

PROOF OF PURCHASE

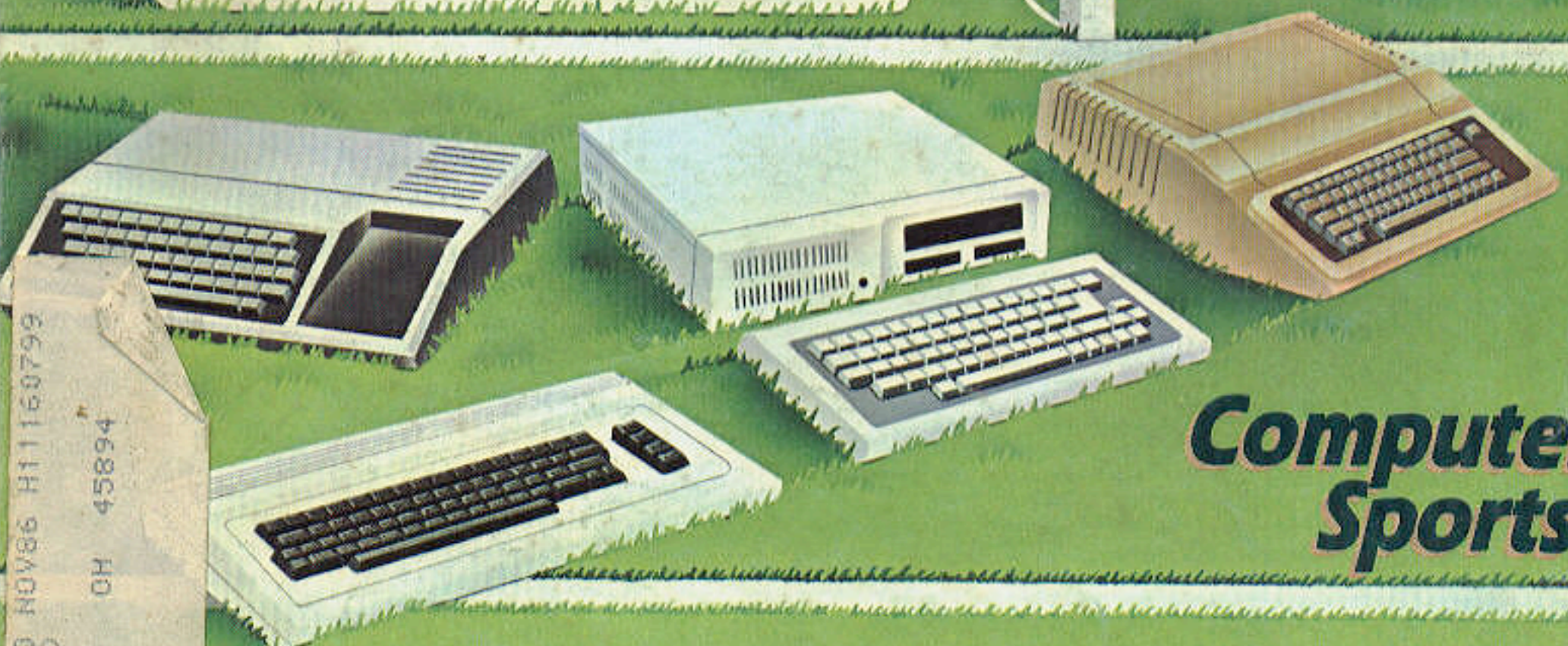
Software Giveaway with Purchase of this Magazine
See back cover

HOME COMPUTERTM magazine

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

Vol. 4 No. 4

\$3.50 In USA
\$4.50 In Canada



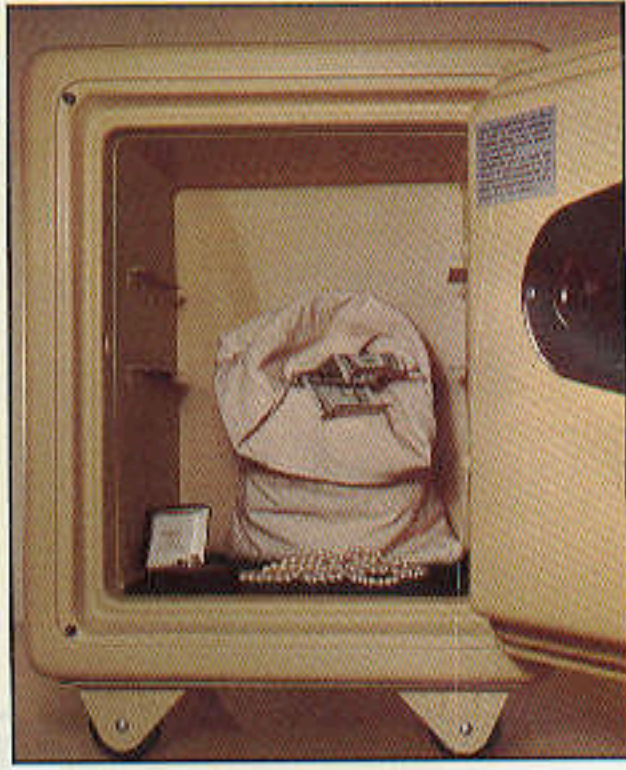
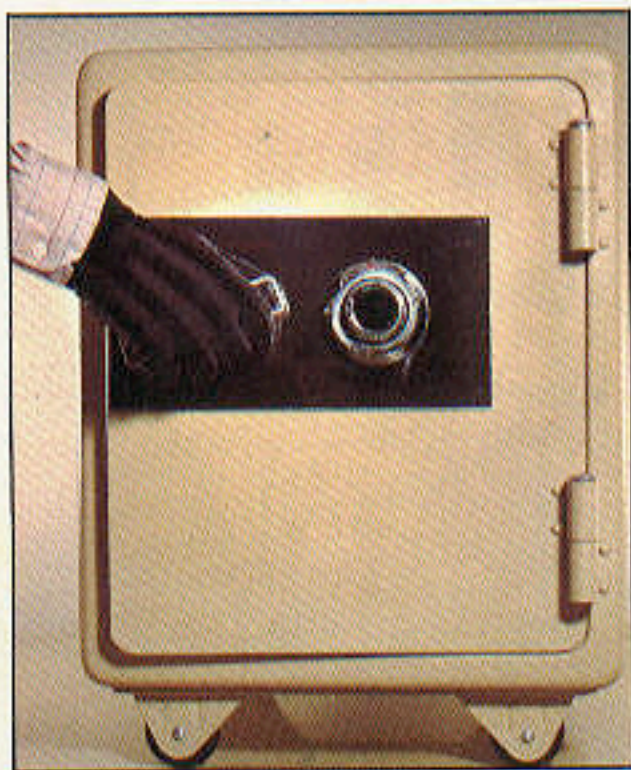
Computer Sports

- The Real Truth About the Apple IIc
 - How To Add a 2nd PCjr Drive
 - Souping Up Your TI-99/4A with Speed BASIC & Double-Drive
 - Basic Sessions with Simon on the C-64
 - Dozens of Ready-to-Run Programs—Filing, Investing, & Sports
- Plus Reviews Galore of the Season's Hottest Sports Games**

45894600CPO 80 NOV86 H11160799
CHARLES GOOD
PO BOX 647
VENEDOCIA OH 45894



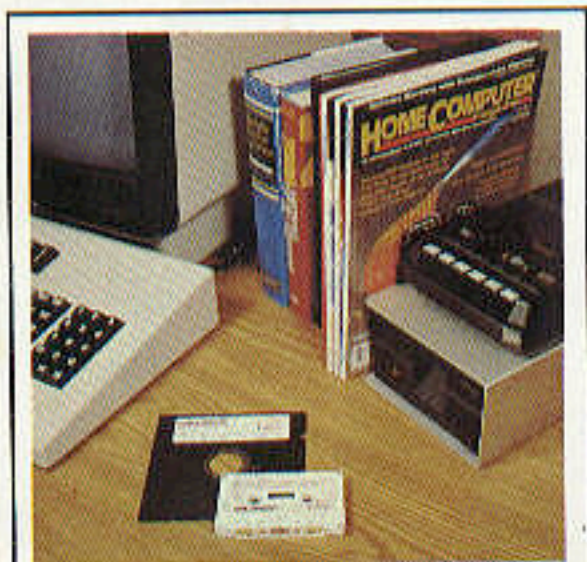
477465



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.

"The 3 Most Common Faults Of Computer Magazines are . . .

