 LB$ = " "
600 DIM SC$(DU%): DIM SC%(DU%,7)
610 DIM BF$(DU%): DIM BF%(DU%,7)
620 DIM HD$(DU%): DIM HD%(DU%,7)
630 DIM HK%(DU%)
640 LV% = 0
650 DIM PG%(5)
660 DIM GS%(7)
670 DIM MD%(2)
680 DIM LW%(2)
690 DIM HS%(DU%): DIM HI$(4)
700 FOR IT = 1 TO 21: READ X$: IF IT = 6 OR IT = 7 THEN 720
710 LE$(IT) = CHR$ ( ASC (X$) - 64): GOTO 730
720 LE$(IT) = X$
730 NEXT
740 HI$(1) = LE$(10):HI$(2) = LE$(11):HI$(3) = LE$(21):HI$(4) = ESC$
750 GOTO 780
760 :
770 DATA "A","Z","B","H","U","<",">","T","R","K","J","E","Q","L","G","X","P","H","D","F","M"
780 RETURN
790 REM HEADING
800 VTAB 1: HTAB 1: INVERSE : PRINT " THE ORGANIZER: ": NORMAL
810 VTAB 1: HTAB 18: PRINT "OUTLINE EDITOR"
820 VTAB 24: HTAB 5
830 PRINT "PRESS <ESC> TO LEAVE";
840 RETURN
850 REM CONTROL LEVEL PROMPTS
860 ON MD%(0) GOSUB 880,920,940,960
870 RETURN
880 GOSUB 1290: GOSUB 1300
890 GOTO 910
900 INVERSE : PRINT "O";: NORMAL : PRINT "UTLINE, ";: INVERSE : PRINT "T";: NORMAL : PRINT "EXT, ";: INVERSE : PRINT "S";: NORMAL : PRINT "ORT, LINE CURSOR";
910 RETURN
920 GOSUB 1300
930 ON MD%(1) GOSUB 1040,1050,1060: RETURN
940 GOSUB 1290: ON MD%(1) GOSUB 1040,1050,1060
950 GOSUB 1300: GOSUB 990: RETURN
960 GOSUB 1290: ON MD%(1) GOSUB 1040,1050,1060
970 IF MD%(1) < 3 THEN GOSUB 1300: ON MD%(1) GOSUB 1090,1110
980 RETURN
990 PRINT "CTRL KEY: ";
1000 ON MD%(1) GOTO 1010,1030
1010 INVERSE : PRINT "<";: NORMAL : PRINT ",";: INVERSE : PRINT ">";: NORMAL : PRINT ",";
1020 PRINT "T,R,K,J,E,Q,L,G,X,P,H,D": RETURN
1030 PRINT "K,J,E,Q,L,G,X,P,H,D,F": RETURN
1040 PRINT "OUTLINE EDIT": GOSUB 1070: RETURN
1050 PRINT "TEXT EDITOR": GOSUB 1070: RETURN
1060 PRINT "SORT": RETURN
1070 VTAB 1: HTAB 27: PRINT "ROW ";SC%;: HTAB 34: PRINT "COL 0": RETURN
1080 GOSUB 1300: ON MD%(1) GOTO 1090,1110
1090 ON MD%(2) GOTO 1260,1270,1130,1140,1150,1160,1170,1180,1190,1200,1210,1220,1230,1240
1100 STOP
1110 ON MD%(2) GOTO 1150,1160,1170,1180,1190,1200,1210,1220,1230,1240,1250
1120 STOP
1130 PRINT "TEXT EDITOR": RETURN
1140 PRINT "SORT": RETURN
1150 PRINT "POINTER UP": RETURN
1160 PRINT "POINTER DOWN": RETURN
1170 PRINT "LINE EDIT": RETURN
1180 PRINT "NUMBER OF LINES TO SCROLL: "SL%: RETURN
1190 PRINT "INSERT LINE": RETURN
1200 PRINT "GRAB LINE": RETURN
1210 PRINT "COPY LINE": RETURN
1220 PRINT "PASTE LINE": RETURN
1230 PRINT "HOLD BUFFER DISPLAY": RETURN
1240 PRINT "DELETE LINE": RETURN
1250 PRINT "FORMAT TEXT": RETURN
1260 PRINT "GO BACK A GENERATION": RETURN
1270 PRINT "GO FORWARD A GENERATION": RETURN
1280 HTAB 1: VTAB 23: PRINT "DISK BUSY, PLEASE WAIT";: RETURN
1290 VTAB 1: HTAB 11: CALL - 868: RETURN
1300 VTAB 23: HTAB 1: CALL - 868: RETURN
1310 REM OUTERMOST LEVEL
1320 MD%(0) = 2: RETURN
1330 REM SECOND MODE LEVEL STEERING ROUTINE
1340 ON MD%(1) GOTO 1360,1370,1460
1350 STOP
1360 MD%(0) = 3: RETURN
1370 GOSUB 5970:MD%(0) = 3
1380 IF GS%(7) = 0 AND FP%(3) = 0 THEN MD%(0) = 1:MD%(1) = 1: RETURN
1390 IF ZM = 1 THEN MD%(0) = 4:MD%(2) = 3:ZM = 0
1400 GOSUB 860
1410 ON MD%(0) - 2 GOSUB 1500,1590
1420 IF MD%(0) > = 3 AND SC%(SC%,6) = 1 THEN 1400
1430 IF MD%(1) < > 1 THEN GOSUB 5690:MD%(1) = 1
1440 MD%(0) = 1:MD%(1) = 1: GOSUB 7160
1450 RETURN
1460 GOSUB 1730
1470 MD%(0) = 1:MD%(1) = 1
1480 RETURN
1490 REM OUTLINE ENTRY OPTIONS
1500 VC = TM% + SC%:HC = 1: GOSUB 9180: GOSUB 7070
1510 IF IN$ = ESC$ THEN MD%(0) = 0: GOSUB 1690: RETURN
1520 ON MD%(1) GOTO 1530,1550
1530 FOR CX = 6 TO 19: IF IN$ = LE$(CX) THEN CX% = CX - 5: GOTO 1570
1540 NEXT : PRINT BL$;: GOTO 1500
1550 FOR CX = 10 TO 20: IF IN$ = LE$(CX) THEN CX% = CX - 9: GOTO 1570
1560 NEXT : PRINT BL$;: GOTO 1500
1570 MD%(2) = CX%
1580 MD%(0) = 4: RETURN
1590 ON MD%(1) GOTO 1600,1630
1600 ON MD%(2) GOSUB 5690,5970,1660,1680,2300,2520,2840,6110,3150,3530,3710,4640,4920,5010
1610 IF MD%(2) = 3 OR MD%(2) = 4 THEN RETURN
1620 GOTO 1640
1630 ON MD%(2) GOSUB 2300,2520,2840,6110,3150,3530,3710,4640,4920,5010,5450
1640 IF ZM = 0 THEN MD%(0) = 3: RETURN
1650 MD%(0) = 4:MD%(2) = 11 - 4 * (MD%(1)):ZM = 0: RETURN
1660 IF FP%(3) = 0 AND SC%(SC%,5) = 0 THEN MD%(0) = 3:MD%(1) = 1: RETURN
1670 MD%(0) = 1:MD%(1) = 2: GOTO 1690
1680 MD%(0) = 1:MD%(1) = 3: GOTO 1690
1690 GOSUB 1300: GOSUB 1280
1700 PG%(3) = 1:PG%(4) = PG%(1): GOSUB 7640
1710 GOSUB 7240: GOSUB 8020: GOSUB 7690: RETURN
1720 REM SORT ROUTINE
1730 GOSUB 9100: FOR IT = 1 TO 3: ON IT GOSUB 1790,1800,1810: NEXT
1740 VC = 11:HC = 1: GOSUB 9180: IF IN$ = ESC$ THEN GOSUB 6880: RETURN
1750 IF IN$ > = "1" AND IN$ < = "3" THEN INVERSE : ON VAL (IN$) GOSUB 1790,1800,1810: GOTO 1770
1760 PRINT BL$;: GOTO 1740
1770 GOSUB 2280
1780 ON VAL (IN$) GOSUB 1820,1860,2000: RETURN
1790 VTAB 5: HTAB 1: PRINT "1) SORT CHILDREN OF CURRENT PARENT": NORMAL : RETURN
1800 VTAB 7: HTAB 1: PRINT "2) SORT GENERATION THROUGHOUT OUTLINE": NORMAL : RETURN
1810 VTAB 9: HTAB 1: PRINT "3) SORT ALL CHILDREN IN THE OUTLINE": NORMAL : RETURN
1820 GOSUB 7640: IF SR%(0) < = 1 THEN GOSUB 7690: GOSUB 6880: RETURN
1830 BF%(0,0) = SC%(0,7): GOSUB 2140: GOSUB 2230
1840 GOSUB 2260: IF BF%(0,0) > 0 THEN FOR IT = 1 TO 7:SC%(0,IT) = GS%(IT): NEXT :SC$(0) = GS$: GOSUB 8300
1850 PG%(5) = 1: GOSUB 6790: GOSUB 7690: GOSUB 6880: RETURN
1860 VTAB 11: HTAB 5:VC = 11: CALL - 868
1870 PRINT "ENTER GENERATION TO SORT: "
1880 HX = 1: FOR IT = 1 TO 4:ET$(IT) = "": NEXT
1890 IF HX < 1 THEN HX = 1
1900 IF HX > 4 THEN HX = 4
1910 HC = HX + 29: GOSUB 9180
1920 IF IN$ = ESC$ THEN RETURN
1930 IF IN$ = LE$(4) OR IN$ = LF$ THEN HX = HX - 1: GOTO 1890
```

Continued

```
1940 IF IN$ = LE$(5) OR IN$ = RT$ THEN HX = HX + 1: GOTO 1890
1950 IF IN$ = RB$ THEN 1980
1960 IF IN$ >= "0" AND IN$ <= "9" OR IN$ = " " THEN ET$(HX) = IN$:HX = HX + 1: GOTO 1890
1970 PRINT BL$;: GOTO 1910
1980 LW%(2) = VAL (ET$(1) + ET$(2) + ET$(3) + ET$(4)): VTAB 11: HTAB 26: CALL - 868: PRINT LW%(2):LW%(2) = LW%(2) + 1
1990 LW%(0) = 1: GOSUB 2010: RETURN
2000 LW%(0) = 0: GOSUB 2010: RETURN
2010 GOSUB 2280: GOSUB 7640: IF LW%(0) = 1 AND LW%(2) < > 1 THEN 2040
2020 GS%(7) = FP%(4): GOSUB 6710
2030 GOSUB 2140: GOSUB 2230:BF%(0,0) = 0: GOSUB 2260: IF LW%(0) = 1 AND LW%(2) = 1 THEN 2090
2040 BF%(0,0) = FP%(4):LW%(1) = 2
2050 GOSUB 2130: IF (LW%(0) = 0 OR LW%(2) = LW%(1)) AND GS%(2) > 0 THEN GOSUB 2120: GOSUB 2130
2060 IF LW%(0) = 0 OR (LW%(0) = 1 AND LW%(1) < LW%(2)) THEN IF GS%(2) > 0 THEN BF%(0,0) = GS%(2):LW%(1) = LW%(1) + 1: GOTO 2050
2070 IF GS%(4) > 0 THEN BF%(0,0) = GS%(4): GOTO 2050
2080 IF GS%(1) > 0 THEN BF%(0,0) = GS%(1): GOSUB 2130:LW%(1) = LW%(1) - 1: GOTO 2070
2090 IF SC%(0,7) = 0 THEN GS%(7) = FP%(4):SC%(0,2) = FP%(4): GOTO 2110
2100 GS%(7) = SC%(0,7): GOSUB 8340: FOR IT = 1 TO 7:SC%(0,IT) = GS%(IT): NEXT :GS%(7) = GS%(2)
2110 GOSUB 6710:PG%(5) = 1: GOSUB 6790: GOSUB 7690: GOSUB 6880: RETURN
2120 GOSUB 2130:GS%(7) = GS%(2): GOSUB 6710: GOSUB 2140: GOSUB 2230: GOSUB 2260: RETURN
2130 GS%(7) = BF%(0,0): GOSUB 8340: RETURN
2140 D = 2 ^ INT ( LOG (SR%(0)) / LOG (2)) - 1
2150 FOR IT = 1 TO SR%(0) - D: FOR G = 1 TO 2:BF%(0,G) = IT + D * (G - 1): GOSUB 2220: NEXT G: IF BF$(1) <= BF$(2) THEN 2200
2160 TX = SR%(IT + D):SR%(IT + D) = SR%(IT): IF IT <= D THEN SR%(IT) = TX: GOTO 2200
2170 FOR JI = IT - D TO 1 STEP - D:G = 1:BF%(0,G) = JI: GOSUB 2220: IF BF$(1) <= BF$(2) THEN 2190
2180 SR%(JI + D) = SR%(JI): NEXT JI
2190 SR%(JI + D) = TX
2200 NEXT IT:D = INT (D / 2): IF D > 0 THEN 2150
2210 RETURN
2220 GS%(7) = SR%(BF%(0,G)): GOSUB 8250: FOR XI = 1 TO 7:BF%(G,XI) = GS%(XI): NEXT :BF$(G) = GS$: RETURN
2230 GS%(7) = SR%(1): GOSUB 8250:GS%(3) = 0: IF SR%(0) = 1 THEN GS%(4) = 0: GOSUB 8300: RETURN
2240 GS%(4) = SR%(2): GOSUB 8300:GS%(7) = SR%(SR%(0)): GOSUB 8250:GS%(4) = 0:GS%(3) = SR%(SR%(0) - 1): GOSUB 8300: IF SR%(0) = 2 THEN RETURN
2250 FOR IT = 2 TO SR%(0) - 1:GS%(7) = SR%(IT): GOSUB 8250:GS%(4) = SR%(IT + 1):GS%(3) = SR%(IT - 1): GOSUB 8300: NEXT : RETURN
2260 IF BF%(0,0) = 0 THEN FP%(4) = SR%(1): GOSUB 8020: RETURN
2270 GS%(7) = BF%(0,0): GOSUB 8250:GS%(2) = SR%(1): GOSUB 8300: RETURN
2280 VTAB 11: HTAB 1: CALL - 868: GOSUB 1280: RETURN
2290 REM MOVE CURSOR UP
2300 IF SC% = 1 THEN 2320
2310 SC% = SC% - 1: GOSUB 7060: RETURN
2320 IF SC%(1,3) = 0 THEN RETURN
2330 PG%(2) = 1: GOSUB 7640: GOSUB 6590
2340 PG%(5) = PG%(5) - BF%
2350 PG%(4) = PG%(1):PG%(3) = PG%(1) - BF% + 1: IF PG%(3) < 1 THEN PG%(3) = 1
2360 GOSUB 7240: GOSUB 7690
2370 PG%(1) = BF% + PG%(1)
2380 IF PG%(1) > DU% THEN PG%(1) = DU%
2390 SC% = BF% + 1: GOSUB 7060
2400 FOR IT = PG%(1) TO 1 STEP - 1:LN% = IT
2410 IF IT > BF% THEN 2450
2420 SC$(IT) = BF$(BF% + 1 - IT)
2430 FOR JI = 1 TO 7:SC%(IT,JI) = BF%(BF% + 1 - IT,JI): NEXT
2440 GOTO 2470
2450 SC$(IT) = SC$(IT - BF%)
2460 FOR JI = 1 TO 7:SC%(IT,JI) = SC%(IT - BF%,JI): NEXT
2470 GOSUB 6960: GOSUB 7100
2480 NEXT
2490 GOSUB 7160
2500 RETURN
2510 REM MOVE CURSOR DOWN
2520 PG%(2) = 0
2530 IF SC% = DU% THEN 2660
2540 IF SC% = PG%(1) THEN 2560
2550 SC% = SC% + 1: GOSUB 7060: RETURN
2560 IF ZL = 1 OR FP%(3) = 0 THEN RETURN
2570 GOSUB 7640: GOSUB 8090: GOSUB 7690
2580 SC%(PG%(1),4) = NX%
2590 PG%(1) = PG%(1) + 1:SC%(PG%(1),7) = NX%
2600 SC%(PG%(1),1) = SC%(SC%,1):SC%(PG%(1),2) = 0:SC%(PG%(1),3) = SC%(SC%,7)
2610 SC%(PG%(1),4) = 0:SC%(PG%(1),5) = 0:SC%(PG%(1),6) = SC%(SC%,6):SC$(PG%(1)) = LB$:ZM = 1
2620 SR%(0) = SR%(0) + 1:SR%(SR%(0)) = SC%(PG%(1),7)
2630 SC% = PG%(1)
2640 GOSUB 7060: GOSUB 7100: GOSUB 7030
2650 RETURN
2660 BF% = 0: GOSUB 7640
2670 IF SC%(DU%,4) = 0 THEN 2690
2680 GOSUB 6590
2690 PG%(3) = 1:PG%(4) = SL%: GOSUB 7240: GOSUB 7690
2700 PG%(5) = PG%(5) + SL%
2710 SC% = DU% - SL%:PG%(1) = BF% + DU% - SL%
2720 VC = TM% + 1:HC = DU%: GOSUB 9130
2730 GOSUB 7060
2740 FOR IT = 1 TO DU% - SL% + BF%:LN% = IT: IF IT > DU% - SL% THEN 2780
2750 SC$(IT) = SC$(IT + SL%)
2760 FOR JI = 1 TO 7:SC%(IT,JI) = SC%(IT + SL%,JI): NEXT
2770 GOTO 2800
2780 SC$(IT) = BF$(IT - DU% + SL%)
2790 FOR JI = 1 TO 7:SC%(IT,JI) = BF%(IT - DU% + SL%,JI): NEXT
2800 GOSUB 6960: GOSUB 7100: NEXT
2810 GOSUB 7160
2820 RETURN
2830 REM LINE EDIT
2840 VC = TM% + SC%:HC = 3
2850 VTAB VC: HTAB HC: CALL - 868
2860 FOR IT = 1 TO 38:ET$(IT) = MID$ (SC$(SC%),IT,1): VTAB VC: HTAB 2 + IT: PRINT ET$(IT);: NEXT
2870 VTAB 1: HTAB 38: CALL - 868: PRINT HC - 2;: GOSUB 9180
2880 IF ASC (IN$) < 31 THEN 2920
2890 ET$(HC - 2) = IN$: IF HC = 40 THEN GOSUB 6280: IF LX% = 1 THEN 2870
2900 HC = HC + 1: IF HC > 40 THEN HC = 40
2910 GOTO 2870
2920 IF IN$ < > ESC$ AND IN$ < > RB$ THEN 2950
2930 SC$(SC%) = "": FOR IT = 1 TO 38:SC$(SC%) = SC$(SC%) + ET$(IT): NEXT
2940 RETURN
2950 CX% = 4 * (IN$ = LF$) + 5 * (IN$ = RT$): IF CX% > 0 THEN 2980
2960 FOR GX = 1 TO 5: IF IN$ = LE$(GX) THEN CX% = GX: GOTO 2980
2970 NEXT : PRINT BL$;: GOTO 2870
2980 ON CX% GOTO 2990,3060,3080,3090,3110
2990 IF MD%(1) = 1 AND (ET$(38) < > " " OR HC = 40) THEN 2870
3000 IF ET$(38) < > " " THEN IM% = 1:TV = VC:TH = HC: GOSUB 6280: GOSUB 2300:VC = TV:HC = TH
3010 IF IM% = 0 THEN 3030
3020 FOR IT = 1 TO 38:ET$(IT) = MID$ (SC$(SC%),IT,1): NEXT :IM% = 0
3030 FOR IT = 38 TO HC - 1 STEP - 1:ET$(IT) = ET$(IT - 1): VTAB VC: HTAB 2 + IT: PRINT ET$(IT);: NEXT
3040 ET$(HC - 2) = " ": VTAB VC: HTAB HC: PRINT ET$(HC - 2);
3050 GOTO 2870
3060 IF HC = 40 THEN 2870
3070 FOR IT = HC - 2 TO 37:ET$(IT) = ET$(IT + 1): VTAB VC: HTAB IT + 2: PRINT ET$(IT);: NEXT :ET$(38) = " ": VTAB VC: HTAB 40: PRINT " ";: GOTO 2870
3080 FOR IT = 1 TO 38:ET$(IT) = " ": VTAB VC: HTAB 2 + IT: PRINT ET$(IT);: NEXT : GOTO 2870
3090 HC = HC - 1: IF HC < 3 THEN HC = 3
3100 GOTO 2870
3110 HC = HC + 1: IF HC > 40 THEN HC = 40
3120 GOTO 2870
3130 RETURN
3140 REM INSERT LINE
3150 IF FP%(3) = 0 THEN RETURN
3160 GOSUB 7640
3170 FOR IT = 1 TO 7:BF%(2,IT) = SC%(SC%,IT):BF%(3,IT) = SC%(0,IT):BF%(0,IT) = 0: NEXT
3180 BF$(2) = SC$(SC%):BF$(3) = SC$(0):BF$(0) = LB$
```

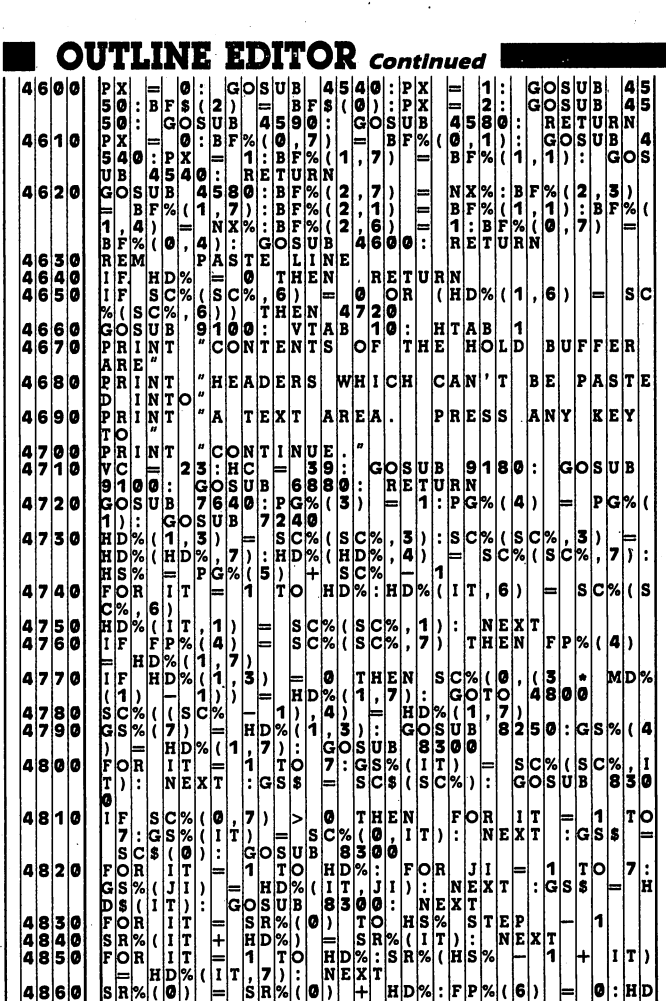
Continued

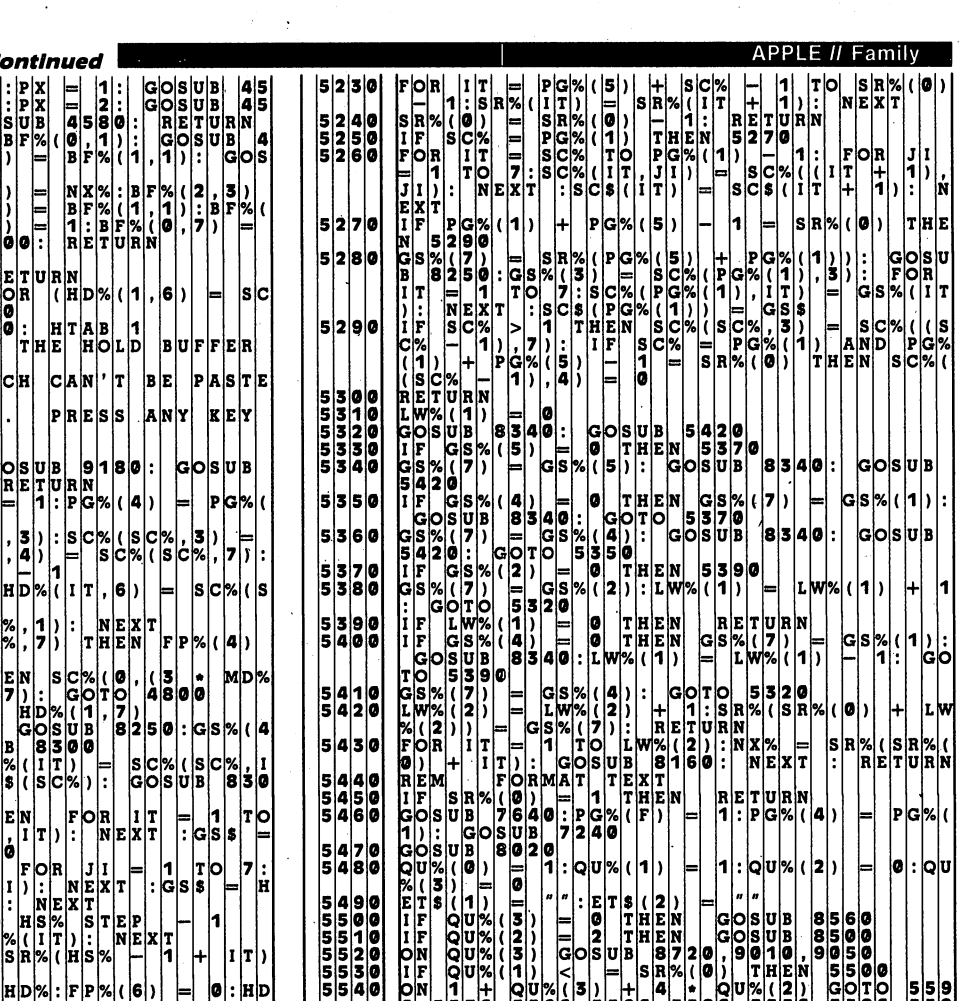
```
31900
32000
32100
32200