1 Diluted Content

For most computer magazines, editorial content plays second fiddle to advertising. In many of today's "successful" publications, strong tutorial and programming material is sparse—lightly sprinkled in between page after page of splashy ads—almost as an afterthought . . . Some of the "most successful" magazines try to compensate for this weak content ratio by printing and binding an issue that is three to four times the amount of paper a reader can comfortably handle and digest. And in the "less successful" publications, the void from unsold ad space is packed with "fluff"—items like re-hashed press releases and photos, filler articles in search of a story, "big name" opinion columns, and old "news" items that contribute virtually nothing to a reader's computer knowledge and enjoyment.



2 Clumsy Design

Unfortunately, layout and design also play second fiddle to advertising space needs. Intervening advertisements break up the "flow" of textual and visual material, making comprehension *more difficult*. Furthermore, all the effort spent in producing clever illustrations, crisp photos, creative typography, and harmonious color usage is often for naught—thanks to visual clashes with adjacent advertising "art."

3 Slanted Focus

Some computer magazines have been known to "do anything to get an ad"—such as publishing "canned" stories touting advertisers, taming down reviews, ignoring competing products from non-advertisers, etc. However, most computer magazines today live by a strong code of ethics. But ethics *isn't* the entire problem. For although a magazine professes to have an editorial content "untarnished" by its advertising content, few (if any) go out of their way to avoid a more subtle bias—editorial calendars dictated by ad-sales staff needs, and the expedient practice of dispensing only *good* reviews because of ". . . too many new products to waste limited magazine space reviewing bad ones."

. . . And Here's What We Just Did About Them."

We on the staff of *Home Computer Magazine* have a standard of "success" that is different from all the rest. When a magazine's success depends upon how well it serves the *reader*, rather than how well it sells the *advertiser*, an amazing thing happens—excellence is inevitable.

All we ask is that you examine our magazine closely. Notice the care we take in balancing: (1) the amount of coverage per computer brand; (2) the article and program mix (of entertainment, productivity, education, and utilities); and (3) the comprehension level for a diverse group of readers. Notice, too, the full measure of high-quality software programs included in each issue. And you won't find anyone else who presents their magazine's software listings in as clear and consistent a format—or who offers *all* the issue's programs on floppy disk or cassette tape for only \$3.95 delivered! As for strong tutorial and "how-to" material, balanced reviews, and elegance of magazine design—you be the judge. We think you'll discover, now more than ever, what we've been saying all along: *Once you compare—there's no comparison.*"

FOR IMMEDIATE RELEASE

COMPUTER MAGAZINE MAKES UNPRECEDENTED MOVE

Emerald Valley Publishing Co. announced today that beginning with its September 1984 issue, *Home Computer Magazine* will no longer carry outside advertising.

According to publisher Gary M. Kaplan, "We want *Home Computer Magazine* to stand out and be recognized as the best publication in its field. By removing the advertising content from the magazine, we have the editorial and artistic freedom to produce a truly unique publication that will set the standard for editorial quality, integrity, and readability for the entire industry."

The new magazine format will allow each article to be presented in its entirety without being interrupted by distracting advertising material. It will also prevent articles from being broken by intervening editorial material resulting from a less-than-flexible layout required to accommodate the needs of advertisers.

"We have thoroughly analyzed the financial considerations of this unprecedented move," Kaplan continued. "Our profitability projection has yielded very favorable results, and undoubtedly reflects the current magazine's uncommon strengths: its extremely high sell-through percentage on newsstands; its large, inexpensively acquired subscriber base; and its companion ON DISK Revue (tm), a spin-off software line recently introduced at the Summer Consumer Electronics Show and slated for retail distribution this fall."

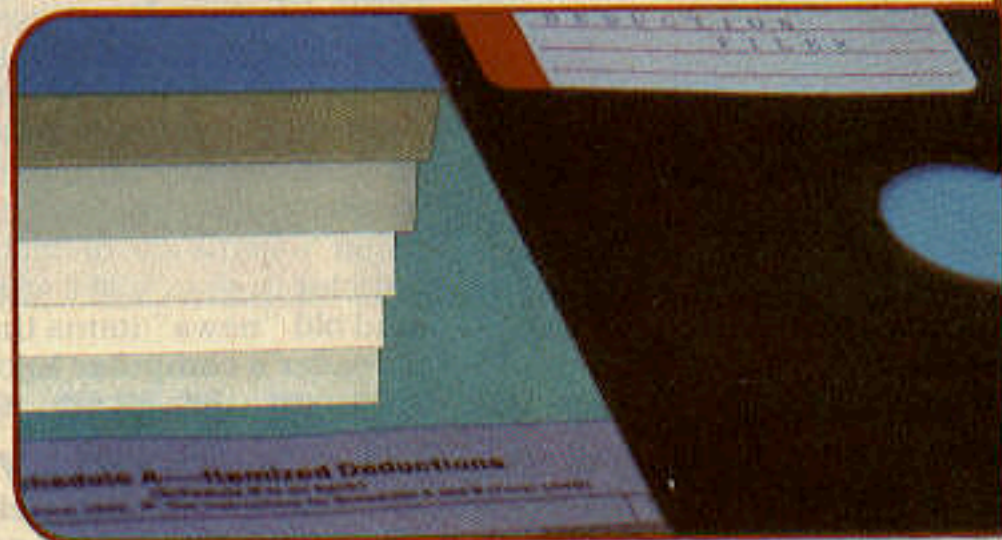
Subscribers to *Home Computer Magazine* will also be kept abreast of additional product availability through a separately mailed, 32-page publication called *Home Computer Digest* (tm). This supplementary publication will be mailed approximately nine times per year and will contain mail-order advertising plus limited editorial material geared to readers who purchase products by mail.

###

Gary M. Kaplan
Publisher

HOME COMPUTER™
















magazine







FEATURES

- 11 //c: The Core of a New Machine** 
 Yet more fruit from the Apple tree. *by Peter Baum and the HCM Staff*
- 20 On the Home Court: Computer Sports Simulation**
 Exploring the ties between computers and athletics. *by Wayne Koberstein*
- 35 Razzle Dazzle** 
 Quick graphics magic for the 99/4A. *by W.K. Balthrop*
- 45 Simon Sez** 
 Plug in 114 new BASIC commands to the C-64. *by W.K. Balthrop*
- 48 Tax Deduction Filer**     
 10-40 good buddy! *by Roger Wood*
- 51 Kaleido Computer**      
 A new medium for an old toy. *by Melody Covington and the HCM Staff*
- 69 Multiplan Medium (part VIII)**     
 Applying "net present value" plus a review of Multiplan for the C-64. *by Patricia Swift*
- 73 Have No Fear: Assembly Language Won't Byte (part IV)** 
 Assembly language on the 99/4A made easy—really! *by Peter Lottrup and the HCM Staff*
- 76 The RS-232 Interface: Your Link to the Periphery**      
 The "standard" explained. *by Patricia Swift*
- 82 One for the Money, Two for the Slow—Adding a Second Drive to the PCjr** 
 And they said it couldn't be done... *by David G. Brader*

GAMEWARE BUFFET™

- 56 Boolean Brain**  
 Wander down logic paths inside your computer. *by W. K. Balthrop*
- 58 Stadium Jumping**     
 This game doesn't include leaping over sports facilities. *by Kent & Kathy Gemmel and the HCM Staff*
- 60 Market Madness**     
 A vigorous exercise in investing. *by Brian Lee and the HCM Staff*
- 68 Elementary Addition and Subtraction**   
 An educational program for the preschool crowd. *by Mark Dewese and the HCM Staff*

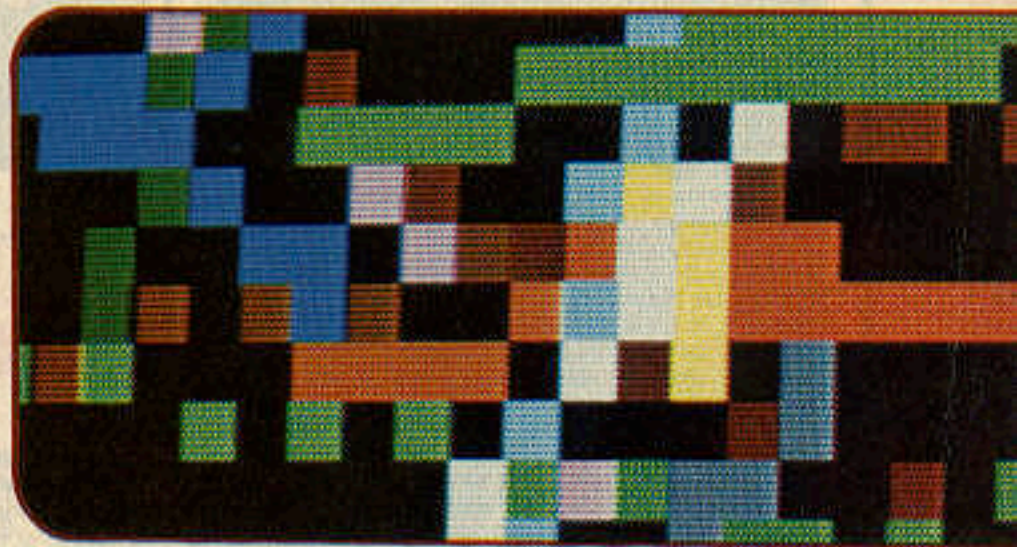
PRODUCT REVIEWS













- 23 International Soccer** 
 Is it a game, or is it real? *A Review*
- 24 One-on-One**  
 Sweat it out with the pros. *A Review*
- 25 Star League Baseball** 
 One, two, three strikes you're out! *A Review*