32300
32400

32500

32600
32700

32800
32900

33000

33100

33200

33300

33400

33500

33600
33700
33800

33900
34000

34100
34200
34300
34400

34500
34600

34700
34800
34900
35000
35100
35200
35300

35400
35500

35600

35700

35800

35900
36000

36100
36200

36300

36400
36500

36600
36700
36800
36900
37000
37100

37200
37300
37400

37500
37600
37700
37800

37900

38000
38100

38200
38300
38400

38500
38600
```

```
38700
38800

38900

39000
39100
39200
39300
39400

39500
39600

39700
39800
39900
40000

40100
40200
40300
40400
40500
40600
40700
40800

40900
41000
41100
41200
41300

41400
41500

41600
41700
41800
41900

42000

42100
42200
42300
42400
42500
42600

42700

42800
42900

43000
43100

43200

43300
43400
43500
43600

43700
43800

43900

44000

44100
44200
44300
44400

44500
44600
44700

44800
44900
45000
45100

45200
45300
45400

45500
45600

45700

45800

45900
```

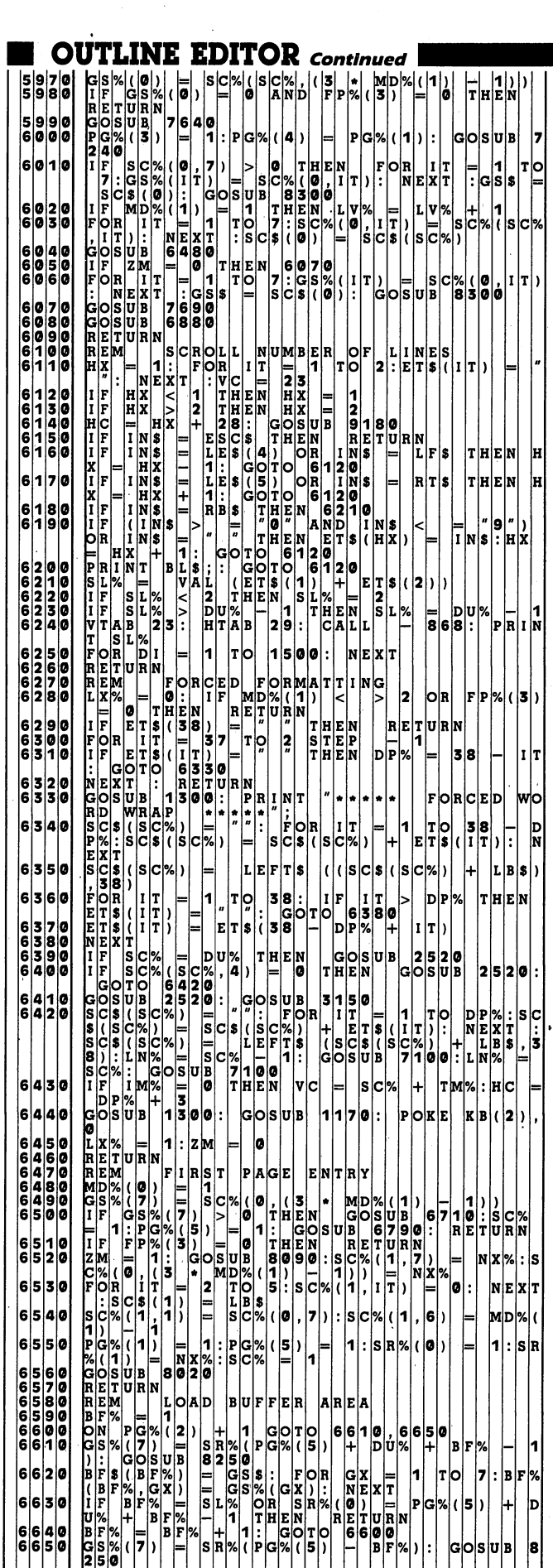
Continued

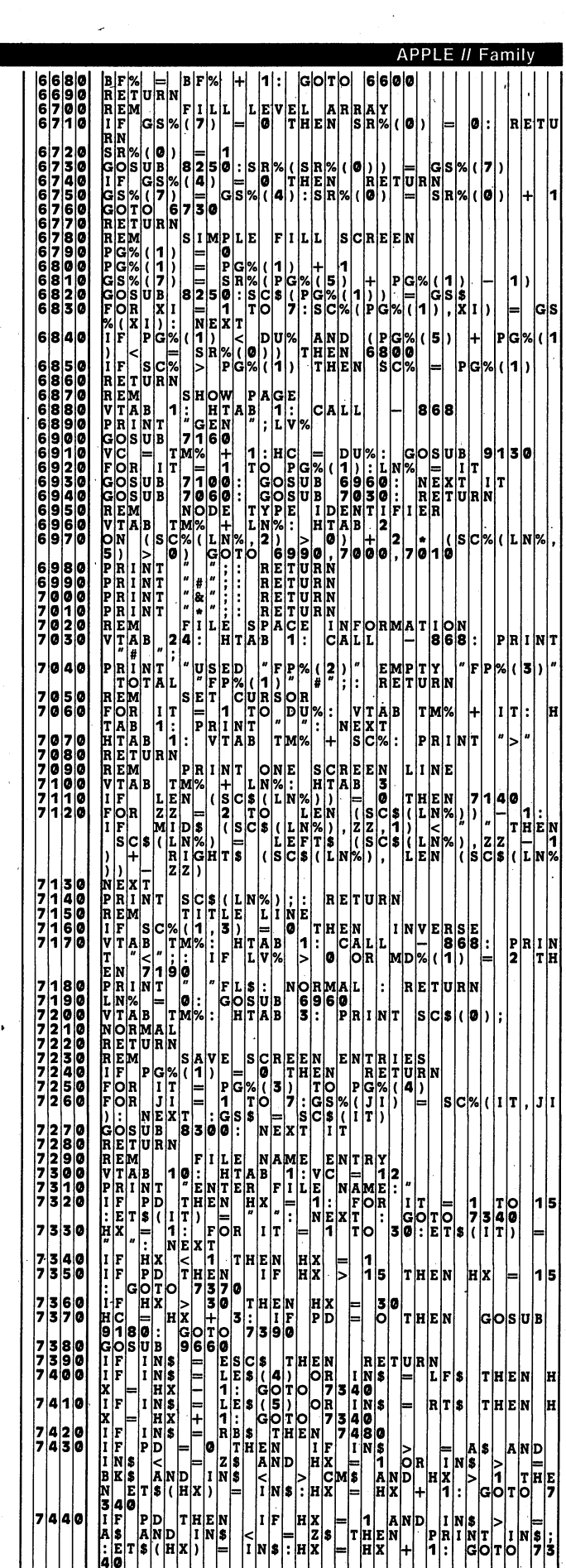
OUTLINE EDITOR *Continued*





Continued





Continued

OUTLINE EDITOR *Continued*