CONTENTS











SEPTEMBER, 1984 VOLUME 4 NUMBER 4



- 26 HCM Video Olympiad**  
Find out who wins the gold. *A Review*
- 28 Country Club** 
Tee off in your living room. *A Review*
- 29 Pole Position**     
Road racing action. *A Review*
- 30 Bermuda Race**  
Yachting on the open sea. *A Review*
- 31 Buck 'n' Kirk** 
Compare 2 starstruck games: Buck Rogers and Star Trek. *A Review*
- 32 Jr. Addition:
A Review of the
Tecmar JrCaptain Peripheral** 
Juicing up Junior. *A Review*
- 36 SST BASIC Compiler System** 
Give your TI-99/4A supersonic speed. *A Review*
- 40 Two for TI:
A Review of the
CompuAdd Dual Disk-Drive Package** 
Double your storage, double your fun. *A Review*

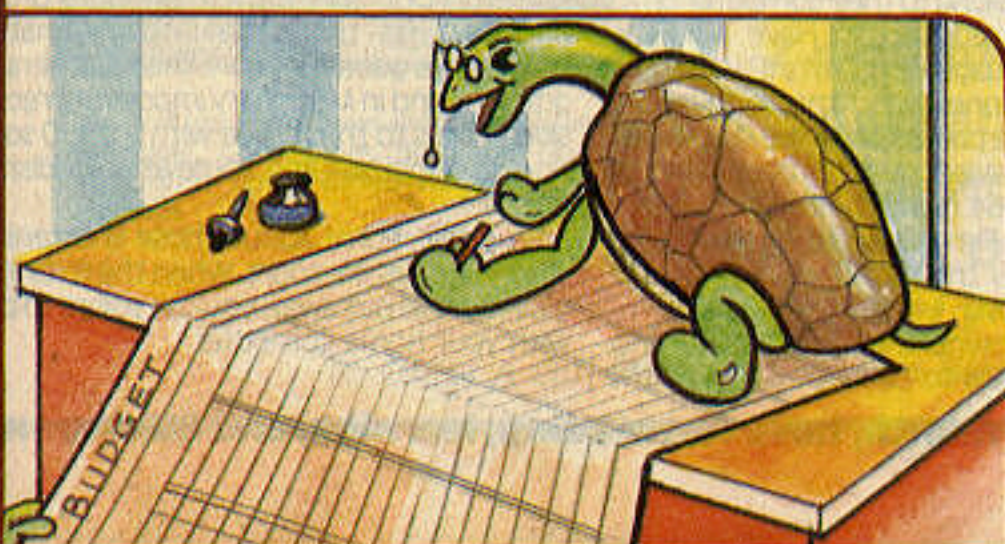
- 42 Mousing Around on the Apple II** 
Mac-magic for the lie. *A Review*
- 44 A Shortcut to 99/4A Printing:
A Review of the Axiom GP-100 TI II** 
Printing without an expansion system. *A Review*
- 70 Multiplan for the C-64** 
How this package compares to other machine versions *A Review*

LOGO TIMES™

- 87 LOGO Spreadsheet**     
If you thought LOGO was just for graphics, read on. *by Rich Haller and the HCM Staff*
- 90 Missionary Impossible**     
Don't rock the boat with these cannibals. *by Roger Kirchner and the HCM Staff*

DEPARTMENTS

- | | | | |
|----|-----------------------|---------------------------|-----------------------------------|
| 3 | On Screen | 93 | Program Listing Contents |
| 6 | Inside/Outside HCM | 130 | DeBugs on Display* |
| 8 | Letters to the Editor | Home Computer Tech Notes: | |
| 19 | HCM Review Criteria | 78 | Apple |
| 38 | Group Grapevine | 79 | TI |
| 46 | Industry Watch | 80 | IBM |
| 47 | Any Questions? | 81 | Commodore |
| 63 | HCM Product News | | |
| 92 | Program Typing Guide | | *See also "Letters to the Editor" |



HOME COMPUTER

Home Computer Magazine (ISSN 0747-055X) is published monthly by Emerald Valley Publishing Co., P.O. Box 5537, Eugene, OR 97405. 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 \$7 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 \$3.00, and \$3.75 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 5537, Eugene, OR 97405. 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 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.

Each separate contribution to this September 1984 issue and the issue as a collective work is Copyright © 1984 by Emerald Valley Publishing Co. All rights reserved. Copying done for other than personal or internal reference use without the permission of Emerald Valley Publishing Co. is prohibited. Requests for special permission or bulk orders should be addressed to the publisher.

Limited License for use of programs in Home Computer Magazine. Emerald Valley Publishing Co. (EVP) is the owner of all rights to the computer programs and software published in this magazine. To allow for use of the software by the purchaser of the magazine, EVP grants to such purchaser only, the limited license to enter these programs into the purchaser's computer, and to place such programs on a diskette or cassette for the purchaser's personal use.

Any other use, distribution, sale, or copying of these computer programs without the written consent of EVP is expressly prohibited and in violation of this limited license and the copyright laws.

Home Computer Magazine, HCM, and Home Computer Digest are trademarks of Emerald Valley Publishing Co.

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

Hendrik Broekhoff, Stephen A. Cordon,
Ann Dahm, Richard Haller, Scott Kindt,
Randy Thompson, Rebecca Van Dalsem

Contributing Editors

Michael Brownsworth, William M. Goodman,
Henry Gorman, Jr., Richard Haller, S. T. Holl,
Roger Kirchner

Asst. to the Publisher Rhea J. Grundy
Production Manager Norman Winney, Jr.
Creative Director Gel-Lei Gom
Typesetting Curtis Byrd

Photography

Nelson Stevens, K.D. Wainsworth

Production Assistant Rachel Knight
Customer Relations Tel. (503) 341-1029
Sharon Hinshaw

Dealer Sales & Distribution Tel. (503) 341-1032
Wendell Anderson, Michael Flagg,
Paula Holland

Main Switchboard Tel. (503) 485-8796



Illustration by Sharon Harker

Outside HCM

Kick-off! That soccer ball is bouncing right into your living room with a new generation of sports simulation games. Today, these games are sporting increased sophistication, inspired by the XXIII Olympiad and the proliferation of computer-graphics in athletic training. Now, with the help of Home Computer Magazine, you can view and participate in this wide world of computer sports—complete with the thrill of a victory RUN, but without the agony of "de feet."

INSIDE HCM

Wait a minute, ole sport! Summer fun doesn't end when the days get shorter—and neither does the fun of summer sports. You may have to go indoors when the ground gets soggy or the wind blows cool, but—you can still enjoy the thrill of swinging a bat, running the 440, or kicking a goal if you're fortunate enough to own a home computer.

This month, Home Computer Magazine features a special review of the new sports simulation games. We kick off this section with *On the Home Court*, a blimp's eye view of this exciting field of gaming software, including an interview with athlete/programmer, Eric Hammond. We hope all the fans—and the budding software artists—out there will find this article and the accompanying reviews stimulating and even inspiring.

But first, take a slice at some fresh produce as we peer into an Apple IIc: *The Core of a New Machine*. Like our report on the IBM PCjr published three issues ago, this article—with extensive documentation and photos—gets right to the heart of the compatibility issue.

Now and then, it helps to have a little advice—in the form of our in-depth tutorials—to guide you along the road to computer glory. This month, HCM has several articles to instruct and encourage the user-in-training. For example, learn about net present value in Part VIII of *Multiplan Medium*, which exercises another aspect of a very useful electronic worksheet. (*Multiplan* was recently issued for the C-64, so Commodore owners will notice a special review of this program accompanying the tutorial.)

BASIC 99/4A programmers who are still intimidated by talking to their computer on its own level are advised to *Have No Fear: Assembly Language Won't Byte (Part IV)*. And for both beginners and more advanced 99/4A users, we pause for a little *Razzle Dazzle*, a short and sweet graphics treat. A similar treat awaits C-64 owners with *Simon Sez*, a mini-tutorial in Simon's BASIC. Both of these articles premier this month as new, regular features in HCM.