```
7460 PRINT BL$;:: GOTO 7370
7470 IF PD THEN 7510
7480 FOR I=T TO 1 STEP -1
7490 XT=XT*10+ASC(MID$(X$,I,1))-48: NEXT: RETURN
7500 NEXT I: GOTO 7540
7510 FOR I=T TO 1 STEP -1
7520 XT=XT*10+ASC(MID$(X$,I,1))-48: NEXT: RETURN
7530 NEXT I
7540 FL$="": FOR I=1 TO XT: FL$=FL$+CHR$(ASC("A")+I-1): NEXT: RETURN
7550 DR$="": PRINT DRIVE NUMBER: INPUT DR$: VTAB 1
7560 PRINT CHR$(4);"PR#";DR$
7570 VTAB 10: HTAB 37: PRINT "  ": VTAB 10: HTAB 37: INPUT "";A$: RETURN
7580 GET A$: IF A$="" THEN 7580
7590 IF A$=CHR$(13) OR A$=" " THEN A$="": RETURN
7600 IF A$="1" OR A$="2" THEN DR$=A$: RETURN
7610 PRINT BL$;: GOTO 7570
7620 RETURN
7630 REM  OPEN A FILE
7640 VTAB 23: PRINT CHR$(4);"VERIFY";D$
7650 IF SP="" THEN SP=CHR$(0): PRINT CHR$(4);"OPEN";D$: GOTO 7590
7660 PRINT CHR$(4);"OPEN";D$;",L";SP: PRINT CHR$(4);"WRITE";D$: PRINT CHR$(4);"CLOSE": RETURN
7670 REM  CLOSE A FILE
7680 REM
7690 VTAB 23: PRINT CHR$(4);"VERIFY";D$
7700 PRINT CHR$(4);"CLOSE";D$
7710 RETURN
7720 REM  FILE WRITE
7730 VTAB 23: PRINT CHR$(4);"WRITE";D$
7740 PRINT CHR$(4);"WRITE";D$;",R";RL: PRINT CHR$(4);"CLOSE";D$: RETURN
7750 PRINT DX$
7760 RETURN
7770 REM  FILE READ
7780 VTAB 23: PRINT CHR$(4);"READ";D$
7790 PRINT CHR$(4);"READ";D$;",R";RL: PRINT CHR$(4);"CLOSE";D$: RETURN
7800 TX$="": FOR GX=1 TO FD: GET IN$: TX$=TX$+IN$: NEXT: RETURN
7810 FOR GX=1 TO FD: GET IN$: TX$=TX$+IN$: NEXT
7820 RETURN
7830 REM  DECODE POINTERS
7840 FOR ZX=1 TO 9: ZX$=ZX$+CHR$(ASC(MID$(X$,ZX,1))-ASC("A")): NEXT
7850 GS%=((EX$+ZX$,1,1)-(EX$,ZX+1,1))*100+((EX$+ZX$,2,1)-(EX$,ZX+2,1))*10+((EX$+ZX$,3,1)-(EX$,ZX+3,1))
7860 NEXT
7870 RETURN
7880 REM  ENCODE POINTERS
7890 EX$="": FOR GX=1 TO 5: RIGHT$(QZ$+STR$(GS%(GX)),4): EX$=EX$+CHR$(VAL(MID$(QZ$,4,1))*16+VAL(MID$(QZ$,3,1))): NEXT
7900 FOR GX=1 TO 4: EX$=EX$+CHR$(VAL(MID$(QZ$,GX,1))*16+VAL(MID$(QZ$,GX+4,1))): NEXT
7910 RETURN
7920 REM  SET FILE PARAMETERS
7930 NEXT
7940 RETURN
7950 REM  GET FILE PARAMETERS
7960 DX$=MID$(X$,1,7): FD=ASC(MID$(X$,8,1))
7970 GOSUB 7780
7980 FOR GX=1 TO FD: GET IN$: TX$=TX$+IN$: NEXT
7990 FP%=(LEN(TX$)-4)/256: NEXT
8000 RETURN
8010 REM  STORE FILE PARAMETERS
8020 DX$=LEFT$(X$,25)
8030 IF FP%=0 THEN FP%=1
8040 FOR GX=1 TO FP%: PUT IN$: NEXT
8050 TX$=TX$+RIGHT$(QZ$+STR$(FP%),4)
8060 GOSUB 7730
8070 RETURN
8080 REM  GET A NEW NODE
8090 IF FP%=0 THEN NX%=0: RETURN
8100 FP%=FP%-1: NX%=FP%: IF FP%=0 THEN FP%=1
8110 NX%=FP%: IF FP%=0 THEN FP%=1
8120 DX$=LEFT$(X$,4)+MID$(X$,9,3)
8130 FP%=VAL(TX$)
8140 RETURN
8150 REM  STORE A NODE TO EMPTY LIST
8160 IF NX%=0 THEN RETURN
8170 DX$=MID$(X$,1,4)+RIGHT$(QZ$+STR$(NX%),4)
8180 TX$=TX$+DX$: FP%=NX%
8190 PUT IN$: NEXT
8200 FP%=FP%-1: IF FP%=0 THEN FP%=1
8210 GOSUB 7730
8220 RETURN
8230 REM  GET AN OCCUPIED NODE
8240 DX$=LEFT$(X$,4)+MID$(X$,9,3)
8250 EX$=LEFT$(TX$,10)+MID$(TX$,11,1)
8270 GOSUB 7840: GS%=GS%+1
8280 RETURN
8290 REM  STORE TO AN OCCUPIED NODE
8300 GOSUB 7890
8310 TX$=EX$+MID$(TX$,11)
8320 RETURN
8330 REM  GET
8340 GS%=(ASC(MID$(X$,1,1))-48)*100+(ASC(MID$(X$,2,1))-48)*10+(ASC(MID$(X$,3,1))-48)
8350 TX$=MID$(X$,4)
8360 RETURN
8370 REM  PUT
8380 X$=STR$(GS%): X$=MID$(X$,2)+LEFT$("   ",3-LEN(X$))
8390 X$=X$+TX$
8400 GOSUB 7730
8410 X$=LEFT$(X$,3)
8420 FOR I=1 TO 3: HD%=HD%*10+ASC(MID$(X$,I,1))-48: NEXT
8430 RETURN
8440 REM  ISOLATE HOLD BUFFER
8450 HD%=(1,3)
8460 FOR I=1 TO HD%: HD%=HD%(I,1)=0: NEXT
8470 FOR I=1 TO HD%: HD%=HD%(I,I)): NEXT: GOSUB 7730: RETURN
8480 RETURN
8490 REM  FORMATTED LINE
8500 GS%=QU%(0): GOSUB 8250: QU%=QU%(0)
8510 GET$=LEFT$(GET$(1),1)
8520 GU%=QU%(2): GOSUB 8300: QU%(0)=QU%
8530 FOR I=1 TO 38: QU%(4)=QU%(4)*2+QU%(3): NEXT: RETURN
8540 RETURN
8550 REM  READ A LINE TO FORMAT
8560 GS%=QU%(1): GOSUB 8250: QU%=QU%(1)
8570 FOR I=1 TO 38: QU%(4)=QU%(4)*2+QU%(3): NEXT: RETURN
8580 FOR I=1 TO 38: QU%(3)=QU%(4)-INT(QU%(4)/2)*2: QU%(4)=INT(QU%(4)/2): NEXT: RETURN
8590 IF QU%(5)<>QU%(4) THEN QU%(3)=0: RETURN
8600 NEXT
8610 FOR I=1 TO 38: QU%(5)=QU%(5)*2+QU%(3): NEXT: RETURN
8620 IF QU%(5)<>QU%(4) THEN QU%(3)=0: RETURN
8630 IF MID$(GS$,I,1)<>"*" THEN 8650
8640 NEXT
8650 QU%(5)=QU%(5)-QU%(4)+1
8660 GET$=MID$(GS$,I,1): QU%(4)=QU%(4)+QU%(3)*2^(I-1): RETURN
8670 IF GET$="*" THEN QU%(3)=2: RETURN
8680 IF GET$="*B" THEN QU%(3)=3: RETURN
8690 QU%(3)=1: RETURN
8700 RETURN
8710 REM  REGULAR FILL
8720 GET$=GET$+GET$: X=LEN(GET$)
8730 IF X=38 THEN 8810
8740 IF X<38 THEN 8820
8750 GET$=RIGHT$(GET$,38)
8760 GET$=LEFT$(GET$,38)
8770 IF LEFT$(GET$,2)="*B" THEN 8900
8780 GET$=RIGHT$(GET$,LEN(GET$)-1)+LEFT$(GET$,1)
8790 GET$=LEFT$(GET$,LEN(GET$)-1)+RIGHT$(GET$,1)
8800 GOSUB 8770
8810 QU%=2: GOSUB 8370: RETURN
8820 QU%=3: GOSUB 8370: RETURN
8830 IF GET$="*" THEN QU%=3: RETURN
8840 IF GET$="*B" THEN QU%=3: RETURN
8850 NEXT
8860 GET$=GET$+GET$
8870 IF LEN(GET$)>38 THEN GET$=LEFT$(GET$,38)
8880 QU%=2: IF LEN(GET$)<38 THEN GET$=GET$+" ": GOTO 8880
8890 GET$=LEFT$(GET$,38): RETURN
8900 QU%=2: GOSUB 8370: QU%=3: RETURN
8910 GET$=LEFT$(GET$,LEN(GET$)-1)+CHR$(ASC(RIGHT$(GET$,1))+128)
8920 GOSUB 8770
8930 FOR I=1 TO 38: QU%=QU%*2+(GET$<>" "): GET$=LEFT$(GET$,LEN(GET$)-1): NEXT: RETURN
8940 IF MID$(GET$,I,1)<" " THEN QU%=QU%+1
8950 NEXT
8960 QU%=QU%*3: RETURN
8970 QU%=QU%*3+(GET$="*")+(GET$="*B"): RETURN
8980 RETURN
8990 RETURN
9000 REM  PARAGRAPH
9010 IF QU%=2 THEN GOSUB 8500
9020 GET$=QU%(1)*3: IF QU%=3 THEN GET$="*B": RETURN
9030 RETURN
9040 REM  BLANK LINE
9050 IF QU%=2 THEN GOSUB 8500
9060 GET$="*B": GOSUB 8500
9070 QU%=0
9080 RETURN
9090 REM  CLEAR PAGE
9100 HCC=3: HCC$=CHR$(HCC): PRINT HCC$"PR#0"
9110 GOSUB 9130: RETURN
9120 REM  BLANK PORTION OF PAGE
9130 FOR GX=1 TO 24-PEEK(37): PRINT: NEXT: RETURN
```

Continued

OUTLINE EDITOR *Continued*

APPLE // Family

```
9140  VTAB GX: HTAB 1
9150  CALL -868: NEXT GX
9160  RETURN
9170  REM GET CHARACTER
9180  VTAB VC: HTAB HC
9190  IN$ = "": GET IN$: IF IN$ = "" OR ASC (IN$) < 96 THEN 9200
9210  IF ASC (IN$) > 31 THEN VTAB VC: HTAB HC: PRINT IN$;
9220  RETURN
9230  REM RETURN TO ORGANIZER
9240  ZQ = 1: HOME : VTAB 7: HTAB 1
9250  HOME : VTAB 7: HTAB 1
9260  PRINT "INSERT THE PROGRAM DISK INTO DRIVE 1, THEN PRESS ANY KEY: ";
9270  VC = 8: HC = 21: GOSUB 9180
9280  IF PD THEN GOSUB 9620
9300  HOME : VTAB 10: HTAB 1: PRINT "RETURNING TO THE MAIN MENU...."
9310  PRINT CHR$ (4);"RUN ORGANIZE,D";DR$
9320  RETURN
9330  REM ERROR HANDLER
9340  HOME : PRINT CHR$ (4)"CLOSE"
9350  HOME : VTAB 7: HTAB 1
9360  X = PEEK (222)
9370  ON ( (X = 6 OR X = 5) AND ZQ = 0) + ( (X = 6 OR X = 5) * 2 + 3 * ( (X = 255) GOTO 9460,9530,9560
9380  PRINT "ERROR NUMBER "X" ;
9390  PRINT "AT LINE "( PEEK (218) + PEEK (219) * 256)
9400  VTAB 12: HTAB 1
9410  PRINT "PRESS ANY KEY TO CONTINUE: ";
9420  GET IN$: IF IN$ = "" THEN 9420
9430  RUN
9440  RETURN
9450  REM DETAIL ERROR EXPLANATIONS
9460  PRINT "FILE ";FL$;" ": PRINT "NOT FOUND ON DISKETTE."
9480  PRINT "": PRINT CHR$ (4)"DELETE FL$
9490  GOTO 9400
9500  PRINT "FILE ";FL$;" ": PRINT "IS NOT A GOOD DATA FILE."
9520  GOTO 9400
9530  PRINT "'ORGANIZER' PROGRAM NOT ON DISK"
9540  PRINT "INSERTED IN DRIVE 1."
9550  GOTO 9400
9560  PRINT "CONTROL 'C' KEY WAS PRESSED,"
9570  PRINT "STOPPING THE PROGRAM."
9580  GOTO 9400
9590  HTAB 1: VTAB 22: PRINT "PLACE DISK IN DRIVE 1 AND PRESS RETURN"
9600  GOSUB 9640: IF KB < > (141) THEN 9600
9610  HTAB 1: VTAB 22: CALL -868
9620  PRINT CHR$ (4);"PREFIX,D";DR$
9630  RETURN
9640  KB = PEEK (-16384): IF KB > 127 THEN POKE -16368,0: RETURN
9650  GOTO 9640
9660  VTAB VC: HTAB HC
9670  IN$ = "": GET IN$: IF IN$ = "" OR ASC (IN$) < 96 THEN 9680
9690  RETURN
```

HCM

The Organizer for the COMMODORE 64

MAIN MENU

COMMODORE 64

```
100  REM ********** THE ORGANIZER **********
110  REM * THE ORGANIZER *
120  REM * MAIN MENU *
130  REM **********
140  REM COPYRIGHT 1984, 1985
150  REM EMERALD VALLEY PUBLISHING CO.
160  REM BY WILLIAM K. BALTHROP
170  REM AND THE HCM STAFF
180  REM HOME COMPUTER MAGAZINE
190  REM VERSION 5.1.1
200  REM COMMODORE 64 BASIC
210  REM USE THE FILE NAME "ORGANIZE" TO SAVE THIS FILE ON DISK
220  POKE 46,90:CLR
230  NOP=4:DIM SC$(NOP,1):POKE 53281,12:POKE 53280,12:PRINT CHR$(8)
240  PRINT "■SHIFT CLR■CTRL WHT■" TAB(13) "THE ORGANIZER":PRINT TAB(15) "■CRSRDOWN■MAIN MENU"
250  PRINT "■CRSRDOWN■SELECT ONE:":GOSUB 420
260  FOR Z=1 TO NOP:R=4+Z*2:C=6:GOSUB 480:PRINT STR$(Z);")";:SC$(Z,0)=NEXT
270  R=13:C=4:GOSUB 480:GET A$:IF A$="" THEN 270
280  SEL=VAL(A$):IF SEL<1 OR SEL>4 THEN 270
290  IF SC$(SEL,1)="*" THEN PRINT "■SHIFT CLR■":END
300  PRINT "■SHIFT CLR■":R=10:C=7:GOSUB 480:PRINT "PLACE PROGRAM DISK IN DRIVE 1"
310  PRINT "AND PRESS RETURN"
320  IF PEEK(197)<>1 THEN 320
330  OPEN15,8,15,"I0":OPEN 6,8,6,+SC$(SEL,1)+",S,R":REM FORCE ERROR 64
340  INPUT#15,EN,EM$,ET,ES
350  IF EN=64 THEN CLOSE6:CLOSE15:PRINT "■SHIFT CLR■CRSRDOWN■GETTING ";SC$(SEL,0);":LOAD ■SHIFT CLR■6 ■CRSRDOWN■";SC$(SEL,1),8
360  PRINT "■SHIFT CLR■DOWN■ DISK ERROR #";EN;EM$:IF EN<>0 THEN CLOSE6:CLOSE15:GOTO 280
370  PRINT "■CRSRDOWN■ TRYING TO GET "T$(SEL,0)
380  PRINT "■SCRSRDOWN■ PLEASE CORRECT ERROR AND TRY AGAIN"
390  PRINT "■CRSRDOWN■ PRESS [RETURN] TO CONTINUE":CLOSE6:CLOSE15
400  IF PEEK(197)<>1 THEN 400
410  RUN
420  REM * SET SC$ ARRAY *
430  FOR SC=1 TO NOP:READ SC$(SC,0),SC$(SC,1):NEXT SC:RETURN
440  DATA ODATA EDITOR,OUTLINE
450  DATA ODATA REPORTS,REPORTS
460  DATA FILE MANAGER,FILEMGR
470  DATA QUIT,* MOVE CURSOR R=ROW C=COLUMN *
480  POKE 781,R:POKE 782,C:POKE 783,0:SYS 65520:RETURN
```

HCM

REPORTS

COMMODORE 64

```
100  REM ********** REPORTS **********
110  REM * DUMMY REPORTS *
120  REM * PROGRAM *
130  REM **********
140  REM LOOK FOR REPORTS
150  REM PROGRAM IN VOL. 5 NO.2 OF
160  REM HOME COMPUTER MAGAZINE
170  REM THIS PROGRAM SHOULD BE SAVED WITH THE FILE NAME "REPORTS"
180  REM
190  PRINT "■SHIFT CLR■8CRSRDOWN■THE REPORTS PROGRAM WILL BE SUPPLIED"
200  PRINT "WITH VOLUME 5, NO. 2 OF HOME COMPUTER MAGAZINE."
210  PRINT "PRESS RETURN TO GO BACK TO MAIN MENU"
220  IF PEEK(197)<>1 THEN 220
230  LOAD "ORGANIZE",8
240  END
```

HCM

FILE MANAGER

COMMODORE 64

```
100  REM ********** THE ORGANIZER **********
110  REM * FILE MANAGER *
120  REM **********
130  REM
140  REM COPYRIGHT 1984, 1985
150  REM EMERALD VALLEY PUBLISHING CO.
160  REM BY WILLIAM K. BALTHROP
170  REM AND THE HCM STAFF
180  REM HOME COMPUTER MAGAZINE
190  REM VERSION 5.1.1
200  REM COMMODORE 64 BASIC
210  REM
220  REM USE THE FILE NAME "FILEMGR" TO SAVE THIS FILE ON DISK
230  REM
240  CLR:GOSUB 1310:GOSUB 1340
250  PRINT "■CRSRDOWN■SELECT ONE:":FOR Z=1 TO 5:R=4+Z*2:C=6:GOSUB 1280:PRINT Z;")";:NEXT
260  READ A$:PRINT " STR$(Z);")";A$:NEXT
270  GET A$:IF A$="" THEN 270
280  SEL=VAL(A$):IF SEL<1 OR SEL>5 THEN 270
290  ON SEL GOTO 300,520,830,980,430
300  PRINT "■SHIFT CLR■":OPEN6,8,0,"$0":OPEN15,8,15:GOSUB 1360:IF EN>20 THEN 1380
310  GET#6,A$:A$=A$
320  GET#6,A$:A$=A$
330  IF A$="" THEN 400
340  GET#6,A$:L$=A$
350  PRINT#6,ASC(A$+NL$)+ASC(L$+NL$)*256;
360  GET#6,A$
370  IF A$="" THEN PRINT:GOTO 320
```

Continued

PROGRAM LISTING

```
380 PRINT A$;
390 GOTO 360
400 CLOSE6:PRINT TAB(5) "PRESS [RETURN] TO CONTINUE":DIR=0
410 GET A$:IF A$<>CR$ THEN 410
420 RUN
430 REM CHAIN TO MAIN MENU
440 PRINT"{SHIFT CLR}":R=10:C=7:GOSUB 1280:PRINT "PLACE PROGRAM DISK IN DRIVE"
450 PRINT"                AND PRESS RETURN"
460 IF PEEK(197)<>1 THEN 460
470 OPEN15,8,15,"I0":OPEN6,8,6,"ORGANIZE,S,R":GOSUB 1360:REM FORCE ERROR 64
480 IF EN=64THEN CLOSE6:CLOSE15:PRINT"{SHIFT CLR}{CRSR DOWN}GETTING MAIN MENU":LOAD"ORGANIZE",8
490 CHAIN=-1:CLOSE6:GOTO 1380
500 DATA LIST FILE NAMES ON DISK,CREATE A NEW ORGANIZER FILE
510 DATA DELETE AN ORGANIZER FILE,ENLARGE AN ORGANIZER FILE,EXIT TO MAIN MENU
520 REM CREATE NEW ORGANIZER FILE
530 PRINT"{SHIFT CLR}" TAB(13)"FILE MANAGER":PRINT TAB(13) "{CRSR DOWN}CREATE FILE{CRSR DOWN}"
540 PRINT "FILE NAME: ";:OT$="":LO$=SP$:HI$="↑":LN=12:GOSUB 1440:F$=OT$
550 IF F$="" THEN RUN
560 R=7:C=0:GOSUB 1280:PRINT"HOW MANY RECORDS (3 TO 719):    {3 SHIFT CRSR LEFT}";
570 OT$="":LO$="0":HI$="9":LN=3:GOSUB 1440:NU$=OT$
580 IF NU$="" THEN F$="":GOTO 530
590 IF VAL(NU$)>719 THEN 560
600 IF NU$="" THEN 530
610 NU=VAL(NU$)+1
620 OPEN15,8,15:OPEN6,8,6,F$+".ORD,S,R":GOSUB1360:IF EN=62THENCLOSE6:GOTO650
630 IF EN=64 THEN PRINT "REPLACE ";F$;"? (Y/N)": GOSUB 900:GOTO 650
640 IF EN>19 THEN 1380
650 PRINT "{2 CRSR DOWN}**FILE BEING CREATED":PRINT "{CRSR DOWN}**PLEASE WAIT"
660 OPEN 2,8,2,+F$+".ORD,L,"+CHR$(127):GOSUB 1360:IFEN>19THENCLOSE2:GOTO 1380
670 RC=1:GOSUB 770:GOSUB 820
680 A=NU:GOSUB 780:PT$=TP$:GOSUB 810
690 HF$=A$+"000002"+A$+"000000000000002":PRINT#2,HF$:REM HEADER FILE
700 RC=NU:GOSUB 770:PT$=TP$:GOSUB 810:PRINT#2,NL$
710 RC=NU:GOSUB 770:PT$=PL$:GOSUB 810
720 LNK$="000000000000000":PRINT#2,LNK$:REM LAST LINK
730 FOR RC=2 TO NU-1:GOSUB 770:A=RC+1:GOSUB 780:GOSUB 820
740 PT$=PL$:GOSUB 810
750 LNK$="000000000"+A$+"000":PRINT#2,LNK$:NEXT RC:REM INTERMEDIATE LINKS
760 CLOSE2:CLOSE15:RUN
770 HB%=INT(RC/256):LB%=RC-(256*HB%):RC$=CHR$(LB%)+CHR$(HB%):RETURN
780 A$=RIGHT$(STR$(A),LEN(STR$(A))-1)
790 IF LEN(A$)<3 THEN A$="0"+A$:GOTO 790
800 RETURN
810 PRINT#15,"P";FC$;RC$;PT$:RETURN
820 R=16:C=2:GOSUB 1280:PRINT "CREATING RECORD #";RC;:RETURN
830 GOSUB 840:CLOSE15:RUN
840 PRINT "{SHIFT CLR}":PRINT TAB(13) "FILE MANAGER":PRINT TAB(8) "{CRSR DOWN}DELETE ORGANIZER FILE"
850 PRINT "{CRSR DOWN}FILE NAME: ";:OT$="":LO$=" ":HI$="↑":LN=12:GOSUB 1440
860 IF OT$="" THEN RUN
870 F$=OT$:OPEN15,8,15:OPEN6,8,6,F$+".ORD,S,R":GOSUB 1360:REM FORCE ERROR 64
880 IF EN<>64THEN CLOSE6:GOTO 950
890 PRINT "{CRSR DOWN}{CTRL RVS OFF}ARE YOU SURE YOU WANT TO DELETE: ":PRINT F$;"? (Y/N)"
900 GET A$:IF A$="" THEN 900
910 IF A$="Y" THEN 930
920 CLOSE15:RUN
930 PRINT#15,"S0:"+F$+".ORD":GOSUB1360:IF EN>19 THEN 1380
940 RETURN
950 CLOSE6:CLOSE15:PRINT" {CRSR DOWN}";F$;" NOT ON DISK--PLEASE TRY AGAIN";:GOTO 1410
960 GOTO 900
970 REM ENLARGE ORGANIZER FILE
980 CLOSE15:PRINT "{SHIFT CLR}":PRINT TAB(13) "FILE MANAGER"
990 PRINT TAB(7) "{CRSR DOWN}ENLARGE ORGANIZER FILE"
1000 PRINT "{CRSR DOWN}FILE NAME: ";:OT$="":LO$=SP$:HI$="↑":LN=12:GOSUB 1440:F$=OT$
1010 IF F$="" THEN RUN
1020 OPEN15,8,15,"I0":OPEN3,8,3,+F$+".ORD,S,R":GOSUB 1360:IF EN<>64 THEN1040
1030 CLOSE3:OPEN2,8,2,+F$+".ORD":GOSUB 1360
1040 IF EN>19 THEN CLOSE2:CLOSE3:GOTO 1380
1050 RC=1:GOSUB 770:PT$=TP$:GOSUB 810:INPUT#2,HF$
1060 FOR J=0 TO 8:HF%(J)=VAL(MID$(HF$,J*3+1,3)):NEXT J:OL=(HF%(0)-1):LM=HF%(3)
1070 FM=HF%(2)
1080 PRINT TAB(8) F$;" HAS";HF%(0)-1;"RECORDS."
1090 PRINT "{CRSR DOWN}HOW MANY RECORDS DO YOU WANT TO ADD?"
1100 R=12:C=0:GOSUB 1280:PRINT "  (";STR$(719-OL);" MAXIMUM) :   {3 SHIFT CRSR LEFT}";
1110 OT$="":LO$="0":HI$="9":LN=3:GOSUB 1440:NU$=OT$
1120 IF(NU$="")+(VAL(NU$)>(719-OL))+(VAL(NU$)<2)THEN F$="":CLOSE2:GOTO 980
1130 NU=VAL(NU$):HF%(0)=HF%(0)+NU:HF%(3)=HF%(0):A=HF%(0):GOSUB 780:H0$=A$
1140 IF FM THEN A=HF%(2):GOSUB 780:H2$=A$:GOTO1160
1150 HF%(2)=OL+2:A=HF%(2):GOSUB 780:H2$=A$
1160 A=HF%(3):GOSUB 780:H3$=A$:HF$=H0$+MID$(HF$,4,3)+H2$+H3$+RIGHT$(HF$,15)
1170 PT$=TP$:GOSUB 810:PRINT#2,HF$
1180 RC=HF%(3):GOSUB 770:PT$=TP$:GOSUB 810:PRINT#2,NL$
1190 RC=HF%(3):GOSUB 770:PT$=PL$:GOSUB 810
1200 LNK$="000000000000000":PRINT#2,LNK$:REM LAST LINK
1210 IF LM=0 THEN1240
1220 RC=LM:GOSUB 770:A=OL+2:GOSUB 780:GOSUB 820:PT$=PL$:GOSUB 810
1230 LNK$="000000000"+A$+"000":PRINT#2,LNK$
1240 PT$=PL$
1250 FOR RC=OL+2 TO HF%(3)-1:GOSUB 770:A=RC+1:GOSUB 780:GOSUB 820:GOSUB 810
1260 LNK$="000000000"+A$+"000":PRINT#2,LNK$:NEXT RC:REM INTERMEDIATE LINKS
1270 CLOSE2:CLOSE15:RUN
1280 REM * MOVE CURSOR R=ROW C=COLUMN *
1290 POKE 781,R:POKE 782,C:POKE 783,0:SYS 65520:RETURN
1300 REM INITIALIZE VARIABLES***
1310 CR$=CHR$(13):BK$=CHR$(20):LT$=CHR$(157):RT$=CHR$(29):PL$=CHR$(112)
1320 TP$=CHR$(1):NL$=CHR$(0):FC$=CHR$(2):SP$=CHR$(32)
1330 PRINT CHR$(8):POKE53281,12:POKE53280,12:RETURN
1340 PRINT "{SHIFT CLR}" TAB(13) "{CTRL WHT}THE ORGANIZER":PRINT TAB(14) "{CRSR DOWN}FILE MANAGER":RETURN
1350 REM CHECK DISK ERROR ***
1360 INPUT#15,EN,EM$,ES,ET:RETURN
1370 REM PRINT DISK ERROR MESSAGE ***
1380 CLOSE15:PRINT "{SHIFT CLR}{6 CRSR DOWN}     DISK ERROR #";EN;"{CTRL RVS ON}";EM$
1390 IF CHAIN THEN PRINT "{CRSR DOWN}  TRYING TO GET MAIN MENU"
1400 PRINT "{5 CRSR DOWN}     PLEASE CORRECT AND TRY AGAIN"
1410 PRINT "{CRSR DOWN}     PRESS [RETURN] TO CONTINUE"
1420 IF PEEK(197)<>1 THEN 1420
1430 RUN
1440 REM INPUT ROUTINE ***
1450 GET IN$: IF IN$="" THEN IF LEN(OT$)<LN THEN PRINT"{CMDR @}{SHIFT CRSR LEFT}";:GOTO 1450
1460 IF IN$=BK$ AND OT$<>"" THEN PRINT BK$;:OT$=LEFT$(OT$,LEN(OT$)-1):GOTO 1450
1470 IF IN$=CR$ THEN PRINT"{CTRL RVS OFF}":GOTO 1510
1480 IF IN$<LO$ OR IN$>HI$ THEN 1450
1490 TS$=OT$+IN$:IF LEN(TS$)> LN THEN 1450
1500 OT$=TS$:PRINT "{CTRL RVS OFF}";IN$;:GOTO 1450
1510 IF LEN(OT$)<LN THEN PRINT" "
1520 RETURN
```

HCM

OUTLINE EDITOR

```
100 REM ***********************************
110 REM *                                 *
120 REM *  ZERO-BASED OUTLINE EDITOR      *
130 REM *  FOR THE COMMODORE 64           *
140 REM *  BY ANDREW VCOMMOS              *
150 REM *  COPYRIGHT (C) 1984             *
160 REM *  ALL RIGHTS RESERVED            *
170 REM *                                 *
180 REM ***********************************
190 REM
200 REM BASIC "OUTLINE" PROGRAM
210 REM
220 CLR:FS=CHR$(147):DIM OS$(710):GOSUB 6700
230 IF OS$(0)="" THEN OS$(0)=FS+"ENTER OUTLINE TITLE:"+CHR$(13)+CHR$(13)
240 GOSUB 6000:PRINT OS$(0):PRINT:PRINT
250 FOR I=1 TO 710:IF OS$(I)="" THEN NEXT:I=1:GOTO 270
260 PRINT OS$(I):NEXT
270 PRINT:PRINT "[ENTER] TO CONTINUE";:GOSUB 6500
280 GOSUB 6000:PRINT "OUTLINE MENU":PRINT:PRINT
290 PRINT "1 - ENTER/EDIT OUTLINE":PRINT
300 PRINT "2 - SAVE OUTLINE TO DISK":PRINT
310 PRINT "3 - LOAD OUTLINE FROM DISK":PRINT
320 PRINT "4 - PRINT OUTLINE":PRINT
330 PRINT "5 - CLEAR OUTLINE":PRINT
340 PRINT "6 - EXIT":PRINT:PRINT
350 INPUT "ENTER CHOICE (1-6)";C$:C=VAL(C$):IF C<1 OR C>6 THEN 280
360 ON C GOTO 370,430,440,460,470,480
370 GOSUB 6000:PRINT "ENTER OUTLINE":PRINT:PRINT
380 PRINT "USE [RETURN] TO ENTER EACH LINE.":PRINT
390 PRINT "TO INDENT, USE [RIGHT] AND [LEFT]":PRINT
400 PRINT "CURSOR KEYS.":PRINT:PRINT
410 PRINT "TO EXIT, ENTER A BLANK LINE.":PRINT:PRINT
420 PRINT "[ENTER] TO CONTINUE";:GOSUB 6500:GOTO 280
430 GOSUB 6000:PRINT "SAVE OUTLINE":PRINT:PRINT:GOSUB 6600:GOSUB 6800:GOTO 280
440 GOSUB 6000:PRINT "LOAD OUTLINE":PRINT:PRINT:GOSUB 6600:GOSUB 6900:GOTO 280
460 GOSUB 6000:PRINT "PRINT OUTLINE":PRINT:PRINT:GOSUB 7000:GOTO 280
470 GOSUB 6000:PRINT "CLEAR OUTLINE":PRINT:PRINT:INPUT "ARE YOU SURE (Y/N)";A$
480 IF A$="Y" OR A$="y" THEN OS$(0)="":GOTO 280
490 REM
500 REM *** INDENT ROUTINE ***
510 REM
520 REM GOSUB 4600:PRINT#2,I$:GOSUB 4700:RETURN
530 REM
540 REM *** OUTDENT ROUTINE ***
550 REM
560 REM GOSUB 4600:PRINT#2,MID$(I$,2):GOSUB 4700:RETURN
570 REM
580 REM *** CHANGE ROUTINE ***
590 REM
600 REM GOSUB 4600:PRINT#2,"  "+I$:GOSUB 4700:RETURN
610 REM
620 REM *** ENTER/EDIT ROUTINE ***
630 REM
640 REM
650 REM
660 REM
670 REM
680 REM
690 REM
700 REM
710 REM
720 REM
730 REM
740 REM
750 REM
760 REM
770 REM
780 REM
790 REM
800 REM
810 REM
820 REM
830 REM
840 REM
850 REM
860 REM
870 REM
880 REM
890 REM
900 REM
910 REM
920 REM
930 REM
940 REM
950 REM
960 REM
970 REM
980 REM
990 REM
1000 REM
1010 REM
1020 REM
1030 REM
1040 REM
1050 REM
1060 REM
1070 REM
1080 REM
1090 REM
1100 REM
1110 REM
1120 REM
1130 REM
1140 REM
1150 REM
1160 REM
1170 REM
1180 REM
1190 REM
1200 REM
1210 REM
1220 REM
1230 REM
1240 REM
1250 REM
1260 REM
1270 REM
1280 REM
1290 REM
1300 REM
1310 REM
1320 REM
1330 REM
1340 REM
1350 REM
1360 REM
1370 REM
1380 REM
1390 REM
1400 REM
1410 REM
1420 REM
1430 REM
1440 REM
1450 REM
1460 REM
1470 REM
1480 REM
1490 REM
1500 REM
1510 REM
1520 REM
```

Continued

PROGRAM LISTING