For IBM PCjr owners, we have a real granddaddy of a "how-to" article—*One For the*

Money... Two For the Slow: Adding a Second Disk Drive to PCjr. If you're tired of switching disks in and out of Junior, you may well be attracted by this relatively inexpensive alternative. And any computer user can discover the principles of interfacing—or how to hook all those computer gizmos together—with *The RS-232 Interface: Your Link to the Periphery*.

You can also learn a lot from our incisive product reviews, as we examine: *Tecmar Jr. Captain*, a memory-expander—and more—for the PCjr; the *SST Compiler*, taking off from BASIC to the speed of assembly language; dual drives for the TI-99/4A, with our own photo-guide to installation; *AppleMouse II* (with *MousePaint*) for the Apple IIe; and a TI printer by Axiom that doesn't need an expansion box.

With our HCM key-in programs, you can step up to home plate—your computer—and score a RUN every time. From its center-field position in this issue, our BASIC software section covers all the bases: Lead off on tax-time now with *Tax Deduction Filer*, and learn the art of creating colorful graphics patterns with *Kaleido Computer*, a combined program/tutorial. C-64 and TI-99/4A owners can try an adventure in computer education inside the *Boolean Brain*, while Apple and IBM users—who played *Boolean Brain* last month—receive our practical learning aid, *Elementary Addition and Subtraction*. And everyone—on all machines we cover—can practice *Stadium Jumping* (with horses, not over stadiums), or experience a taste of *Market Madness* (stocks, not groceries).

Two LOGO programs complete our line-up: A useful *LOGO Spreadsheet* provides proof positive that this often-underestimated language is good for more than just simple graphics, and in *Missionary Impossible*, recursion comes to the rescue with a LOGO solution to a logical—and somewhat ticklish—puzzle.

So don't worry about those shortening summer days, ole sport. When the mits and gloves are all put away, you can still go to bat with your home computer.

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



*The Perfect Gift For
Any Occasion!*

MAKE SOMEONE HAPPY!

GIVE A GIFT SUBSCRIPTION TO

HOME COMPUTER[™] magazine

As a *bonus* gift we'll send YOU

or the *GIFT RECIPIENT*

2 issues of ON DISK[™] or ON TAPE[™]

ABSOLUTELY FREE!

Comments From Our Happy Readers

"You broke new ground when you introduced your unique typesetting style for your program listings; well, you have done it again by separating all of the listings in the magazine into one section in the center of the magazine. I call that genuine brilliance!
Warren Agee, Livonia, MI

"Well it happened again. Your magazine arrived in the mail, and I'm completely delighted with it."
Chris L. Chaffin, Omaha, NE

"I have subscribed to your magazine since its inception. I must say it has been most informative and has provided me with answers to many of my questions. Your feel for what the public wants is uncanny!"
Larry A. Hamel, Millington, TN

"I just received your August issue. I ordered a 3 year subscription exactly 1 year ago, and I have seen it grow in size and quality. This latest issue, with the separate section of program listings, reaffirms my wise subscription investment."
Mike Oliver, Clarendon Hills, IL

"When I saw the new version of your magazine I was elated! Naturally I subscribed.
Doug Barker, Exeter, CA

"I was a former subscriber to the 99'er Home Computer Magazine and I thought it was great. Then when I got the first issue of the Home Computer Magazine, I was twice as happy. It was a lot of information and great articles. Keep up the good work!"
Jenny Bures, Thousand Oaks, CA

"You have done a superb job of reaching other types of Home Computer enthusiasts and expanding your clientele while not depriving us 99'ers or leaving us by the wayside. The quality of the magazine is unsurpassed by any other, and I have looked at several different magazines! Hats off to you folks for your originality and continued endeavor to reach perfection."
John R. Stewart, Tucson, AZ

"...I am extremely pleased that a magazine such as Home Computer Magazine is around. I find the magazine extremely well written and of invaluable aid. . .keep up the good work with the magazine."
James L. Grigsby, Richmond, KY

Thanks to your thoughtfulness. . .your friends, family, and associates can enjoy a gift that keeps on giving all year through! They'll enjoy

12 BIG Issues of HCM

delivered right to their door each and every month. . .

And you or the gift recipient will receive

2 months of

our magazine program service

**ON TAPE[™] or ON DISK[™]
ABSOLUTELY FREE!!**

To Order A Gift-Subscription, Use Bind-In Card in Center of Magazine.



Dozens of top quality key-in-and-RUN programs for Apple, Commodore, IBM, and Texas Instruments home computers appear in each issue.



FREE software—ON TAPE[™] or ON DISK[™]—the same high-quality programs published monthly in the magazine. This cassette tape or floppy disk program service—normally a \$3.95 per month extra cost—is the convenient, accurate and affordable way to save hundreds of typing hours.

TI/Okidata Link

Dear Sir:

I experienced problems in connecting an Okidata printer to the TI-99/4A computer just as did Mr. Wolly Barabash.

After some trial and error, I found that the printer would work with only a minor change. The connections that I used are:

| TEXAS INSTRUMENTS | | OKIDATA | |
|-------------------|---------------|---------|-------------|
| Term. | Description | Term. | Description |
| 1 | HANDSHAKE OUT | 1 | DATA STROBE |
| 2-9 | DATA | 2-9 | DATA |
| 10 | HANDSHAKE IN | 11 | BUSY |
| 11 | LOGIC GROUND | 10-30 | DATA RETURN |

These are the standard connections that you would expect for connecting a Centronics printer to the TI computer. But to make it work, I connected a 270k Ohm resistor in series with the HANDSHAKE IN - BUSY wire. I installed the resistor inside the Centronics connector at the Okidata end of the cable.

I asked both TI and Okidata if this connection was acceptable, but got no response to that particular question. The printer has been working fine for over a year now.

Thomas Nisius
Westlake, OH 44145

For those users that are do-it-yourselfers, you may wish to give this a try. For those of you who would rather play it safe, read the following two letters.

Dear Sir:

I am writing to congratulate you on your superb magazine and its great coverage of the great TI-99/4A.

This concerns a letter from Wolly Barabash that says he has a problem finding an interface for his Okidata printer. I do not have an Okidata printer, but I have seen the needed cable. He can order it from: Tenex, P. O. Box 6578, South Bend, IN 46660. The part number is 10036, parallel cable-Okidata.

Again, thanks for a great magazine.

Brian Neidig
Taylor, TX 76574

Dear Sir:

In your August 1984 issue a reader expressed his dismay to hooking up his Okidata to a TI. There is a company called Innovative Electronics and Computing, 4150 Fox Street, A-5, Denver, CO 80216. I am presently using an OKI92 with their cable with no problems at all. The cost is about \$25, the stock number is (CBL-1146). I hope this information will help those who are interested in hooking up an OKI to a TI.

Joe Rodomista
Selden, NY 11784

Apple 3d IIe Question

Dear Sir:

I'm frustrated! Being new at computing yet fairly intelligent and resourceful, I was intrigued by the "Apple Graphics in Three Dimensions" of Michael Brownworth's two-part article. Not trusting myself to type in the whole program, I sent for the two disks that you offered. But, try as I might, I have not been able to load a single image onto my screen.

I have a 64K Apple IIe with two disk drives and a monochrome monitor. Here are the problems I

have encountered with your set of programs:

1) Every time I choose one of the three display objects (cube, pyramid, house) from the menu, the screen just goes blank and nothing else will work.

2) When I typed RUN EDITOR 3-D, I got an error message "undefined statement error in 3620." Obviously I fail to understand something very basic and crucial about this program. Is every necessary file provided on the disk that comes with Vol. 4, No. 2 or do I have to go back to the disk for No. 1 and copy something?

The other programs on the disks run okay.

Barbara Matthies
Ames, Iowa 50010

Everything you need is on the disk for Vol. 4, No. 2—it's in the form of a turn-key menu-driven system as described on page 40 in Vol. 4, No. 2 of HCM. The disk does not, however, include the same HELLO program described in the article—if it did, the disk would always start running the Applesoft 3-D system upon booting up. To use any of the 3-D programs, Barbara, all you have to do after booting the disk is type: EXEC LOMEM.EXEC. Then you select what you want to do from the menu that appears and the appropriate program is RUN for you. There is a detailed discussion in the Apple Tech Note on page 99 of the same issue covering some of the aspects of this technique.

So, You Want Less Advertising

Dear Sir:

How about more articles for the PC and Commodore 64 and less advertising?

Compared to the program listings in Compute's Gazette, your listings are not easy to read. The listings are just too small to look at without getting eyestrain. On the positive side, your articles are very well-written and understandable.

One last thought. How about publishing some programs that use Simon's BASIC?

Ira Rubin
New York, NY 10023

Well, I hope we have anticipated your request for less advertising in this issue. As you can see we have eliminated all outside advertising from the magazine. We have anticipated another one of your wishes. Take a look in the Table of Contents and find the special feature entitled, "Simon Sez" to start learning about using Simon's BASIC. As for the matter of listing size, readers tell us that we more than compensate for it with our typeset clarity and quantity. We do also offer readers with poorer eyesight (as well as those who don't have the time or desire to type) the option of very inexpensive prerecorded cassette tapes or diskettes (see back cover).

Easy Script Price Was Wrong

Dear Sir:

I would like to comment on two of the product reviews in the last issue of Home Computer Magazine. The first is the review of Easy Script for the Commodore 64. This was a very good review except for the price that was listed. You gave the price as \$99.95. I believe this is the wrong price. I have been using this program and only paid \$39 for it. I know this is not list price, but I am sure that the price you gave was too high. This has to be the best word processor that you can buy for the price. As stated in the review, the manual is really the only fault with the program. I would like to see listed

in your reviews the language in which the programs are written (machine language for Easy Script).

The second review is for the Home Accountant. I have been using this program on the Commodore 64 since the first of the year. I find several things wrong with this review. I cannot say anything except for the version for the C-64. I have had considerable trouble getting a version that works as it should. After much letter-writing I received another disk and now have a disk where everything works. This is indeed a powerful program as stated in the review, but it is very slow as it is written in BASIC. I believe this should have been in the article. The graph that is shown in the article is nothing like the ones in the C-64 version. The colors used in the C-64 version are very dull and drab. With the colors available, better colors should have been used. I don't know what version you tested, but it was not for the Commodore 64. I will replace this program with a faster one when I can find one.

Now, I would like to comment on the magazine itself. I have always liked the format of the magazine. I find the program listings the easiest of any magazine to key-in.

Jim Gibson
St. Joseph, MO 64504

Easy Script on the Commodore 64, according to the manufacturer, has a suggested retail price of \$54.95. So, Jim, you are right.

It sounds like you had a great deal of bad luck getting a decent copy of the Home Accountant from Continental Software. The C-64 version that we received worked without any of the problems that you describe, other than the speed of the program, which we did mention in the last part of the article. Being written in BASIC, as you state, does make the program slow. In addition, the 1541 Commodore disk adds its own factor of slowness to the operation. The photograph in the article was taken from the Apple IIe version, and not the Commodore 64.

TI Spelling Checker

Dear Sir:

In the August 1984 issue of HCM, William Koseluk asked if anyone knew of a spelling-checker program for TI-Writer. Tom Kirk has developed a program to test the spelling of words; it operates out of the TI-Writer utility option. It comes on two disks and includes a 20,000-word dictionary. In addition, users may add their own dictionaries to the system. However, the program has not been released pending the completion of the instruction manual. Interested users may write for more information to: Tom Kirk, 2606 Ponderosa Drive, Omaha, NE 68123.

Loring Rose
Pantego, NC 27860

Loring, your letter arrived the same day that a package arrived from Mr. Kirk. He has sent us a copy for review of his 99/4 Auto Spell-Check program (which is available from his company Dragonslayer American Software Company, reachable after 5 p.m. Nebraska time at (402) 291-8323). Look for our review of this package in the near future.

More Reusable Peripherals for TI

Dear Sir:

By now I am certain you have received many letters stating the very same thing, but even at the risk

of not being original, let me say it again. What a pleasant surprise to see your magazine on my neighborhood grocer's magazine shelf. It was truly like meeting a friend one thought to have passed away. No offense, but I had visions of your work, if not you personally, being in the literary "Valhalla!" Welcome to the living.

In a more recent issue, John Paulson's letter in Vol. 4, No. 2 regarding peripherals that can be used by both the TI and future state-of-the-art equipment: Taxan's RGB 210 can also handle composite color output as well as RGB. I am currently using such a monitor which sells for under \$250.

Harry Plettner
Schaumburg, IL

Thanks for the tip on the Taxan RGB 210 monitor, Harry. We have also found that the new Sears Performance television is ideal for home computer use. This Sears unit retails for \$349 and it includes both RGB and composite inputs, plus it is a color television to boot. We have been using one in the Editorial Department for several months now and are very pleased with it.

LAST MINUTE DeBUG

In the August Issue of *Home Computer Magazine*, after type-in verification, our paste-up crew inadvertently sliced part of line 780 from the IBM PC and PCjr version of *Spider Graphics*. Line 780 should read:

```
780 IF B$ > "9" THEN COL = ASC(B$) - 55 ELSE  
COL = ASC(B$) - 48
```

Frozen TI Computer in Argentina

Dear Sir:

I have recently taken out a subscription to your magazine and must praise you for its content—first class! I like the ready-to-run programs as I enjoy following the thought and sometimes changing the steps.

Could you please help me on a technical point on my TI-99/4A? Twice now, when making up a program, I have pressed "Enter" at the end of a line and nothing has happened. The screen freezes, the flashing dot disappears, and nothing can be done to move from that position, no matter what key is pressed. The only way out appears to be to switch off, thereby losing all the program typed in, and starting again. On one occasion, I had typed in one of your programs and this happened on the last line! Why does this happen and is there anything that can be done to save the situation without switching off? A friend of mine tells me he has had the same experience.

I shall be very grateful if you would give me some advice on this situation by either writing to me by air mail to the above address or by publishing this letter with your answer in your magazine (which I receive by air mail).

M.K. Atkinson
Buenos Aires, Argentina

Several things can cause a home computer to "freeze" as you described. First is a static electricity discharge from the operator shuffling around in the chair on a carpet. Second, is poor AC power. Third, and least likely, is the actual malfunction of the computer itself. Two inexpensive precautions that you can take immediately are (1) make sure the computer is in a static-free environment and (2) SAVE whatever you're working on every few lines so that if the computer does hang-up you will not lose all

of your work. This is a good habit to get into and we recommend it for anyone using any computer.

Apple DOS or DOS Not

Dear Sir:

I was a subscriber of 99'er Magazine and found it extremely helpful in my education on the TI-99/4A. However, when TI discontinued the machine, I felt the need to move to a computer which was more firmly implanted in the market. So, I moved to an Apple IIe as it seems many of your other readers did.

In the past months I have learned much about Apple and I am a bit disturbed by some of your program listings for the Apple II series. I feel that one of the main purposes of copying a program out of a magazine is to learn the programming techniques of other programmers. Because of this, the listed programs should illustrate the suggested form of the manufacturer (unless it is demonstrating some unique function or capability of the machine). As an example, I cite your recent spreadsheet program (which is quite good), Snap-Calc (HCM August 1984). Under the Apple II series listing, the printer

we recommend something very similar to what you suggest for the program to run properly when loaded under ProDOS.

Conversation About JoyTalk

Dear Sir:

Can you persuade your technical section to come up with lots more projects like "Joytalk" which was very successful? If one was to follow the ads, it would take \$2000 to kit up for the equivalent of Joytalk in Ireland, with PEB, 32K memory, and RS232 before the alphabet could appear on a printer. No hardware or software in Ireland.

Andy MacMahon
County Cork, Ireland

Andy, we've been looking for projects to put in the magazine that are similar to Joytalk. We would welcome any assistance from readers who have built neat little "black boxes" to hook up to their home computers.

TI/C-64 Statistics Software Sought

Dear Sir:

I was initially a subscriber to your magazine when it was devoted entirely to the 99/4 series of computers and was quite satisfied with it. Following the departure of Texas Instruments from the home computer market, I purchased a Commodore 64 with disk drive and monitor in order to ensure that I would be able to take advantage of newer software which might not be produced for the discontinued 99/4A. Your decision to expand the magazine to cover four computers, including the Commodore 64, was greatly appreciated by this subscriber.

I request your help in locating commercial statistics software that I can use on either of these machines for data reduction obtained from a research project to be conducted in my office at this VA. The software which is sold by the manufacturers is not sufficient for my needs as I require programs which would do multiple linear regression, analysis of variance as well as the more basic statistical functions. Any referrals to possible software vendors with these sorts of programs would be appreciated.

David R. Moody
Salem, VA 24153

Another challenge. Do any of you folks out there know of software that will help David out? If so, please drop us a line.

No DMA on PCjr

Dear Sir:

I am directing this letter to Gary Kaplan and/or William K. Balthrop, authors of an excellent article about the PCjr in your Volume 4, No. 1 issue, entitled "A Detailed Look Inside the Peanut's Shell." I am interested in knowing if DMA (Direct Memory Access) is available from any third-party vendors you may know of. I am planning to order Tecmar's Jr. Captain and they do not include it. I am writing IBM additionally as I understand an upgrade version of the PCjr is due. It may or may not have DMA.