```
1530 PRINT "  --COPYING RECORDS--";
1540 IF MD=1 THEN HF%(4)=HF:HF%(5)=HL:GOTO 1560
1550 HF%(6)=HF:HF%(7)=HL
1560 IF GC=1 THEN GOSUB 450:GOSUB 1800:RETURN
1570 IF MD=2 THEN GOSUB 1020:GOTO 2210
1580 NR%=LNK%(HL,3):PR%=LNK%(HF,2)
1590 IF PA% THEN GOSUB 1820:GOTO 1680
1600 IF PR% OR NR% THEN 1630:REM PA%=0
1610 GOSUB 1380:PRINT CL$;:GOSUB 5050:HF%(8)=RN:GOSUB 450:TX$="":RN=HF%(8)
1620 LNK%(RN,3)=0:GOSUB 560:GOTO 1810
1630 GOSUB 1760
1640 IF(PR%)*(NR%=0)THEN LNK%(PR%,3)=0:RN=PR%:GOSUB 420:CC%=PR%:GOTO 1810
1650 IF(NR%)*(PR%=0)THENLNK%(NR%,2)=0:RN=NR%:GOSUB420:SF=NR%:HF%(8)=SF:GOTO 1810
1660 LNK%(PR%,3)=NR%:LNK%(NR%,2)=PR%:RN=PR%:GOSUB 420:RN=NR%:GOSUB 420
1670 CC%=NR%:GOTO 1810
1680 IF PR% OR NR% THEN 1700:REM PA%>0
1690 LNK%(PA%,1)=0:RN=PA%:GOSUB 420:CC%=PA%:PA%=LNK%(PA%,0):GN=GN-1:GOTO 1810
1700 GOSUB 1760:IF(PR%)*(NR%)THEN 1740
1710 IF(PR%)*(NR%=0) THEN LNK%(PR%,3)=0:RN=PR%:GOSUB 420:GOTO 1810
1720 LNK%(NR%,2)=0:LNK%(NR%,0)=PA%:RN=NR%:GOSUB 420
1730 LNK%(PA%,1)=NR%:RN=PA%:GOSUB 420:CC%=NR%:GOTO 1810
1740 LNK%(PR%,3)=NR%:RN=PR%:GOSUB 420:LNK%(NR%,2)=PR%:RN=NR%:GOSUB 420
1750 CC%=NR%:GOTO 1810
1760 IF PR% THEN LNK%(HF,2)=0:RN=HF:GOSUB 420
1770 IF NR% THEN LNK%(HL,3)=0:RN=HL:GOSUB 420:RETURN
1780 RETURN
1790 LNK%(TP,4)=NR%:RN=TP:GOSUB 420:IF NR%=0 THEN 1830
1800 IF MD=2 THEN R=SC+2:GOSUB 5790:PRINT SP$;:RETURN
1810 GOSUB 450:SF=LNK%(PA%,1):GOSUB 5430:RETURN
1820 A=HF:GOSUB 1870:REM SEVER LNK%(HL-HF,0)
1830 IF A=HL THEN RETURN
1840 A=LNK%(A,3)
1850 IF A=0 THEN RETURN
1860 GOSUB 1870:GOTO 1830
1870 LNK%(A,0)=0:RN=A:GOSUB 420:RETURN
1880 REM MD=1 AND GC=0 -- DELETE HOLD BUFFER
1890 GOSUB 5850:PRINT "   --DELETING HOLD BUFFER--";
1900 CP%=HF%(5):GOTO 1920
1910 IF LNK%(CP%,3) THEN CP%=LNK%(CP%,3):GOTO 1910
1920 IF LNK%(CP%,1) THEN CP%=LNK%(CP%,1):GOTO 1910
1930 IF LNK%(CP%,2) THEN BT%=LNK%(CP%,2):GOTO 1950
1940 BT%=0
1950 IF LNK%(CP%,4) THEN GOSUB 2060
1960 IF HF%(3) THEN 1980
1970 HF%(2)=CP%:HF%(3)=CP%:GOSUB 2220:GOTO 1990
1980 LNK%(HF%(3),3)=CP%:RN=HF%(3):GOSUB 420:HF%(3)=CP%:GOSUB 2220
1990 IF BT%=0 THEN 2010
2000 LNK%(HF%(3),0)=0:RN=HF%(3):GOSUB 420:CP%=BT%:GOTO 2030
2010 IF HF%(3)=HF THEN LNK%(CP%,0)=0:RN=CP%:GOSUB 420:GOTO 2040
2020 CP%=LNK%(HF%(3),0):LNK%(HF%(3),0)=0:RN=HF%(3):GOSUB 420::LNK%(CP%,1)=0
2030 HF%(1)=HF%(1)-1:GOTO 1920
2040 HF%(1)=HF%(1)-1:GOSUB 5370:RETURN
2050 FORI=0TO11:PRINTI"  ";:FORJ=0TO4:PRINTLN%(I,J);:NEXT:PRINT:NEXT:PRINT HF$
2060 ET=LNK%(CP%,4)
2070 IF HF%(3) THEN LNK%(HF%(3),3)=ET:GOTO 2100
2080 HF%(3)=ET:HF%(2)=ET
2090 FOR I=0 TO 4:LNK%(HF%(3),I)=0:NEXT
2100 RN=HF%(3):GOSUB 420:HF%(1)=HF%(1)-1:HF%(3)=ET:GOSUB 450
2110 FOR I=0 TO 2:LNK%(ET,I)=0:NEXT I:RN=ET:GOSUB 420
2120 IF LNK%(ET,3)=0 THEN RETURN
2130 ET=LNK%(ET,3):GOTO 2070:REM ELSE
2140 FT=HF%(6):LT=HF%(7)
2150 IF HF%(3) THEN LNK%(HF%(3),3)=FT:GOTO 2170
2160 HF%(3)=FT:HF%(2)=FT:FOR I=0 TO 4:LNK%(HF%(3),I)=0:NEXT I
2170 HF%(1)=HF%(1)-1:HF%(3)=FT
2180 IF FT=LT THEN 2200
2190 RN=FT:GOSUB 420:GOSUB 450:FT=LNK%(FT,3):GOTO 2150
2200 LNK%(FT,3)=0:LNK%(FT,2)=0:RN=FT:GOSUB 420:GOSUB 450:RETURN
2210 RETURN:REM --TEXT GRAB--
2220 FOR I=1 TO 4:LNK%(CP%,I)=0:NEXT I:RETURN
2230 REM INSERT LINE
2240 GOSUB 5850::PRINT "   --INSERTING LINE--";:IF MD=2 THEN 2370
2250 TL=LP:IF(LNK%(PA%,1)=SC%(LP))+(HF%(8)=SC%(LP))THEN 2310
2260 NR%=SC%(LP):PR%=LNK%(NR%,2):FOR I=20 TO LP+1 STEP-1:SC%(I)=SC%(I-1)
2270 SC$(I)=SC$(I-1):NEXT:SC%(LP)=0:SC$(LP)=""
2280 GOSUB 5050::LNK%(RN,3)=NR%:LNK%(RN,2)=PR%
2290 LNK%(NR%,2)=RN:LNK%(PR%,3)=RN:CC%=RN:GOSUB 5510:CX=2:GOSUB 3460
2300 TX$=SC$(LP):GOSUB 560:RN=NR%:GOSUB 420:RN=PR%:GOSUB420:RETURN
2310 NR%=SC%(LP):FOR I=20 TO 1 STEP-1:SC%(I)=SC%(I-1):SC$(I)=SC$(I-1):NEXT
2320 SC%(0)=0:SC$(0)="":GOSUB 5050:LNK%(RN,3)=NR%:LNK%(NR%,2)=RN:RE=RN
2330 RN=NR%:GOSUB 420:RN=RE:CC%=RN:GOSUB 5510
2340 IF(PA%=0) THEN HF%(8)=RN:GOSUB 450:GOTO 2360
2350 LNK%(RN,0)=PA%:LNK%(PA%,1)=RN:RE=RN:RN=PA%:GOSUB 420:RN=RE
2360 CX=2:GOSUB 3460:TX$=SC$(LP):GOSUB 560:RETURN
2370 TL=LP:GOSUB 1020:IF LNK%(TP,4)=SC%(LP) THEN 2420
2380 NR%=SC%(LP):PR%=LNK%(NR%,2):FOR I=20 TO TL+1 STEP-1:SC%(I)=SC%(I-1)
2390 SC$(I)=SC$(I-1):LP=I:GOSUB 4810
2400 NEXT:SC%(TL)=0:SC$(TL)="":LP=TL:GOSUB 4810:GOSUB 4950:LNK%(RN,3)=NR%
2410 LNK%(NR%,2)=RN:LNK%(PR%,3)=RN:LNK%(RN,2)=PR%:GOSUB 1020:CX=2:RETURN
2420 NR%=SC%(LP):FOR I=20 TO 1 STEP-1:SC%(I)=SC%(I-1):SC$(I)=SC$(I-1):LP=I
2430 GOSUB 4810:NEXT:LP=TL:GOSUB 4810:SC%(0)=0:SC$(P)="":GOSUB 4950
2440 LNK%(RN,3)=NR%:LNK%(NR%,2)=RN:LNK%(TP,4)=RN:FT=RN:RN=TP:GOSUB 420
2450 GOSUB 1020:CX=2:RETURN
2460 REM PASTE LINE
2470 IF MD=1 THEN HF=HF%(4):HL=HF%(5):GOTO 2490
2480 HF=HF%(6):HL=HF%(7)
2490 IF HF=0 THEN RETURN
2500 GOSUB 5850:PRINT "  --PASTING HOLD BUFFER. PLEASE WAIT--";
2510 NR%=SC%(LP):PR%=LNK%(NR%,2):IF GC THEN 2660
2520 IF(PA%=0)*(PR%=0)*(MD=1) THEN SF=HF:HF%(8)=HF
2530 IF(PA%)*(PR%=0)*(MD=1) THEN LNK%(PA%,1)=HF:SF=HF
2540 IF PR% THEN LNK%(PR%,3)=HF:LNK%(HF,2)=PR%:GOTO 2560
2550 IF MD=2 THEN FT=HF:LNK%(TP,4)=HF:RN=TP:GOSUB 420
2560 IF NR% THEN LNK%(NR%,2)=HL:LNK%(HL,3)=NR%
2570 IF MD=2 THEN RN=HF:GOSUB 420:RN=HL:GOSUB 420:GOTO 2610
2580 LNK%(HF,2)=PR%:LNK%(HL,3)=NR%:RP=HF
2590 LNK%(RP,0)=PA%:RN=RP:GOSUB 420:IF RP<>HL THEN RP=LNK%(RP,3):GOTO 2590
2600 IF PR%=0 AND PA% THEN RN=PA%:GOSUB 420
2610 IF PR% THEN RN=PR%:GOSUB 420
2620 IF NR% THEN RN=NR%:GOSUB 420
2630 IF MD=1 THEN HF%(4)=0:HF%(5)=0:GOSUB 5430:GOTO 2650
2640 HF%(6)=0:HF%(7)=0:GOSUB 1020
2650 GOSUB 450:RETURN
2660 PRINT"{SHIFT CLR}":R=10:C=0:GOSUB 5790:PRINT"THIS IS NOT FUNCTIONAL AT PRESENT"
2670 PRINT"FEATURE IMPLEMENTED IN NEXT ISSUE.":FOR DE=1 TO 2000:NEXT:GOSUB 5430
2680 RETURN
2690 REM HOLD BUFFER DISPLAY
2700 IF HF%(4)=0 THEN RETURN
2710 PRINT"{SHIFT CLR}";:R=0:C=0:GOSUB 5790:PRINT"{CTRL RVSON}HOLD BUFFER DISPLAY --  ";
2720 IF MD=1 THEN PRINT "OUTLINE{CTRL RVSOFF}";:HF=HF%(4):HL=HF%(5):GOTO 2740
2730 PRINT "TEXT ENTRY{CTRL RVSOFF}";:HF=HF%(6):HL=HF%(7)
2740 GOSUB 5850
2750 PRINT "  --GETTING HOLD BUFFER. PLEASE WAIT--";
2760 FOR I=1 TO 22:RN=HF:GOSUB 520:R=I:C=2:GOSUB 5790:PRINT TX$
2770 IF HF=HL THEN 2790
2780 HF=LNK%(HF,3):NEXT I
2790 GOSUB 5850
2800 PRINT "          PRESS {CTRL RVSON}RETURN{CTRL RVSOFF} TO CONTINUE";
2810 GETK$:IF K$<>CR$ THEN 2810
2820 GOSUB 5850::PRINT "  --RETURNING TO EDITOR--";:GOSUB 5430::RETURN
2830 R=24:C=0:GOSUB 5790::PRINT CL$;:GOSUB 5790::RETURN
2840 RETURN
```

Continued

```
2850 DELETE LINE
2860 GOSUB 5850::PRINT " --DELETING LINE--";:TC=SC%(LP)
2870 IF MD=2 THEN 2920
2880 IF(TC<>HF%(4))*(TC<>HF%(5)) THEN 2960
2890 IF HF%(4)=HF%(5) THEN HF%(4)=0:HF%(5)=0:GOTO 2960
2900 IF TC=HF%(4) THEN HF%(4)=LNK%(HF%(4),3):GOTO 2960
2910 IF TC=HF%(5) THEN HF%(5)=LNK%(HF%(5),2):GOTO 2960
2920 IF(TC<>HF%(6))*(TC<>HF%(7)) THEN 2960
2930 IF HF%(6)=HF%(7) THEN HF%(6)=0:HF%(7)=0:GOTO 2960
2940 IF TC=HF%(6) THEN HF%(6)=LNK%(HF%(6),3):GOTO 2960
2950 IF TC=HF%(7) THEN HF%(7)=LNK%(HF%(7),2)
2960 RE=SC%(LP):PR%=LNK%(RE,2):NR%=LNK%(RE,3)
2970 IF(MD=1)*(LNK%(RE,4))THEN CP%=RE:GOSUB 2060
2980 IF MD=1 THEN GOSUB 3200:GOTO 3000
2990 GOSUB 1020
3000 IF(PR%)+(NR%)THEN 3080
3010 IF MD=2 THEN 3060
3020 IF PA% THEN 3040
3030 SC%(0)=0:GOSUB 3220:RN=RE:GOSUB 420:HF%(1)=HF%(1)-1:GOTO 3190
3040 LNK%(PA%,1)=0:RN=PA%:GOSUB 420:CC%=PA%:PA%=LNK%(PA%,0):GN=GN-1:RN=RE
3050 GOSUB 420:HF%(1)=HF%(1)-1:GOTO 3190
3060 SC%(LP)=0:SC$(LP)="":GOSUB 3220:RN=RE:TX$=SP$:GOSUB 560:HF%(1)=HF%(1)-1
3070 GOSUB 450:GOSUB 950:GOSUB 4950:RETURN
3080 IF PR% THEN CC%=PR%:LNK%(PR%,3)=NR%:RN=PR%:GOSUB 420:GOTO 3160
3090 CC%=NR%:LNK%(NR%,2)=0:GOSUB 420:IF MD=2 THEN 3130
3100 CC%=NR%:LNK%(NR%,2)=0:GOSUB 450:SF=NR%:IF PA% THEN 3120
3110 HF%(8)=NR%:GOTO 3190
3120 LNK%(PA%,1)=NR%:RN=PA%:GOSUB 420:GOTO 3190
3130 LNK%(TP)=NR%:RN=TP:GOSUB 420:FT=NR%
3140 GOSUB 3220:RN=RE:GOSUB 420:HF%(1)=HF%(1)-1:GOSUB 450
3150 GOSUB 4920:GOSUB 950:RETURN
3160 IF NR% THEN LNK%(NR%,2)=PR%:RN=NR%:GOSUB 420:GOTO 3180
3170 LNK%(PR%,3)=0:RN=PR%:GOSUB 420
3180 GOSUB 3220:RN=RE:GOSUB 420:HF%(1)=HF%(1)-1:IF MD=2 THEN 3130
3190 GOSUB 450:RN=CC%:GOSUB 5430:RETURN
3200 IF LNK%(RE,1) THEN CP%=RE:GOSUB 1920:HF%(1)=HF%(1)+1:RETURN
3210 RETURN
3220 FOR J=0 TO 4:LNK%(RE,J)=0:NEXT:IF HF%(3) THEN 3240
3230 HF%(2)=RE:HF%(3)=RE:RETURN
3240 LNK%(HF%(3),3)=RE:RN=HF%(3):GOSUB 420::HF%(3)=RE:RETURN
3250 PRINT"{SHIFT CLR}":R=10:C=0:GOSUB 5790::PRINT"THIS IS NOT FUNCTIONAL AT PRESENT"
3260 PRINT"FEATURE IMPLEMENTED NEXT ISSUE.":FOR DE=1 TO 2000:NEXT:GOSUB 5430
3270 RETURN
3280 REM CURSOR UP
3290 IF LNK%(SC%(LP),2)=0 THEN 740
3300 GOSUB 3450::PRINT "--MOVING LINE POINTER UP--";:R=LP+2:GOSUB 5790
3310 IF LP THEN PRINT"{SHIFT CRSR UP}{SHIFT CRSR LEFT}>{SHIFT CRSR LEFT}";:LP=LP-1:RN=SC%(LP):GOTO 700
3320 RN=SC%(0):FOR Z=0TO11:RN=LNK%(RN,2):IF LNK%(RN,2)=0THENSF=RN:LP=0:GOTO 690
3330 NEXT:SF=RN:LP=0:GOTO 690
3340 REM CURSOR DOWN
3350 GOSUB 3450::PRINT "--MOVING LINE POINTER DOWN--";:R=LP+2:GOSUB 5790
3360 IF LP<20 THEN PRINT"{CRSR DOWN}{SHIFT CRSR LEFT}>{SHIFT CRSR LEFT}";:LP=LP+1:GOTO 700
3370 SF=SC%(11):LP=0:GOTO 690
3380 REM BACK A GENERATION
3390 IF PA%=0 THEN 700
3400 GOSUB 3440
3410 CC%=PA%:PA%=LNK%(PA%,0):GN=GN-1:SF=0:LP=0:GOTO 690
3420 REM FORWARD A GENERATION
3430 GOSUB 3440:PA%=SC%(LP):CC%=LNK%(PA%,1):GN=GN+1:SF=0:LP=0:GOTO 690
3440 GOSUB 3450:PRINT " --CHANGING GENERATIONS. PLEASE WAIT--";:RETURN
3450 R=24:C=0:GOSUB 5790:PRINT CL$;:GOSUB 5790:RETURN
3460 REM TEXT EDITOR
3470 IF LEN(SC$(LP))<CX-1 THEN SC$(LP)=LEFT$(SC$(LP)+CL$,CX-1)
3480 GOSUB 5370:T$=SC$(LP):R=24:C=0:GOSUB 5790:PRINT CL$;:GOSUB 5790
3490 IF MD=1 THEN PRINT " --LINE EDIT--";:GOTO 3510
3500 PRINT " --TEXT EDIT--";:
3510 GOSUB 5400:R=LP+2:C=CX:GOSUB 5790:SC$(LP)=T$
3520 IF CX<39 THEN PRINT MID$(T$,CX-1,1);:POKE 212,0
3530 GET K$:IF K$="" THEN PRINT"{SHIFT CRSR LEFT}{CMDR @}{SHIFT CRSR LEFT}";:GOTO 3520
3540 PRINT"{SHIFT CRSR LEFT}";:IF(K$>=SP$)*(K$<ES$)THEN 3680
3550 IF K$=ES$THENRETURN
3560 IFMD=1 THEN 3660
3570 IF K$=DN$ THEN GOSUB 4820:GOTO 3470
3580 IF K$=UP$ THEN GOSUB 4860:GOTO 3470
3590 IFK$=FM$THEN GOSUB 4150:GOTO 3470
3600 IF K$<FU$(1) AND K$>FU$(8) THEN 3660
3610 FOR Z=1 TO 8:IFK$=FU$(Z) THEN 3640
3620 NEXT:GOTO 3660
3630 ON Z GOSUB 3650,3650,1280,1280,2240,2470,2700,2860
3640 GOTO 3470
3650 RETURN
3660 KN=(-1)*((K$=CR$)+2*(K$=BK$)+3*(K$=IS$)+4*(K$=LT$)+5*(K$=RT$)+6*(K$=EL$))
3670 ON KN+1 GOTO 3520,3970,3770,3840,3910,3930,3960
3680 IF CX=38 THEN 3750
3690 IF(CX=2)*(LEN(T$)>1)THEN T$=K$+MID$(T$,2,LEN(T$)-1):GOTO 3730
3700 IF(CX=2) THEN T$=K$+T$:GOTO 3730
3710 IF LEN(T$)=CX-1 THEN T$=LEFT$(T$,LEN(T$)-1)+K$+SP$:GOTO 3730
3720 T$=LEFT$(T$,CX-2)+K$+RIGHT$(T$,LEN(T$)-(CX-1)):GOTO 3730
3730 PRINT K$;:CX=CX+1:IF K$<>CHR$(34) THEN 3510
3740 POKE212,0:GOTO 3510
3750 IFMD=1THENT$=LEFT$(T$,36)+K$+SP$:PRINTK$;:POKE212,0:PRINT LT$;:GOTO 3520
3760 T$=LEFT$(T$,36)+K$:GOSUB 4010:GOTO 3510
3770 IF CX=2 THEN 3520
3780 IF CX<>38 THEN 3810
3790 IF MID$(T$,37,1)<>SP$ THEN T$=LEFT$(T$,36)+SP$:PRINTSP$+LT$;:GOTO 3510
3800 T$=LEFT$(T$,35)+SP$:PRINT K$;:CX=CX-1:GOTO 3510
3810 IF CX=3 AND LEN(T$)=2 THEN CX=CX-1:PRINT K$;:T$=SP$:GOTO 3510
3820 IF CX=LEN(T$)+1 THEN PRINTK$;:T$=LEFT$(T$,CX-3)+SP$:CX=CX-1:GOTO 3510
3830 T$=LEFT$(T$,CX-3)+MID$(T$,(CX-1)):PRINT K$;:CX=CX-1:GOTO 3510
3840 REM INSERT CHARACTER
3850 IF((CX=38)+LEN(T$)>37)*(MD=1) THEN 3520
3860 IF CX=LEN(T$)+1 THEN 3510
3870 T$=LEFT$(LEFT$(T$,CX-2)+SP$+MID$(T$,CX-1)+CL$,38):PRINTSP$+MID$(T$,CX);
3880 IFMD=1THEN 3510
3890 IF(LEN(T$)>37)*(RIGHT$(T$,1)>" ")THENGOSUB 4010:GOTO 3480
3900 GOTO 3510
3910 IF CX=2 THEN 3520
3920 PRINT K$;:CX=CX-1:GOTO 3510
3930 IF CX=38 THEN 3520
3940 IF CX=LEN(T$)+1 THEN PRINT K$;:T$=T$+SP$:CX=CX+1:GOTO 3510
3950 PRINT K$;:CX=CX+1:GOTO 3510
3960 C=2:GOSUB 5790:PRINT LEFT$(CL$,37);:C=2:GOSUB 5790::CX=C:T$=SP$:GOTO 3510
3970 IF MD=1 THEN CX=1:RETURN
3980 RN=SC%(LP):TX$=SC$(LP):GOSUB 560:GOSUB 4820:CX=2:GOSUB 5370
3990 GOTO 3470
4000 REM TEXT WORD WRAP
4010 GOSUB 5850::PRINT " --FORCED WORD WRAP. PLEASE WAIT--";
4020 IFRIGHT$(T$,1)=" " THEN 4120
4030 FOR Z=1 TO 38:IF MID$(T$,Z,1)=CHR$(32) THEN 4050
4040 NEXT:GOTO 4120
4050 FORZ=38TO1STEP-1:IFMID$(T$,Z,1)=SP$ THEN 4070
4060 NEXT:WD$=K$:T$=LEFT$(T$,37):GOTO 4080
4070 WD$=MID$(T$,Z+1,LEN(T$)-Z):T$=LEFT$(LEFT$(T$,Z)+CL$,38)
4080 SC$(LP)=T$:GOSUB 4810:GOSUB 4820
4090 IFZZ=1THENSC$(LP)=LEFT$(WD$+CL$,38):T$=SC$(LP):CX=LEN(WD$)+2:GOTO 4110
4100 GOSUB 2240::SC$(LP)=LEFT$(WD$+CL$,38):T$=SC$(LP):CX=LEN(WD$)+2
4110 GOSUB 4810:GOSUB 5370:GOSUB 5850:PRINTCL$;:T$=SC$(LP):RETURN
4120 GOSUB 4820:IFZZ=1THEN GOSUB 2240
4130 IFK$<>CHR$(32)THENSC$(LP)=K$:GOSUB 4810:CX=3
4140 GOSUB 5370:GOSUB 5850:T$=SC$(LP):RETURN
```

PROGRAM LISTING

Continued

OUTLINE EDITOR *Continued*