Colin Smith
Lewiston, NY 14092

We believe at this time, the only way to get around the DMA problem would be some form of hardware and software modification inside the PCjr itself.

Continued

Letters

TO THE EDITOR

The new version of PCjr that was just announced still does not include DMA (see the IBM new product announcement elsewhere in this issue). We are extremely interested in finding a method to circumvent this DMA problem, as I'm sure you are. If other readers have already met this challenge of the jr., please let us know.

TI Bulletin Boards Abound

Dear Sir:

In response to two of your past letters regarding telecomputing, I submit a list of TIBB(tm) bulletin board systems that are currently operating in the USA and Canada. This program, part X BASIC and part machine language, is for sale by Mr. Ralph Fowler of Kennesaw, Georgia and is also marketed by CR Distributing, who advertised in your August issue, page 153. Mr. Fowler can also be contacted via his own BBS at (404) 425-5254.

Additionally, Mr. John Clulow is presently producing a Bulletin Board System program, which he apparently will distribute to users groups when completed.

An excellent source of BBS listings is available on the ONLINE BBS at (913) 649-1207. Also, the POST section of THE SOURCE frequently advertises new boards in operation. I assume COMPUSERVE's post section would contain listings as well.

T.L. Atkinson
Dartmouth, Nova Scotia

Thanks for the information, Terry. Unfortunately, we don't have space to print the entire list of bulletin boards here. We can include yours however, for our readers. For folks in and around Nova Scotia, the number is (902) 434-3121 to reach Terry's TIBBS.

Dear Sir:

It is very hard for me to believe a magazine of your caliber would just now be hearing about 99/4A Bulletin Board Systems when for quite a few years, Ralph Fowler has been marketing his TIBBS system, and also the CALTEX-99 systems have been on-line for two years.

There are six CALTEX systems and approximately 40+ TIBBS systems on-line nationwide, TIBBS being the system I am running at (415) 355-3092.

Our color TI-Lines TIBBS runs 24 hours and provides continued support for the TI-99/4A and also runs on a 99/4A. The system contains a listing of all other TI boards nationwide and other items of interest.

Mark S. Wong
San Mateo, CA 94404

In the big, wide, wonderful world of home computing there are probably several things which we have overlooked—and we appreciate it when anyone draws them to our attention. Thank you very much, Mark.

Zork on IIc

Dear Sir:

I am thinking of buying the new Apple IIc and I have a few questions about it.

In your magazine you have programs for the Apple II, II+, and IIc. Will these programs work on the IIc? Will you cover the IIc? Also, in Volume 4, No. 2 you reviewed Zork by Infocom. Zork can

be run on the Apple IIe, but can it be run on the IIc? Could you please review the Apple IIc?

Like so many others, I would like to thank you very much for such a fine magazine.

Danny Newton
Phoenix, AZ

Danny, as you can see from our previous issue and this one, we definitely are covering the Apple IIc and the Apple programs that we put in the magazine are tested on the IIc. If there are any differences at all in the operation, they will be so noted in the article. Regarding Infocom's Zork I which we reviewed for the Apple IIe, we have tested it on the Apple IIc and it appears to work on this machine as well.

Eliminate Accidental QUITing

Dear Sir:

I wrote to tell you of the great job you are doing with your magazine. I enjoy typing in your programs.

I also wrote to inform TI-99/4A users who are bothered when they accidentally push the quit command (FCTN +) when they meant to push the plus sign. They can disable the quit command if they have memory expansion and TI Extended BASIC. They can remove it by typing in:

100 CALL INIT
110 CALL LOAD(-31806,16)

Then press RUN and your quit troubles are over.

Jeff Markey
Fort Dodge, IA 50501

Thanks, Jeff, for that tip. You may have just saved the sanity of thousands of Extended BASIC programmers.

Using TI Printer With C-64

Dear Sir:

Since you cover both the TI-99/4A and the Commodore 64 computers, may I appeal to you for advice that no computer store has been able to give?

I have a Commodore 64 computer and a TI-99/4A printer. How can I connect the printer to the computer? As far as I know, the printer does not have a Centronics parallel capability.

It would be nice to be able to use the user's port on the C-64 for a modem and at the same time have the printer hooked up. Is this possible?

Your advice would certainly be welcome.

Edmond Reynolds
Somerset, CA 95684

The Texas Instruments 99/4 Impact Printer has both a serial input and a Centronics parallel input. To use the parallel input with your Commodore 64 (or VIC-20), will require the purchase of a special interface adaptor such as the Cardco Centronics parallel adaptor. This adaptor sells for \$99.95 and is available from Cardco, Inc. 300 S. Topeka, Wichita, KS 67202. Although it is difficult to get it set up initially, the results are worth the effort. We recommend this unit.

Count-Sil vs Snap Calc

Dear Sir:

This letter regards the review of our product Count-Sil in your August issue of Home Computer Magazine.

We are very disappointed not only in the review but the fact it was placed in an issue which contained a six-page write-up of Snap Calc, a free spreadsheet from Home Computer Magazine. The write-up on Snap Calc professed all its great features, its ease of use, etc . . . The review on Count-Sil did not point out the same features plus the many additional benefits we offer. In fact, as a reader of this issue, I would not buy my product when I can either type in Snap Calc or order a copy from Home Computer.

Since we knew a review was going to be in this issue we put a two-color half-page ad in the issue. When I saw the new issue, I was shocked that I would be in competition with Home Computer Magazine and saw that I had just invested in an ad that (based on the issue) I will be lucky to get back 1/10 of my investment.

I wish your magazine had kept us better informed and had recognized the effect of putting the Snap Calc article in with our review.

Sandy Foote, President
Systems Interface, Ltd.
Nepean, Ontario, Canada K2G 3J3

The August issue's theme was productivity. We consider Snap Calc to be productivity software, so it was featured in that particular issue. Likewise, we consider Count-Sil to be productivity software, so it was reviewed in that particular issue. The coverage of Snap Calc was extensive because the entire product is in the magazine. Snap Calc is fully exposed to our readers, and they are free to evaluate the software without risking any money. Our readers can only evaluate Count-Sil by purchasing the product or trusting our review. We feel the Count-Sil review was a fair assessment of your product.

Our primary business interest has always been to secure loyal readers for Home Computer Magazine. Over the years, we've discovered that standard magazine business practices—including the catering to the needs (and sometimes, whims) of advertisers such as yourself—can actually hinder editorial efforts to serve the best interests of a magazine's readers.

Therefore, starting with this issue you're now reading, we have eliminated this potential conflict of interest by not accepting any outside advertising (see editorial by publisher on page 3). It probably won't make you feel any better, Sandy, to hear all this, but we on the staff of this magazine feel very proud to be pioneering this new editorial freedom and reader-service orientation. HCM

Special Announcement:

Home Computer Magazine is looking for "One Liner's".

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 prize and be printed here!

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, Joyystick, Trash Can optional

Performance:
Engrossment:
Documentation:

*** 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.

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?

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•prēs•teh•dī•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.



ON THE HOME COURT

Computer Sports Simulations

Sweat socks lying in the corner, dirty shorts and jersey adding their special aroma to the darkened room—these are familiar props to the “workout” generation. But what’s this eerie glow? Ahhh, it’s a video screen, with little figures running about, throwing, catching, making a steal . . . Not real athletes here, but cartoon-like players magically responding to the commands of the human who sits on the edge of a chair—sweating, flushed, and totally involved.

by **Wayne Koberstein**
HCM Staff

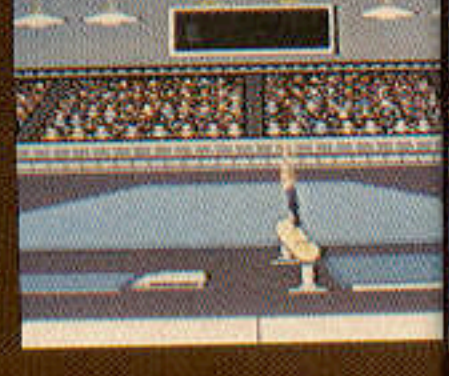
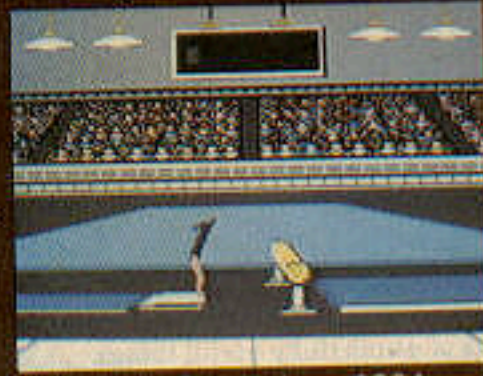
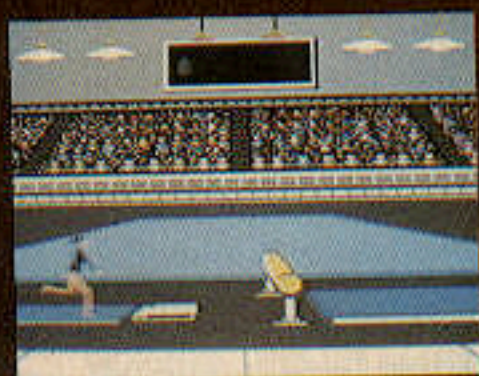
Since the invention of television, many sports fans have spent a lot of time watching rather than playing. You might think computer sports are another lazy, vicarious way to play. Sweatless and safe, replacing gloves and bats with joysticks and fire buttons, sports simulation games may indeed lead to greater indolence. But is this really true? Ask 20-year-old Eric Hammond, who—with the help of two star athletes, Dr. J and Larry Bird—created *One-on-One*. Eric has been programming since the age of 16. Basketball, however, may be his real love. Talking to this young man will dispel any myths you might have about the stereotypical “nerd hacker.”

Computers and sports have intermingled since this new technology first entered the public arena. Initially, the machines simply kept records and figured

averages. Eventually, they became even better at it than everyone’s Uncle Henry—that veritable walking library of sports trivia. Nowadays, as shown dramatically in commercials for the 1984 Olympics, computers actually teach people how to run, jump, and throw better. Trainers at all levels of virtually every sport are using software designed to regulate daily routines, and are comparing an athlete’s actual performance to an ideal performance generated by a computer.

From Pong . . .

As computers have entered the sports world, sports have entered the home court in a new form of video magic. When *Pong* served up the first volley of video games in 1972, our fundamental relationship to television radically changed. Suddenly, we could be more than just passive observers—we could make something happen on the screen! Of course, what we





could make happen was rather crude, and the device we used then was limited to its one simple function, but the basic appeal of video games was the same then as it is now: We can take control of the action and create a unique performance each time we play.

From *Pong* to *One-on-One*, sports games have often led the video arcade field. Both individual and team sports are natural subjects for this medium, with rules long "play-tested" and a wealth of tradition and built-in excitement. Sports simulation is a logical step forward in the evolution of both sports and computers, and its benefits are not confined to one field or the other.

We have seen a continuous chain of improvements in gaming technology since video games helped introduce the concept of the home computer. Early sports simulation games took the rules of a given sport and boiled them down to a basic framework. Stick figures or even little dots represented players; the final scheme often resembled the crude sketches coaches draw to depict plays. Early games were also like chess, in which position is more important than motion. Today's games combine real-time action, life-like animation, and quick feedback between screen and joystick to create a sense of tactile control—of actually performing in the particular sport. Visual excitement and humor, surprises, and sometimes elaborate strategies conjure up the illusion of being there—*inside the event*.

...To One on One

More memory, better graphics, and faster processors in the home models have brought many games (as well as prodigal sons and daughters) from the arcades into our living rooms. That is one side of the change. On the other side are steady advances in the programmers' art. In today's market, a successful video game must perform at a high tech-

nological level, and it must do so in a way that is both original and clever. Just as Olympic athletes defy the limits of their own bodies, software artists sometimes transcend the context of the machine—scoring victories through the sheer strength of their intelligent ideas. Nothing becomes boring so quickly as a game whose fancy new graphics mask a tired old algorithm.

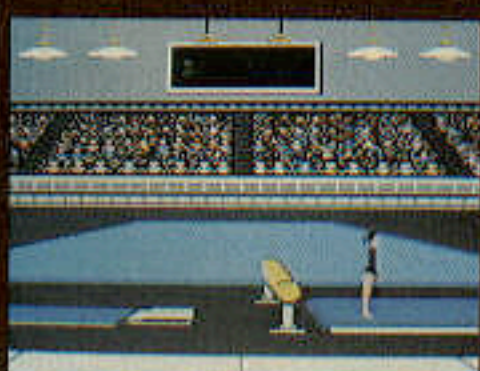
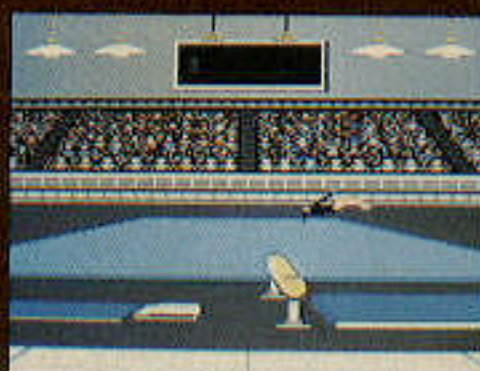
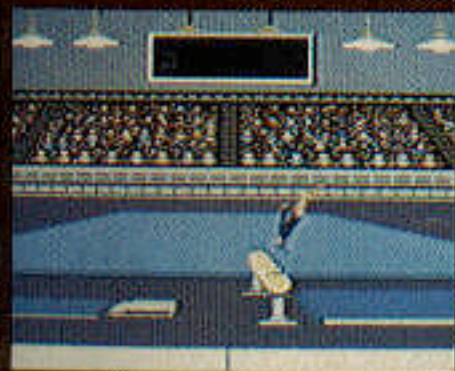
Sports simulation games may have also more relevance to "real" sports than most of us now realize. As trainers develop more techniques for the computer as a training instrument, and as game programmers borrow from these techniques and apply them to simulations, these games will have increasing value to the serious athlete. Every new medium has helped athletes better understand their sport by providing a new perspective. Merely watching great players in action—which through film and TV, more people have a chance to do—can greatly improve a learning player's game. But imagine the possibilities

"When *Pong* served up the first volley of video games in 1972, our fundamental relationship to television radically changed."

in being able to "play" against moves borrowed from the best players in any given sport. Kinetic training of this kind in the home—imparting good timing, reflexes, and a sense of proper motion—

could inspire a whole new generation of athletes.

Although programmers have made good use of their improved palette, they have not fully exploited its potential. Many home video games have reached the technical limits of speed and graphics of the Commodore or Apple II family computers. But other, new machines such as the Macintosh or PCjr have capabilities beyond those that programmers usually work with. For example, the Commodore and Apple II models have almost identical processors: the 6510 on the C64, the 6502 on the Apple IIe, and the 65C02 on the IIc. PCjr uses the 8088, a 16-bit chip that runs about 4-1/2 times faster than 8-bit processors on the other machines. PCjr also sports much higher resolution graphics and a beautiful selection of colors. But programmers have been slow to write for the IBM model—perhaps because they are simply





more accustomed to the old chips, or because IBM itself has not encouraged this kind of development. [Also from this standpoint, it's a shame that Texas Instruments withdrew from the home computer consumer marketplace prior to introducing its model 99/8—a powerful 16-bit machine with a new high-speed microprocessor (TMS9995) and abundant RAM. It just might have been the "ultimate" home machine for sports simulations.—Ed.]

And Beyond . . .

In the near future, we will probably continue to see improvements in sports games programs. Expect to see more involvement of experienced athletes in software design, more clever strategies, more natural on-screen motion, better graphics, sound, speed, and performance. Even with the existing technology of the "older" machines, there is plenty of room for refinement. In the far future of sports simulation, one thing seems likely: In our lifetimes, today's games will seem more primitive than *Pong* now does. Home computers will grow in memory and processing speed, and we will find new ways to interact with them. What else?

We will probably not have to wait another 6000 years to see the kind of moving hologram depicted in the short video game sequence of *Search for Spock*. By then, the joystick will be an ancient artifact, and the hologram replaced by something even scriptwriters have difficulty imagining.

. . . Your opponent appears in the playing area. He is a figure out of the distant past: Julius Irving, one of the greatest players in the ancient game of basketball. He seems so real! With a smile, he reaches out to shake your hand as you step onto the court. His hand is warm. This is perfect! His every move has been carefully reconstructed from the central archive files. They have missed nothing. And now it's your turn to go one-on-one against the best. Sensing each move you make, the computer responds with a countermove by the great Dr. J. Soon you are sweating, flushed, and totally involved . . .

HCM

An Interview with Software Artist, Eric Hammond

Eric Hammond has been programming computer games since 1980. His other professional credits include *Marauder*, *Battle Cruiser*, and *Maze Craze Construction Set*. A music major at Principia, a private school in Illinois, Eric plans to attend UCLA or UCSD in the future to become a member of the college basketball team.