```
4150 M$="":N$="":BR=0:GR=0:DN=0:GOSUB102
     0:BR=LN%(TP,4):GOSUB46900:IFDN=1THEN
     4470
4160 GR=LNK%(BR,3):IFGR=0THENRETURN
4170 GOSUB 4710::IFDN=1THEN 4470
4180 SP=37-LEN(M$)
4190 IFSP>1THEN 4260
4200 RN=BR:TX$=M$:GOSUB 560
4210 IFLNK%(BR,3)=0THEN 4470
4220 BR=LNK%(BR,3):GOSUB 46900:IFGR<>BRTH
     EN 4250
4230 IFLNK%(GR,3)=0THEN 4470
4240 GR=LNK%(GR,3):GOSUB 4710:IF DN=1 TH
     EN 4470
4250 GOTO 4180
4260 IFN$<>"" THEN 4300
4270 RN=GR:TX$=N$:GOSUB 560:IFLNK%(GR,3)
     =0THEN 4470
4280 GR=LNK%(GR,3):GOSUB 4710::IFDN=1THE
     N 4470
4290 GOTO 4260
4300 FORZ=1TOLEN(N$)::IFMID$(N$,Z,1)<CHR$
     (33)THEN 4320
4310 NEXT:WD$=N$:GOTO 4340
4320 IFZ=1THENN$=LEFT$(N$,LEN(N$)-1):GOT
     O 4300
4330 WD$=LEFT$(N$,Z-1)
4340 LB=LEN(N$)
4350 IFLB+1<=SPTHEN 4420
4360 IFLNK%(BR,3)=0THEN 4470
4370 BR=LNK%(BR,3):GOSUB 46900:IFDN=1THEN
     4470
4380 IFBR<>GRTHEN 4180
4390 IFLNK%(GR,3)=0THEN 4470
4400 GR=LNK%(GR,3):GOSUB 4710::IF DN=1 TH
     EN 4470
4410 GOTO 4190
4420 IFM$=""THEN M$=WD$:GOTO 4440
4430 M$=M$+""+WD$
4440 IFZ<1THENN$=""+N$:GOTO 4460
4450 N$=MID$(N$,LB+1,LEN(N$)-LB)
4460 RN=BR:TX$=M$:GOSUB 520:RN=GR:TX$=N$
     :GOSUB 520:GOTO 4180
4470 NR=LNK%(BR,3):IFNR=0THEN 4520
4480 LNK%(BR,3)=0:TX$=M$:RN=BR:GOSUB 560
     :LNK%(HF%(3),3)=NR:RN=HF%(3):GOSUB
     420
4490 LNK%(NR,2)=0:HF%(1)=HF%(1)-1:RN=NR:
     GOSUB 420
4500 IFLNK%(NR,3)=0THENHF%(3)=NR:GOSUB 4
     50:GOTO 4520
4510 NR=LNK%(NR,3):GOSUB 4490
4520 FT=LNK%(TP,4):GOSUB 49200:CX=2:LP=0:
     GOSUB 950:RETURN
4530 IF(M$<>"*P")*(M$<>"*B")THENRETURN
4540 BR=LNK%(BR,3):IFBR<>GRTHEN 4570
4550 M$=N$:IFLNK%(BR,3)=0THENDN=1:RETURN
4560 DN=0:GR=LNK%(BR,3):GOSUB 4710:RETUR
     N
4570 RN=BR::GOSUB 520::A$=TX$:GOTO 4530
4580 IFN$<>"*P"ANDN$<>"*B"THENRETURN
4590 BR=LNK%(BR,3):IFBR=GRTHEN 4650
4600 RN=BR:M$=N$:TX$=M$:GOSUB 560:TX$=""
     :RN=GR:GOSUB 560
4610 IFLNK%(GR,3)=0THENDN=1:RETURN
4620 DN=0:BR=LNK%(BR,3):GOSUB 46900:IFLNK%
     (GR,3)=0THENDN=1:RETURN
4630 GR=LNK%(GR,3):GOSUB 4710:IFN$="*B"O
     RN$="*P"THEN 46000
4640 RETURN
4650 IFLNK%(GR,3)=0THENDN=1:RETURN
4660 DN=0:BR=LNK%(GR,3):GOSUB 46900:IF DN
     =1THENRETURN
4670 IFLNK%(BR,3)=0THENDN=1:RETURN
4680 GR=LNK%(BR,3):GOSUB 4710::RETURN
4690 GOSUB 47000:GOSUB 45300:RETURN
4700 RN=BR:GOSUB 520:GOSUB 45300:TX$=TX$:GO
     SUB 47300:RETURN
4710 GOSUB 47200:GOSUB 45800:RETURN
4720 RN=GR:GOSUB 520:GOSUB 64000:N$=TX$:GO
     SUB 47700:RETURN
4730 IFM$=""THENRETURN
4740 FORS=LEN(M$)TO1STEP-1:IFMID$(M$,S,1)
     >CHR$(32)THEN 47600
4750 NEXT:M$=""::RETURN
4760 M$=LEFT$(M$,S):RETURN
4770 IFN$=""THENRETURN
4780 FORS=LEN(N$)TO1STEP-1:IFMID$(N$,S,1)
     >CHR$(32)THEN 47600
4790 NEXT:N$=""::RETURN
4800 M$=LEFT$(N$,S):RETURN
4810 R=LP+2:C=2:GOSUB 57900:PRINT SC$(LP)
     :RETURN
4820 IFLUP<20 THEN 48400
4830 GOSUB 102000::LP=10:FT=SC%(11):FORZ=0
     TO20:SC%(Z)=0:SC$(Z)="":NEXT:GOSUB
     950
4840 LP=LP+1:IFSC%(LP)=0THENGOSUB 49500:S
     C%(LP)=RN:ZZ=1:T$=CL$:RETURN
4850 ZZ=0:T$=LEFT$(SC$(LP)+CL$,38):RETUR
     N
4860 IF LP THEN 49100
4870 GOSUB 102000:IFLNK%(SC%(0),2)=0THENRE
     TURN
4880 FORZ=1TO11:IFLNK%(SC%(0),2)=0THEN 4
     900
4890 SC%(0)=LNK%(SC%(0),2):NEXT
4900 FT=SC%(0):GOSUB 950:LP=0:T$=LEFT$(S
     C$(LP)+CL$,38):RETURN
4910 LP=LP-1:T$=LEFT$(SC$(LP)+CL$,38):RE
     TURN
4920 FORI=0TO20:SC%(I)=0:SC$(I)="":NEXT:
     RETURN
4930 R=24:C=0:GOSUB 57900:PRINT
4940 PRINT-----:RETURN
4950 IF HF%(2)<>0 THEN 49700
4960 GOTO 50600
4970 RN=HF%(2):HF%(1)=HF%(1)+1:HF%(2)=LN
     K%(HF%(2),3):SC%(LP)=RN:RE=RN
4980 SC$(LP)=SP$:IF HF%(2)=0 THEN HF%(3)
     =0
4990 GOSUB 45000
5000 RN=RE:R=LP+2:C=2:GOSUB 57900:PRINT L
     EFT$(CL$,37);
5010 IF(LNK%(TP,4))*(LP)THEN 50300
5020 LNK%(TP,4)=RN:LNK%(RN,0)=TP:RE=RN:R
     N=TP:GOSUB 42000:RN=RE:PR%=0:GOTO 504
     0
5030 PR%=SC%(LP-1):LNK%(PR%,3)=RN:RE=RN:
     RN=PR%:GOSUB 42000:RN=RE
5040 LNK%(RN,2)=PR%:LNK%(RN,1)=0:FORI=3T
     04:LNK%(RN,I)=0:NEXT:GOSUB 42000:RETU
     RN
5050 IF HF%(2)<>0 THEN 51600
5060 PRINT"■SHIFT CLR■":R=6:C=10:GOSUB 5
     7900:PRINT"THE FILE IS FULL":CLOSE2
5070 PRINT"■CRSRDOWN■YOU MUST EITHER EN
     LARGE THE FILE FROM FILE MANAGER
5080 PROGRAM, OR DELETE RECORDS USING THE
     POINTLINE EDITOR."
5090 PRINT"■CRSRDOWN■ PRESS ■CTRL RVSON
     ■RETURN■CTRL RVSOFF■ TO GO BACK TO
     THE EDITOR"
5100 PRINT"■CRSRDOWN■O
     R"
5110 PRINT"■CRSRDOWN■ PRESS ■CTRL RV
     SON■-■CTRL RVSOFF■ TO GO TO THE MAI
     NMENU(197)
5120 A=PEEK(197)
5130 IF A=1 THEN RUN
5140 IF A=57 THEN GOTO 2400
5150 GOTO 51200
5160 RN=HF%(2):HF%(1)=HF%(1)+1:HF%(2)=LN
     K%(RN,3):SC%(LP)=RN
5170 IF HF%(2)=0 THEN HF%(3)=0
5180 GOSUB 45000
5190 R=LP+2:C=2:GOSUB 57900:PRINT LEFT$(C
     L$,37);
5200 IF PA%=0 AND HF%(1)=1 THEN 52900
5210 IF PA%=0 THEN 52500
5220 IF LNK%(PA%,1)<>0 THEN 52400
5230 LNK%(PA%,1)=RN:LNK%(RN,2)=0:RE=RN:R
     N=PA%:GOSUB 42000:RN=RE
5240 IF(PA%)*(LNK%(PA%,1)=RN)THEN 52800
5250 IF(PA%=0)*(LP=0)THEN LNK%(PA%,1)=RN:
     RE=RN:RN=PA%:GOSUB 42000:RN=RE:GOTO 5
     2800
5270 PR%=SC%(LP-1):LNK%(PR%,3)=RN:RE=RN:
     RN=PR%:GOSUB 42000::RN=RE
5280 IF(LP=0)+(LNK%(PA%,1)=RN)THEN PR%=0
5290 LNK%(RN,0)=PA%:LNK%(RN,1)=0:LNK%(RN
     ,2)=PR%:LNK%(RN,4)=0
5300 GOSUB 42000:RE=0:RETURN
5310 PRINT"■SHIFT CLR■GEN:";GN TAB(10) "
     ";"■SHIFT CLR■RIGHT$(HF%(1)),3);
5320 PRINTTAB(19)"FREE:";RIGHT$(STR$(H
     F%(0)-HF%(1)-1),3);
5330 PRINTTAB(28)"R:";RIGHT$(STR$(LP+1
     ),2);
5340 PRINTTAB(33)"C:";RIGHT$(STR$(CX-1
     ),2);
5350 R=1:C=0:GOSUB 57900::PRINT"■CTRL R
     VSON■<";HD$;LEFT$(CL$,38-LEN(HD$))
     "■CTRL RVSOFF■"
5360 RETURN
5370 R=0:C=15:GOSUB 57900:PRINT"■3SHI
     FT CRSRLEFT■";RIGHT$(STR$(HF%(1)),3
     );
5380 C=24:GOSUB 57900::PRINT"■3SHIFT
     CRSRLEFT■";RIGHT$(STR$(HF%(0)-HF%(1
     )-1),3);
5390 R=0:C=30:GOSUB 57900:PRINT RIGHT$(ST
     R$(LP+1),2);
5400 R=0:C=35:GOSUB 57900:PRINT RIGHT$(ST
     R$(CX-1),2);:RETURN
5410 GOTO 53700
5420 REM DISPLAY SCREEEN
5430 PA%=0 THEN HD$=F$:CX=1:IF HF%(1)
     =0 THEN I=0:GOSUB 56100:GOTO 55100
5440 IF PA%>0 THEN RE=RN:RN=PA%:GOSUB 52
     0::RN=RE:HD$=TX$:RN=RE
5450 IF SF=0 AND PA%>0 THEN SF=LNK%(PA%,
     1):GOTO 54700
5460 IF SF=0 AND PA%=0 THEN SF=HF%(8)
5470 IF SF=0 THEN I=0:GOSUB 56100:GOTO 55
     100
5480 RE=RN:RN=SF:FOR I=0 TO 20:GOSUB 520
     :SC%(I)=RN:SC$(I)=TX$:RN=LNK%(RN,3)
5490 IF RN=0 THEN I=I+1:GOSUB 56100:RN=RE
     :GOTO 55100:CX=1
5500 NEXT I:RN=RE
5510 GOSUB 53100:FOR I=0 TO 20:IFSC%(I)TH
     EN R=I+2:C=2:GOSUB 57900:PRINTSC$(I)
     LP=LP-1:T$=LEFT$(SC$(LP)+CL$,38):IFLP
     =0THENRETURN
5520 GOTO 49100
5530 REM DISPLAY SCREEN
5540 PRINT"■CLR■":FORI=0TO20:IFSC%(I)TH
     EN PRINTSC$(I)
5550 NEXT:RETURN
5560 REM DISPLAY SCREEN
5570 PRINT"■CLR■":FORI=0TO20:IFSC%(I)TH
     EN PRINTSC$(I)
5580 NEXT:RETURN
5590 REM DISPLAY SCREEN
5600 PRINT"■CLR■":FORI=0TO20:IFSC%(I)TH
     EN PRINTSC$(I)
5610 NEXT:RETURN
```

Continued

OUTLINE EDITOR *Continued*

COMMODORE 64

```
5520 POKE 212,0:NEXT I:FOR I=0 TO 20
5530 IF SC%(I)=CC% THEN LP=I:R=LP+2:C=0:
     GOSUB 5790::PRINT">";:GOTO 5550
5540 NEXT I::LP=0:CC%=0:R=LP+2:C=0:GOSUB
     5790::PRINT"<";
5550 FOR I=0 TO 20:RE=SC%(I):R=I+2:C=1:G
     OSUB 5790
5560 IF LNK%(RE,1)=0 AND LNK%(RE,4)=0 TH
     EN 5600
5570 IF LNK%(RE,1)=0 THEN PRINT "&";:GOT
     O 5600
5580 IF LNK%(RE,4)=0 THEN PRINT "#";:GOT
     O 5600
5590 PRINT"*";
5600 NEXT:RETURN
5610 FOR J=I TO 20:SC$(J)="":SC%(J)=0:NE
     XT J:RETURN
5620 REM INITIALIZATION
5630 DIM SC$(20):DIM SC%(20):DIM HF%(8):
     DIM FU$(8):SP$=CHR$(32)
5640 CR$=CHR$(13):BK$=CHR$(20):LT$=CHR$(
     157):RT$=CHR$(29):UP$=CHR$(145)
5650 DN$=CHR$(17):NL$=CHR$(0):PL$=CHR$(1
     2):TP$=CHR$(1):PT=1:FC$=CHR$(2)
5660 ES$="←":SR$=CHR$(174):IS$=CHR$(148)
     :FOR I=1 TO 39:CL$=CL$+SP$:NEXT
5670 EL$=CHR$(177):FM$=CHR$(187):K=0
5680 FOR J=0TO1:FOR I=1 TO 7STEP2:K=K+1:
     FU$(I+J)=CHR$(132+(K)):NEXT:NEXT
5690 CX=1:RETURN
```

```
5700 REM INITIAL SCREEN
5710 PRINT CHR$(8):POKE53281,12:POKE5328
     0,.:PRINT" SHIFT CLR CTRL WHT TA
     B(13)" THE ORGANIZER"
5720 PRINT" TAB(12)" CRSR DOWN OUTLINE ED
     ITOR"
5730 R=24:C=4:GOSUB 5790::PRINT" CTRL R
     VSON TO RESSE CTRL RVSOFF -  CTRL RVS
     ON TO RETURN TO MAIN MENU CTRL RVSO
     FF ";
5740 R=6:C=1:GOSUB 5790::PRINT"ORGANIZE
     R FILE NAME: ";
5750 OT$="":LO$="":HI$="↑":LN=12:GOSUB
     1160::F$=OT$
5760 IF F$="" THEN RUN
5770 IF F$=ES$ THEN RETURN
5780 REM CURSOR MOVE ROUTINE
5790 POKE 781,R:POKE782,C:POKE783,0:SYS
     65520:RETURN
5800 R1=R:C1=C:R=23:C=0:GOSUB 5790:FOR D
     1=1 TO 3:PRINT CL$;:GOSUB 5790
5810 FOR D2=1 TO 500:NEXT:PRINT"DISK ER
     ROR #";EN;" CORRECT, AND RE-TRY"
5820 FOR D2=1 TO 500:NEXT:NEXT D1:GOSUB
     5790:PRINT CL$;:R=R1:C=C1:GOSUB 579
     0
5830 CLOSE2:CLOSE15:RUN
5840 CC%=NR%:GOTO 1810
5850 R=24:C=0:GOSUB 5790:PRINT CL$;:GOSU
     B 5790:PRINT
```

HCM

HCM Program Bug

DeBUGS on DisPlay

Each issue, as corrections and/or enhancements to our programs are completed and tested in our programming laboratory, the new version of a program is compared to its previous version by our "cross-checking" computer. A listing of all the differences is produced, transmitted to the computerized typesetter, and formatted in the same fashion as our standard listings.

This procedure for "DeBugs on Display" offers two advantages: (1) a standard presentation for updating your HCM programs that is clear and straightforward, and (2) inclusion of all published changes in "update files" which are placed ON DISK™ at the same time the corrections appear in print. This is of special significance to Apple, IBM, and TI (Extended BASIC programs only) ON DISK™ subsribers, because the correction file can be directly "merged" with the original file—automatically updating it! The procedures for accomplishing this are included with the appropriate media.

We are currently working on an easy method of "update merging" for the Commodore 64, and hope to have it ready soon. (More experienced users can refer to this issue's C-64 Tech Note for one such method—although it may be too complicated for beginners, who may prefer just to key-in the changes.)

If you are going to key-in the corrections from "DeBugs on Display" directly into the original program, follow these steps:

1.) Load the original program into your computer's memory.

2.) Key-in the corrections as directed in the "Program Typing Guide" at the beginning of the Listings section.

3.) Any lines in the listing of corrections that state "***DELETED LINE," are to be deleted from the original program by entering the line number only and pressing either the (ENTER) or (RETURN) key (depending on your computer).

Each set of program corrections is prefaced by an identification bar that tells you the program name, and the computer brand to which the correction applies. Make sure you are working with the right listing to ensure satisfactory results.

TABLUT

from HCM Vol. 4, No. 2

APPLE // Family

```
 160 REM  VERSION 4.2.3
 200 IF  PEEK (103) = 1 AND  PEEK (104)
     = 64 THEN GOTO 2100
 203 POKE 103,1: POKE 104,64: POKE 16384
     ,0: POKE 16385,0: POKE 16386,0
 205 PRINT  CHR$ (4);"RUN TABLUT"
1610 HOME : VTAB 21: PRINT  ILLEGAL MOVE
     , TRY AGAIN.";: PRINT  CHR$ (7);: CH
     R$ (7);: CHR$ (7);: FOR TD = 1 TO 60
     0: NEXT TD: PRINT  CHR$ (7);: GOTO
     520
```

ELEMENTARY ADD & SUBTRACT

from HCM Vol. 4, No. 4

IBM PC & IBM PCjr

```
 150 'VERSION 4.4.2
 160 'IBM PCjr WITH CARTRIDGE BASIC
 165 'FROM DOS 2.1
 170 'IBM PC WITH BASICA
```

MARKET MADNESS

from HCM Vol. 4, No. 4

COMMODORE 64

```
 150 REM VERSION 4.4.2
 485 IF S(S)=0 THEN P(P,S,1)=0
1910 IF TB=1 THEN 1160
1915 RT=1:GOSUB 420:::PRINTLEFT$(S$,23)
     "DEAL IN WHICH STOCK: ";:GOSUB 341
     0
1985 TB=1
2980 PRINT"CASH:  $"RIGHT$(STR$(P(P,0,0)
     ),LEN(STR$(P(P,0,0))))-1);
3060 REM **** DELETED LINE
3070 REM **** DELETED LINE
3080 REM **** DELETED LINE
3090 REM **** DELETED LINE
3100 REM **** DELETED LINE
3110 PRINTPR$;:FL=1:GOSUB 3410:GOTO 420
3170 TB=0:NEXTP:FORC=1TONP:P(C,7,0)=P(C,
     7,0)+(P(C,7,0)*.005)
3220 IFINT(WE/4)<>WE/4THEN 3240
3240 TB=0:NEXTWE
3310 IFP(W,0,0)=5000THENPRINTTAB(11)"$ 0
     EVEN...":GOTO 3340
```

MARKET MADNESS

from HCM Vol. 4, No. 4

APPLE // Family

```
 160 REM VERSION 4.4.2
 405 IF S(I)=0 THEN P(P,I,1)=0
 700 P(P,0,0)=P(P,0,0):P(P,S,1)=P(P,S,1)*SH_(P,S,1)+SH:P(P,S,2)=P(P,S)
1410 IF TB=1 THEN POKE 34,0:GOTO 880
1415 RT$="DEALIN$:S=POKE 34,0:GOTO 880
1455 TB=1
1460 ABS(S)=INT(10*RND(1)-5):PRINT"$";B;"EACH.SOUND
     FAIR?";:GET N$:IF N$="N"THEN
     POKE 34,0:GOTO
2200 PRINT"ASH:$";P(P,0,0);:HTAB 24:
     PRINT"INVEST:";P(P,S,1):N
     EXT:PRINT I
2250 REM ****DELETED LINE
2260 REM ****DELETED LINE
2270 REM ****DELETED LINE
2280 REM ****DELETED LINE
2290 REM ****DELETED LINE
2300 REM ****DELETED LINE
2310 PRINT"$";:GOSUB 2540:POKE 34,0:G
     OTO340
2320 POKE 34,0:TB=0:NEXT P:FOR C=1
     TO7,0):P(C,7,0)+(P(C,7,0)
     ,0):NEXT:IF W=WE THEN 2350
2375 FOR P(W,WZ,1)*S(WZ):NEXT
2410 IF P(W,0,0)=5000 THEN HTAB 11:P
     RINT"$0";"EVEN...";:GOTO 2440
```

SNAP-CALC

from HCM Vol. 4, No. 3

COMMODORE 64

```
 160 REM VERSION 4.3.6
1835 OPEN15,8,15:PRINT#15,"S0:"+DF$+".D"
     :PRINT#15,"I0"
1840 OPEN1,8,8,DF$+".D"+",S,W":GOSUB4670
     :IFEFTHEN CLOSE1:CLOSE15:GOTO 1730
1845 GOSUB1860:CLOSE1:CLOSE15:RETURN
1850 OPEN1,1,1,DF$+".D":GOSUB1860:CLOSE1
     :CLOSE15:RETURN
1975 OPEN15,8,15:PRINT#15,"S0:"+LGCN$+".
     L":PRINT#15,"I0"
1980 OPEN 1,8,8,LGCN$+".L"+",S,W":GOSUB
     4670:IFEFTHEN CLOSE1:CLOSE15:GOTO 1
     730
1985 GOSUB2000:CLOSE1:CLOSE15:RETURN
1990 OPEN 1,1,1,LGCN$+".L":GOSUB2000:CLO
     SE1:CLOSE15:RETURN
2265 OPEN 15,8,15,"I0"
2270 OPEN 1,8,8,DF$+".D"+",S,R":GOSUB 46
     70:IFEFTHEN CLOSE1:CLOSE15:GOTO 216
     0
2275 GOSUB2290:CLOSE1:CLOSE15:RETURN
2300 INPUT#1,JNK$
2405 OPEN 15,8,15,"I0"
2410 OPEN 1,8,8,LGCN$+".L"+",S,R":GOSUB4
     670:IF EFTHEN CLOSE1:CLOSE15:GOTO 2
     160
2415 GOSUB2430:CLOSE1:CLOSE15:RETURN
4660 REM **DISK ERROR ROUTINE**
4670 INPUT#15,EN,EM$,ET,ES:IF EN<20 THEN
     EF=0:RETURN
4680 EF=1:PRINT"■2 CRSR DOWN■CTRL RVS ON■D
     ISK ERROR #";EN:PRINT EM$:PRINT"■C
     TRL RVS OFF■TRY AGAIN."
4690 FOR DE=1 TO 3000:NEXT:RETURN
```

SNAP-CALC

from HCM Vol. 4, No. 3

TI-99/4A

```
 150 REM VERSION 4.3.4
 340 FOR T=G TO LC :: X$="" :: FOR N=G T
     O LC :: DISPLAY AT(3,G):" ";:NEXT
     N :: COLUMN";:COL
 370 GOSUB 480 :: W=V :: IF W=205 THEN M
     =M+G :: IF T>G THEN V=L(M+G) :: W=J(
     V,T-G):ELSE W=0
 410 IF U=204 THEN J(N,T)=J(N,T)/W :: M=
     M+G :: GOTO 430
 530 IF F$(G)="LAST" THEN LC=VAL(F$(4)) :
     :(AV=0 AND LC<F)OR(AV=1 AND LC>B+1)
     THEN 510 ELSE 1 AND AV=0 THEN LC=F-1
     ELSE B :: FOR N=
1010 PRINT #G:TEMP$,LC,A,B :: FOR N=
     1 TO A :: FOR R=1 TO 50 :: PRINT #G:K
     (N) :: NEXT :: O=0
1470 IF (PG-1)*14+M>=MAX(F,LC)THEN 1490
```

THE ELECTRONIC HOME SECRETARY

from HCM Vol. 4, No. 2

IBM PC & IBM PCjr

```
 150 ' VERSION 4.2.3
 160 ' IBM PCjr WITH CARTRIDGE BASIC
 170 ' IBM PC WITH BASICA
 180 DIM A$(5,60),O$(5),C$(6),P1(3),P2(3
     ),B$(2,5):DEF SEG=&HF000:IF PEEK(
     &HFFFE)=253 THEN FLAG=1 ELSE FLAG=0
 190 DEF SEG:SCREEN 0:WIDTH 40:CLS
 790 FOR Z=1 TO 5:A$(Z,M)="~LAST RECORD
     IN FILE":NEXT
1650 F$=INKEY$:IF F$="" THEN 1650 ELSE I
     F F$<"1" OR F$>"5" THEN 1650 ELSE F
     =VAL(F$):G2=0
1660 ***DELETED LINE
1670 ***DELETED LINE
1700 IF A$(F,I)<>A$(F,J) THEN 1720
2250 IF FLAG=1 THEN SOUND ON:BEEP OFF
2400 IF FLAG=1 THEN SOUND OFF:BEEP ON:RE
     TURN ELSE RETURN
```

SPIDER GRAPHICS

from HCM Vol. 4, No. 3

IBM PC & IBM PCjr

```
 150 REM VERSION 4.3.2
 160 REM IBM PCjr CARTRIDGE BASIC
 170 REM WITH 64K MEMORY EXPANSION
 180 REM IBM PC BASICA WITH
 185 REM COLOR/GRAPHICS MONITOR ADAPTER
 187 REM AND COLOR MONITOR
```

MISSILE MATH

from HCM Vol. 4, No. 3

IBM PC & IBM PCjr

```
 160 REM VERSION 4.3.2
 170 REM IBM PC BASICA
```

STADIUM JUMPING

from HCM Vol. 4, No. 4

TI-99/4A

```
 150 REM VERSION 4.4.2
1911 CALL HCHAR(X,Y,102)
1912 X=X+1
```

QUIZ CONSTRUCTION SET

from HCM Vol. 4, No. 5 C-64

Quiz-Make will not accept commas in the actual quiz file. Also, numbers cannot be used in the file name.

QUIZ CONSTRUCTION SET

from HCM Vol. 4, No. 5 IBM PC & PCjr

Quiz-Make will not accept double quotes in the actual quiz file.

QUIZ CONSTRUCTION SET

from HCM Vol. 4, No 5 Apple II Family

Quiz-Make will not accept commas in the quiz file.

TAX DEDUCTION FILER

from HCM Vol. 4, No. 5 TI-99/4A

At the end of Line 580, there should be a space between the word NEXT and the letter Z.

SENSATIONAL SOFTWARE GIVEAWAY

1 YOUR CHOICE OF THIS ISSUE'S PROGRAMS AT A REMARKABLE PRICE— ONLY $4.95!

FOR SUBSCRIBERS & NEWSSTAND PURCHASERS ONLY*

To participate in our monthly Software Giveaway, you need to be a bonafide purchaser of ***Home Computer Magazine*** and fill out the questionnaire on the reverse side.

Have you taken advantage of our Software Giveaway before?
☐ Yes ☐ No

You will receive all the programs ON TAPE™ or ON DISK™ (which have versions for your selected machine) whose listings appear in this issue.

IMPORTANT: *The order form below and the questionnaire on the reverse side must be completed. Tear out this entire page and enclose in an envelope along with $4.95 ($6.95 in Canada & Mexico, $9.95 Foreign Airmail). Payment must be made by check, money order, or VISA/MasterCard. Proof of purchase (subscriber label number, sales receipt or any reasonable facsimile thereof) must also accompany this form. *Non-subscriber and non-purchaser price is $9.95 in the U.S.*

TO PROPERLY HANDLE THE VOLUME OF SOFTWARE REQUESTS, ORDERS MUST BE SENT IN ON THIS FORM, NO PHOTOCOPIES. SORRY, WE CANNOT ACCEPT TELEPHONE ORDERS FOR THIS SERVICE.

*FREE SOFTWARE!

When Subscribing To

2 HOME COMPUTER™ magazine

Subscribe or Renew today, and with your paid subscription you will receive **FREE** software—**ON TAPE™** or **ON DISK™**. With a 1-year subscription to ***Home Computer Magazine*** you get **2 FREE Issues** of **ON TAPE™** or **ON DISK™**. Subscribe for 2 years and receive **4 Issues**. And with a 3-year subscription we'll give you **6 full issues** of this convenient software on cassette tape or floppy disk.

***DON'T HAVE A COMPUTER? TAKE A RAINCHECK!**

We'll give you a raincheck for the FREE software so that when you buy a computer, we will send you your choice of ON TAPE™ or ON DISK™ as a FREE BONUS as stated in this offer.

And You Save Up To 40% Off The Single-Copy Price of the Magazine!

SAVE EVEN MORE!

3 AND ENJOY THE CONVENIENCE OF A PROGRAM SUBSCRIPTION

By subscribing to **ON TAPE™** or **ON DISK™** you will save money off the single-copy price **and** receive the same high-quality programs published in each issue of the magazine—**delivered right to your door!**

This cassette tape or floppy disk program service is the convenient, accurate, and affordable way to save hundreds of typing hours.

The Perfect Addition To Your Magazine Subscription!

3-IN-1 ORDER FORM

PLEASE PRINT

Name _____

Address _____

City _____ State _____ Zip _____

TOTAL _____

☐ Check or Money Order Enclosed

MUST BE IN U.S. FUNDS DRAWN ON A U.S. BANK

Bill my ☐ VISA ☐ MasterCard

Account No. _____

Signature _____ Exp. Date _____

Enclose payment or credit card information and mail entire page with completed form to:

HOME COMPUTER MAGAZINE
P. O. Box 70288, Eugene, OR 97401
—OR—
Use our TOLL-FREE LINE for VISA/MasterCard only:
1-800-828-2212 In Oregon, Alaska, Hawaii Tel.(503)485-8796
Open Monday through Friday—7:00AM to 4:00PM
Pacific Standard Time

THIS SECTION MUST BE COMPLETED
INDICATE YOUR CHOICE OF MEDIA: ☐ ON TAPE™ ☐ ON DISK™
INDICATE WHICH COMPUTER MEDIA IS FOR: (check one box only)
Tape formats not supported for IBM and Apple.
☐ **Apple *II* Family** ☐ **C-64** ☐ **IBM PC** ☐ **IBM PCjr** ☐ **TI-99/4A**
☐ **Please send a raincheck for the FREE software.**

Vol. 5 No. 1

1 **YES!** Send me all the programs in this issue: which have versions for my selected machine. I have indicated my choice of media and have enclosed $4.95 ($6.95 in Canada & Mexico, $9.95 Foreign Airmail).
Proof of purchase and completed questionnaire must be included.

2 **YES!** Enter my subscription to ***Home Computer Magazine*** for the term below:
Please check one: ☐ New ☐ Renewal (include subscriber number below)
Label Number: _____

☐ 1-yr (10 issues) $25	☐ 2-yr (20 issues) $45	☐ 3-yr (30 issues) $63
PLUS 2 FREE	***PLUS 4 FREE***	***PLUS 6 FREE***
ON TAPE or ON DISK	ON TAPE or ON DISK	ON TAPE or ON DISK

Canada add $11; Foreign Surface add $21 for 1-yr magazine subscription. Free software offer available in U.S. & Canada only.

3 **YES!** I want to save time and money. Please enter my program subscription* to **ON TAPE** or **ON DISK** for the term listed below:
☐ 8 ISSUES — *ONLY $38* "An Extension To Your 2 Free Issues of Software"
☐ 10 ISSUES — *ONLY $45* "A Full Year Of Program Convenience"
*Does not include magazine.

Canada add $15 for software subscription.
Software subscription not available in other countries at this time.

Please allow 6-8 weeks for your first issue—Magazine & Media shipped separately.

Satisfaction Guaranteed—or the unfilled portion of your subscription will be refunded, less the cost of any premiums you have received.
Defective media gladly exchanged. NO REFUNDS on media.

SGV2510185

3 FANTASTIC OFFERS

HOME COMPUTER™ magazine

QUESTIONNAIRE

Complete and mail to: Home Computer Magazine • P.O. Box 70288 • Eugene, Oregon 97401

FOR ALL READERS

1. **Where did you obtain this copy of Home Computer Magazine?** ☐Subscriber ☐Supermarket ☐Bookstore ☐Users group ☐Newsstand ☐Computer Store ☐Friend ☐Library ☐Other _____
2. **What types of software are you most interested in?** ☐Educational ☐Entertainment ☐Computer Literacy ☐Household Management ☐Job-Related Applications ☐Business ☐Other _____
3. **Are you** ☐Male ☐Female ☐14 or younger ☐15-24 ☐25-34 ☐35-44 ☐45-54 ☐55+
4. **Annual Household Income?** ☐Under $10,000 ☐$10,000-$14,999 ☐$15,000-$19,999 ☐$20,000-$24,999 ☐$25,000-$29,999 ☐$30,000-$39,999 ☐$40,000-$49,999 ☐$50,000+
5. **Occupation?** ☐Professional ☐Management ☐Teacher ☐Student ☐Other _____
6. **What is your ZIP code?** ☐☐☐☐☐
7. **What is the current month and year?** _____
8. **Do you presently own a Home Computer?** ☐No ☐Yes. It is a ☐TI-99/4A ☐Apple II/II+/IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐Other _____

FOR READERS WHO PLAN TO BUY A HOME COMPUTER

9. **Which model do you think you'll purchase?** ☐Apple IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐TI-99/4A ☐Other _____
10. **When do you expect that purchase to be?** ☐less than 3 months ☐3-6 months ☐7-12 months ☐at least 1 year
11. **What do you anticipate your primary use of a home computer will be?** ☐Entertainment ☐Education ☐Computer Literacy ☐Household Management ☐Job-Related Applications ☐Business ☐Other _____

FOR PRESENT HOME COMPUTER USERS

12. **Which home computer(s) do you currently own?** ☐Apple II/II+/IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐TI-99/4A ☐Other _____
13. **What is the primary use of your home computer?** ☐Entertainment ☐Education ☐Computer Literacy ☐Business ☐Job-Related Applications ☐Household Management ☐Other _____
14. **How often is your computer in use?** ☐Less than 1 hour per week ☐1-4 hours ☐5-10 hours ☐11-15 hours ☐16-20 hours ☐over 20 hours
15. **On the average, about how many program listings in each issue of HCM do you key into your computer and use?** ☐None ☐1 ☐2 or 3 ☐4 or more
16. **What peripherals do you currently use?** ☐Disk System ☐Printer ☐Modem ☐Monochrome/Color Monitor ☐Other _____
17. **What do you expect to buy within the next year?** ☐Software ☐Disk system ☐Printer ☐Modem ☐Books ☐Magnetic Media ☐Monochrome/Color Monitor ☐Furniture & Accessories
18. **How much do you expect to spend on computer-related products during the next year?** ☐Less than $25 ☐$25-$49 ☐$50-$99 ☐$100-$249 ☐$250-$490 ☐$500-$999 ☐$1000-$2499 ☐$2500 or more

OPTIONAL: If you would like to help us by participating in a telephone interview, please include your telephone number (____) - _____ here and the most convenient time you can be reached _____:_____ ☐AM ☐PM

COLLECT ALL BACK ISSUES

HOME COMPUTER™ magazine

Please Print

Name _____

Address _____

City _____ State ____ Zip ____

☐ Check or Money Order Enclosed **Total** _____

MUST BE IN U.S. FUNDS DRAWN ON A U.S. BANK

Bill my ☐VISA ☐MasterCard Date Expires _____

Account No. | | | | | | | | | | | | | | | | |

Tel. No. _____ Signature _____

Enclose payment or credit card information & mail with completed form to:

Home Computer Magazine

P.O. Box 70288 • Eugene, OR 97401

Or use our TOLL-FREE Order Line for VISA/MasterCard orders only:

1-800-828-2212

In Oregon, Alaska, Hawaii Tel. (503) 485-8796

ITEMS	PRICE
Home Computer Magazine Back Issues (Circle Issues Desired) Vol. 4 No. 1 Vol. 4 No. 2 Vol. 4 No. 3 Vol. 4 No. 4 Vol. 4 No. 5	$3.95 each — U.S. $4.50 each — Canada $5.50 each — Foreign Surface $7.50 each — Foreign Air
ON DISK & ON TAPE Back Issues (Circle Issues Desired) Vol. 4 No. 1 Vol. 4 No. 2 Vol. 4 No. 3 Vol. 4 No. 4 Vol. 4 No. 5	$5.95 each — U.S. $7.95 each — Canada $9.95 each — Foreign Air
SAVE EVEN MORE — Order Combined Sets (Circle Magazine & Media Sets Desired) Vol. 4 No. 1 Vol. 4 No. 2 Vol. 4 No. 3 Vol. 4 No. 4 Vol. 4 No. 5	$7.90 each set — U.S. $10.90 each set — Canada $10.90 each set — Foreign Surface

Indicate your choice of media: ☐ON TAPE™ ☐ON DISK™

Indicate which computer media is for: (check one)

☐Apple ☐C-64 ☐IBM PC ☐IBM PCjr ☐TI-99/4A

Defective media gladly exchanged. NO REFUNDS on media.

For more information see inside front cover.

Offer & Prices Subject To Change Without Notice.

The Best Of 99'er —Book & Tape Set— Special Close-Out Offer

See Page 8

See Page 8

Please Print

Name _____

Address _____

City _____ State ____ Zip ____

YES! Please send me *THE BEST OF 99'er ON TAPE* along with my FREE copy of the book *The Best Of 99'er.* and the SPECIAL BONUS (while supplies last) of Simon's Saucer™ and the 99'er Programmer's Guide.

Enclosed is $35. (Shipping and Handling FREE!)

For more information see page 8.

Offer & Prices Subject To Change Without Notice.

Defective media gladly exchanged. NO REFUNDS on book or media.

☐ Check or Money Order Enclosed

Total _____

MUST BE IN U.S. FUNDS DRAWN ON A U.S. BANK

Bill my ☐VISA ☐MasterCard Date Expires _____

Account No. | | | | | | | | | | | | | | | | |

Tel. No. _____ Signature _____

Enclose payment or credit card information & mail with completed form to:

Home Computer Magazine

P.O. Box 70288 • Eugene, OR 97401

Or use our TOLL-FREE Order Line for VISA/MasterCard orders only:

1-800-828-2212

In Oregon, Alaska, Hawaii Tel. (503) 485-8796

Save $ $ $ On BACK ISSUES of 99'er HOME COMPUTER magazine™

SPECIAL CLOSE-OUT PRICES FOR MAGAZINES, DISKS & TAPES NOW IN EFFECT FOR TI-99/4A USERS!

Please Print

Name _____

Address _____

City _____ State ____ Zip ____

☐ Check or Money Order Enclosed **Total** _____

MUST BE IN U.S. FUNDS DRAWN ON A U.S. BANK

Bill my ☐VISA ☐MasterCard Date Expires _____

Account No. | | | | | | | | | | | | | | | | |

Tel. No. _____ Signature _____

Enclose payment or credit card information & mail with completed form to:

Emerald Valley Publishing Co.

P.O. Box 70288 • Eugene, OR 97401

Or use our TOLL-FREE Order Line for VISA/MasterCard orders only:

1-800-828-2212

In Oregon, Alaska, Hawaii Tel. (503) 485-8796

99'er Home Computer Magazine, Disk, & Tape Back Issues

Exclusively For The TI-99/4A Home Computer

ISSUE	Magazine Only	Mag. & Media Set	Media Only (Disk or Tape)	ISSUE	Magazine Only	Mag. & Media Set	Media Only (Disk or Tape)	ISSUE	Magazine Only	Mag. & Media Set	Media Only (Disk or Tape)
Vol. 1 No. 6	NOT AVIALABLE			Mar. '83				Aug. '83			
Nov. '82	NOT AVAILABLE			Apr. '83				Sept. '83			
Dec. '82				May '83				Oct. '83			
Jan. '83				June '83				Nov. '83			
Feb. '83				July '83							

☞ INDICATE CHOICE OF MEDIA: ☐ DISK ☐ TAPE

Place an "X" in the corresponding box for each item you wish to order.

QTY	MAGAZINE PRICES U.S.	Canada	Foreign	MAG/MEDIA SET PRICES U.S.	Canada	Foreign	MEDIA PRICES U.S.	Canada	Foreign
1	$3.95	$4.50	$5.50	$7.50	$9.50	$10.50	$3.95	$5.95	$6.95
3	5.95	7.50	8.95	14.95	17.95	18.95	10.95	13.95	14.95
6	10.95	12.50	13.95	28.50	33.50	34.50	20.95	26.95	27.95
12	21.90	24.00	25.95	49.95	55.95	56.95	39.95	44.95	45.95

Defective media gladly exchanged.
NO REFUNDS on media. SUB TOTAL $ _____

Offer & Prices Subject To Change Without Notice.

For more information see inside back cover.

BCV3510185

GUIDE TO HOME COMPUTER™ magazine READER SERVICES

HOME COMPUTER™ magazine

See Rear Bind-In Card

Subscriptions

See Inside Front Cover

Back Issues

ON TAPE™  ON DISK™ 

See Rear Bind-In Card

This Issue's Software

See Rear Bind-In Card

Program Subscriptions

See Inside Front Cover

Back Issues

99'er HOME COMPUTER magazine™

See Inside Back Cover

Back Issues

See Page 8

The Best Of 99'er

See LISTINGS Contents page

Blank-Media Service

LIMITED SUPPLY

A Back-Issue/Software Bonanza Of 99'er HOME COMPUTER magazine™ At Unbeatable Prices

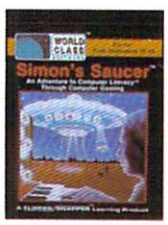
The original 99'er Magazine and 99'er Home Computer Magazine were the forerunners of the present-day Home Computer Magazine. Each of these magazine back issues—exclusively covering the Texas Instruments TI-99/4A—is now available with your choice of either a floppy disk or a cassette tape that contains all the programs in that issue.

AS LOW AS $1.83 EACH MAGAZINE!
See Order Card at Center of Magazine
ABOUT $4 PER SET!

MAGAZINES OR MEDIA MAY BE ORDERED SEPARATELY
To Order, Use Bind-in Card At Center Of Magazine.

✶ SPECIAL MIX OR MATCH BONUS ✶
—WHILE SUPPLIES LAST—

You will receive a FREE Simon's Saucer™ package and a FREE TI-FEST™ commemorative poster when your order includes at least 12 items from this page in any combination of magazines or media.



- A quality, ready-to-run game on cassette tape.
- A durable and attractive ring-binder collector's case for your software library.
- A complete, easy-to-use programming lesson on a deck of colorful flip cards.

SAVE UP TO $44 AND RECEIVE FREE GIFTS WORTH OVER $15!

Issues No. 1-5 OUT OF PRINT
Contents available in book form as "Best of 99'er"—Vol. 1.
See Home Computer Digest or Inquire

ISSUE #6 (Partial Contents)
• How To Produce Sound Effects • Debugging a Game Program • How to Start a User's Group • Verbose: A Speech Vocabulary Expansion Aid • Color Mapping • Dynamic Manipulation of Screen Character Graphics • The Beginner's Guide to Cassette Operation With the Home Computer • Pre-School Block Letters and Data Compaction • Picking the Ponies in TI BASIC • Battle Star Space Game • 3-D Animation on the Home Computer • Programming Tips • Who is LOGO for? • Tower of Hanoi in TI LOGO • A Review of the TI Lesson-Development Software • An Interview with a Game Designer • Learning Assembly Language with a Magic Crayon • and much, much more.

NOVEMBER 1982 (Partial Contents)
• Chatting with Your Micro: Languages for the Home Computer • A Review of the Smith Corona TP-1 Daisy Wheel Printer • The Micro Jukes Arcade Game • A Knight's Tour in TI BASIC • LOGO Has Style • ASPIC: A Language for Children • A p-System Beginners' Tutorial • An Interview with a p-System Pioneer • A Mini-Memory Screen Dump to the Home Computer Printer • Up Scope!—An exciting Undersea Combat Game • Strategy for Munch Man • A Brief Encounter with a TI Hand-Held Computer • 99'er Shopping Bus • A Pocket Battleship • Sub-Programs in Extended BASIC • Arcade & Adventure Game Reviews • and much, much more.

DECEMBER 1982 (Partial Contents)
• Tex-Scribe: A Text Editor for the Home Computer • A Christmas Computer Carol • Managing a Mailing List the Futura Way • Parsec: The Arcade Game • Plotting With the Home Computer—Pixel by Pixel • Preventing the Situation—Oh No! Memory Full • A Colorful Tour of TI-Fest: The Home Computer Show • Santa's Workshop: The Making of a Home Computer • The Turtle Arcade: Movies & Video Games in LOGO • Controlling a BASIC Termite • The 99'er Gold Rush—An Arcade/Adventure in the Home • 99'er Digest of News & Happenings in the TI World • Plus Games, Reviews, and much, much more.

JANUARY 1983 (Partial Contents)
• Computer Assisted Instruction for the Handicapped • p-System Basics • Debugging in LOGO • The Dow-4 Gazelle Flight Simulator • Note Whiz and Pitch Master Musical Game Reviews • Learning With the PLATO Computer Library • Strategies for Adventure Gaming • Death Drones • Using the Line-By-Line Assembler • Close Encounters of the Simon Kind • Electrical Engineering Education Program • Interview With an Arcade Game Designer • TI Invaders • Programming With Pascal • Cyber Dice • News and Happenings in the Home Computer World • Arcade Game Reviews • The Thief Adventure Game • Programming Tips • and much, much more.

FEBRUARY 1983 (Partial Contents)
• Texas Instruments at the Winter Consumer Electronics Show • Home Computer Printers on Review • How to Create Math Daisies in LOGO • Vectors in LOGO • ASPIC: A Language for Teachers • The Joys of Adventuring—Part 3 • Programming Pointers with Chuck-A-Luck—Part 4 • Interview With the Voice of Parsec • Why You Need a Printer for Your Home Computer • Liteline to Titan Space Game • Night Blockade Battleship Game • Tower of Hanoi Pocket Program • Computer Gaming Software Reviews • News of Late Developments in the World of Home Computers • and much, much, more.

MARCH 1983 (Partial Contents)
• An Introduction to the TI-99/2 Basic Computer • The Hex-Bus and the 4/A Connection • Making Your Own Say and Spell Game • Disabled Children Learn and Grow • Super Cataloger—A Review of a Disk Library Utility Program • TI's New CC-40 Compact Computer • Robots and Their Social Impact • Twenty Questions With Robot Redford • The Gravity of LOGO • Joystick Jockey—An Overview of Remote Controllers • Parsec Strategy • Converting Extended BASIC to Assembly Language • Matrix Muncher • Mini Memory Disassembler Utility • Pulling the Shade on Sprites • Letters on LOGO • Tiny Tutorials • Games programs, reviews, and much, much more.

APRIL 1983 (Partial Contents)
• Computer Assisted Savings Planning to Build Your Nest Egg • Tex-Cipher Writes and Decodes Secret Messages • Crossbytes—Computer Vocabulary Crossword Puzzle • Cutting Corners On Your Food Budget Using Coupons • Introducing Financial Planning with Multiplan • The Design Philosophy of the Compact Computer • LOGO Takes On the Popular Fifteen Puzzle • Super Language—Programming Sprites in Mini Memory • Colorful Word World—Reading Readiness for Pre-schoolers • Gameware Buffet's A Maze-ing Boa Alley Game • Giant and Dwarfs Entrapment Game • Game Reviews • Programming Tips • Money Saving Hints • and much, much more.

MAY 1983 (Partial Contents)
• A Consumer's Guide to Word Processing • Word Processing Market Basket • A Generalized Filing Program for VIPs • The Multiplan Medium Balances Your Checkbook and Budget • Activity Accountant Helps School Secretaries with Extracurricular Activities • Maximizing Your Mini Memory's 4K of RAM • Exploring Enhanced BASIC on the Compact Computer • The LOGO Tortoise Debates the BASIC Hare • A Pocket Program to Organize Data with Linked Lists • Mentally Handicapped Learners Team Up with the TI 99/4A • The Wonders of Diskette Storage • Beeline—a Multi-Screen Strategy Game • Lost Ruins—an Archeological Dig Game • 3-D Illusions with Sprites in Depth • Game reviews, Group Grapevine, and much, much more.

JUNE 1983 (Partial Contents)
• Children and Computers Make the 99/4H Connection • Tune Your Guitar with Our TI Tuning Fork • Talk to Your Computer—Voice Technology Is Here • Gameware Buffet's Eat or Be Eaten Aardvark Game • Protect Your Station in the Space Zapper Game • What Multiplan Can and Can't Do • Understanding Inputs and Outputs in Drive For Diskettes—Part 2 • Calculate Loan Schedules on the CC-40 • Go on a LOGO Vacation • Letters on LOGO • A Review of Upper Room Software's Programs for Special Learners • Construct an RS232/Joystick Interface • Group Grapevine • Shopping Bus • A Natural Language Interface for the Professional • Game Reviews • and much, much more.

JULY 1983 (Partial Contents)
• The Evolution of Home Computer Graphics Comes Alive in Graphics Grows Up • Five Data Organizers in Never Out of Sorts • TI & 99'er at the Consumer Electronics Show • WarGames: The Movie and the Book • Editing with Multiplan • The LOGO Logician Presents To Model Is to Learn • LOGO Mosaic Designs Fill the Screen • Your Speech Synthesizer as a Spelling and Foreign Language Teacher • Software for Your Low-cost Printer Port • Gameware Buffet's Treasure Island and the Colorful Switch-A-Row • A Book Review of Learn BASIC for CC-40 Users • 3-D Animation with the TMS 9918A Video Chip • Games Reviews • Group Grapevine • and much, much more.

AUGUST, 1983 (Partial Contents)
• The Home Computer Goes To Work • Bit One, Purl Two at the Fashion Factory • Better Business Bar Graphs in Graphic Persuasion • An Ensemble of Assemblers • Cashflow Helps Money Management • Keystrokes for Thrifty Folks—a Review of Typewiter • Game Reviews of Cavern Quest and Starprobe 99 • Counting Fun for Preschoolers • Peripheral Vision 99 • Mean Machines and Small Potatoes • Multiplan Medium Groups Cells into Rectangles for Easy Text, a LOGO Word Processor • Group Grapevine • Hello, Little Brother—CC-40 Speaks to 99/4A • Plato's Progress Looks at Student Assessment • Pocket Sunrise • Public Investigator • Gameware Buffet's Jungle Jim, Success Formula, and much, much more!

SEPTEMBER, 1983 (Partial Contents)
•Adding on to Your Home Computer • 99'er Directory of Commercially Available Software, Accessories and Peripherals • Peripheral Vision 99 Hardware Reviews • Pocket Sunrise, Part Two in Extended BASIC • Byte Lightning—Mini Memory Plays a Game • TI-WRITER Tutorial • LOGO's Functions, Sets and Turtles • The CC-40 and 99/4A Take the Data, and RUN • PLATO's Progress Looks at Geometry Courseware and the Shape of Things to Come • Gameware Buffet's Challenge of Camelot and The Fly • Game Reviews of Nit Wit and Crime and Punishment • Group Grapevine • 99'er Hall of Fame • 99'er Digest Update on New Products, and much, much more!

OCTOBER, 1983 (Partial Contents)
• Adventures and Fantasy with Your Home Computer • Do-It-Yourself Adventure • Picoprocessor Emulator, a 4-Bit Microprocessor • Have No Fear: Assembly Language Won't Byte, Part 1 • Make Your Mark • Les Izmore and Debug • Computing Cartoon • Once Upon a Tortoise Shell—A LOGO Adventure • Turtle Strut • Number Nibbler for Children • Lots of Plots on Your CC-40 and HX1000 Four-Color Printer/Plotter • Multiplan Medium—Bartender • A Grisly Adventure Bear Hunt • Escape From Wizard's Keep in Extended BASIC • Game Reviews • Shopping Bus • and much, much more.

NOVEMBER, 1983 (Partial Contents)
• Education with your Home Computer • Five Creative Learning Activities for Children • Let's Build America • Have No Fear: Assembly Language Won't Byte, Part 2 • Squeezing the Most Out of TI BASIC • TI-WRITER Tutorial • LOGO Lexicon • Interview with Dale Osborn • TI-WRITER: At Home in the Office • PLATO's Progress • Les Izmore and Debug • The Multiplan Medium • Computer Arrested Instruction—99'er Interviews the Kids • Gameware Buffet: Taco Man and Robo Chase • Game Reviews of Jail Break and Arithmetrek • Hall of Fame • 99'er Digest • and much, much more.

THESE 2 MAGAZINES OUT OF PRINT —TAPES AND DISKS ARE STILL AVAILABLE—

Hurry—Supplies Are Limited. Order Yours Today!

Offer & Prices Subject To Change Without Notice.

ALL PROGRAMS IN THIS MAGAZINE

ONLY $4.95* DELIVERED RIGHT TO YOUR DOOR!

The same high-quality Apple, Commodore, IBM, and Texas Instruments programs with type-in-and-RUN listings in this issue are now available ON DISK™ or ON TAPE™ to newsstand purchasers or subscribers of this magazine.

For only $4.95* postpaid (barely covering the cost of a blank floppy disk or cassette tape), you receive all the programs for your particular brand of computer —Truly A "Software Giveaway!"

To Order, Use The Bind-In Card Inside Rear Cover.

* Current issue Price Only — See Center Bind-In Card For Back Issue Prices. Offer & Prices Subject To Change Without Notice.