HCM: First of all, how did you get the inspiration for *One-on-One*?

ERIC: Electronic Arts called me up about a year-and-a-half ago. They wanted to do a football game—but I play a lot of basketball, and love it, so I suggested we do a basketball game instead. It started when I thought about a one-on-one situation. I wrote down a script of all the features and how to implement the realism in the game. We discussed it for a long time before the start of the project and then went through a solid stage of design.

HCM: How did you work once you started programming?

ERIC: I started working on just the player mechanics. I spent about three weeks working an animation editor, where you can put the head, arms, legs and torso together to make a complete figure. Then I went through to the very end touching up little dots here and there and seeing that the animation was as smooth as possible.

HCM: So you had the game just about designed before Dr. J and Larry Bird even got involved?

ERIC: That's pretty true. They added things like the fatigue line and the spin. How they spin around was added about two months before the game was finished.

HCM: Did you use any film studies or photographs to capture the look and movement of these two players?

ERIC: Julius came out in July of 1983 to a clinic at the local YMCA, and I have about 500 stills of him shooting around, dunking, and so on. I used those as ideas for where the arms should be when he dunks, and how he has his body set in the air. That helped a lot. Also, I have the tape of the 1982 All-Star game with Bird and J in it. I single-stepped with the Beta machine to figure out how these guys move.

HCM: So, Dr. J actually had more direct input than Larry Bird?

ERIC: Yeah, we talked to him twice—and Bird once. We went back in August that same year to Springfield, Mass. They were both back for the Spaulding Endorsers meeting, and we caught them both at the same time, so we took them aside one day and talked about *One-on-One* and showed it to them. We sat Bird down in a motor home with the game, and he had some ideas about how to incorporate fatigue. That's where the idea came from.

HCM: You can tell that the touch they contributed to the game makes it a lot more realistic.

ERIC: Yes. And the biggest thing is that it inspired me to do a really good game. I thought, if they're on it, I want it to

be the best possible.

HCM: Do you think you've started a trend towards more athletes becoming involved in sports simulation games?

ERIC: That would be great, because their involvement just makes the programmer work that much harder, and it was kind of a sweet thing. I'm a very avid basketball player and also a programmer, so I'm sure that helped quite a bit—especially in designing the computer player, because I thought about what would I do, or what would Julius do, you know—how would he go around thinking and go in for the basket, and where would he jump—everything you can think of.

HCM: What machine did you originally design this game on?

ERIC: It was Apple based. And then I just put it over to Commodore and Atan.

HCM: When you went to the Commodore, did you incorporate sprites at all, or did you just stay in bit map?

ERIC: I stayed strictly in bit map. It worked out really well. When I worked with Commodore, I threw out everything as far as operating systems go. I used most of the RAM up there and left it almost like a bare Apple. Commodore was the biggest conversion, because you need a pretty bizarre routine to do all the masking and photograph background graphics.

HCM: Do you usually work with Apple first and then translate to the Commodore?

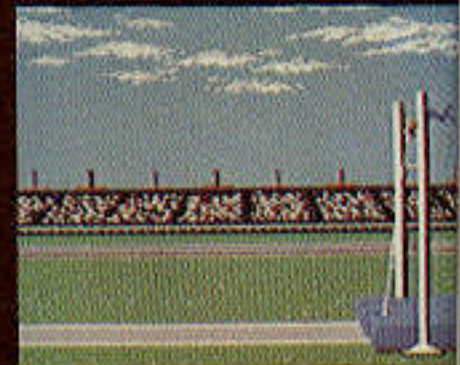
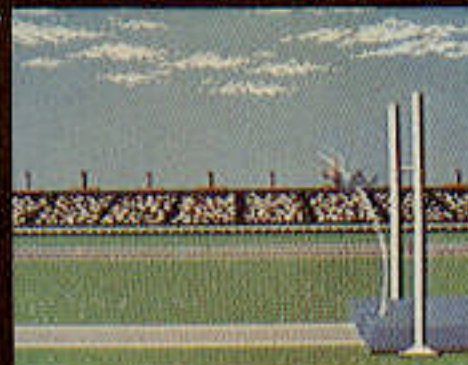
ERIC: If I come on a problem, I might go to the Commodore and see if I can use sprites to help it. But usually, if you do it on the Apple, the other machines fall in line.

HCM: It's like the Apple is the lowest common denominator, is that it?

ERIC: That's exactly right. You can do a lot on the Apple if you just use some tricks here and there.

HCM: How about the future? What are your plans, and what do you see happening to sports games in general?

ERIC: I think you're going to see some really incredible games come out because of the 68000 processor now on the Macintosh. It's pretty exciting, because with that kind of power you can use some really nice intelligence—graphics that haven't been done yet and couldn't be done on a 6502 machine. I'm doing the Mac conversion of *One-on-One* this summer, and I'll be able to put some more things in, like double and triple animation—so it'll be super smooth. In fact, it might turn into a three-on-three.





| | |
|----------------------|--|
| Name: | Country Club |
| Program type: | Golf Simulation |
| Machine: | TI-99/4A |
| Distributor: | User-Happy Simulations 133 North Prairie Whitestar, WI 53190 |
| Price: | \$19.95 disk or cassette |
| System Requirements: | |
| Extender BASIC: | Memory Expansion |
| Performance: | Pool Fair Good Excellent |
| Engagement: | ===== |
| Documentation: | ===== |

increasing the accuracy of a shot. Another accuracy aid is a transparent card (supplied with the package) with direction arrows indicating every fifth position which, when held in front of the screen, serves as a visual guide in choosing a direction for your shot.

Country Club displays fairways as aerial views. Detailed enough to include trees, the fairways are designed to introduce as much variation in terrain and hazards as a real course.

Fore!

The order that players tee off and take their subsequent shots follows traditional golf etiquette. When the last player reaches the putting green, the fairway screen is replaced with an aerial view of the putting green, and the screen displays conditions and information appropriate to the putting green (direction of break, steepness, etc.). Running score summations are listed for each hole after the final shot of the hole is played, and individual handicaps are also displayed.

Country Club

A review
by Tom Green
HCM Staff

When the adjustable club hooked into the golf market, golf purists were aghast; a serious golfer would never consider using such a contraption. Nevertheless, the adjustable club gained popularity.

But a game that simulates golf? To a seasoned player, this must be the last affront—or is it really an affront at all?

User-Happy Simulations thinks not. They offer a golf game for golfers and video-game buffs alike. *Country Club* delivers hours of realistic and challenging golf play, conveniently bringing the fairway into your living space. Now, teeing off is as easy as booting up.

Course Guide

Country Club is noticeably different from other simulation gameware. No pre-game jingle. No arcade tune to accompany tee-off. The lack of music, however, is not meant to defy arcade protocol—there is a practical reason why this was done. The program uses memory space to achieve realism in the kind of variables associated with the game of golf, not for catchy tunes. Let's leave the caddy shack behind now and scout the fairway for hole #1.

You begin by choosing from three options: Advanced, Beginner or Editor. There are two different 18-hole courses: Option 1 will allow play on the more difficult course, and Option 2 is for play on the easier course. Option 3 is a subroutine that allows "hacker" manipulation of fairway graphics.

Handicap scoring adjusts for play levels: Pro, Good, Fair and Clod. By choosing the Clod level, for example, a player introduces more error into the control and power of strokes, but the game becomes more challenging.

You can deduct your total Clod level handicap at the end of a game and

possibly underscore someone at the Pro level.

"At the Clod level, you can deduct your total handicap at the end of the game and possibly underscore someone playing at the Pro level."

The club being used, the force of the shot, the terrain where the shot takes place, and the wind all affect the accuracy of a shot. Players have 15 clubs from which to choose. The wedge and short irons offer the best control, and naturally as a club gets "longer," control decreases. Thus, the club with the least control is the driver, because it is the longest club.

The force of a shot is set on a scale from one to ten, with one being the weakest stroke. However, only 9 and 10 seem to have any effect on control, simulating the unpredictable results of an overpowered shot.

Terrain also affects control, as in real golf: On the putting green and tee, excellent control is possible. Shots from the fairway usually have good control. But when shooting from the rough, control is fair to poor, and from a sand trap, the outcome is unpredictable. The wind is present only in the advanced game, affecting direction and distance. Shots must be adjusted to allow for the bearing and strength of the wind. "Yardage Cards" are supplied with the package to approximate the distance of each club stroke for a certain power and terrain orientation.

Ideally, the direction bearings in a game like this would have 360 settings, to simulate the degrees of a circle. *Country Club* lets you determine direction on scale from 0 to 60 (think of a stopwatch and its hand as a pointer in one of 60 directions). The precision of this scale is more than adequate for

Par For The Course

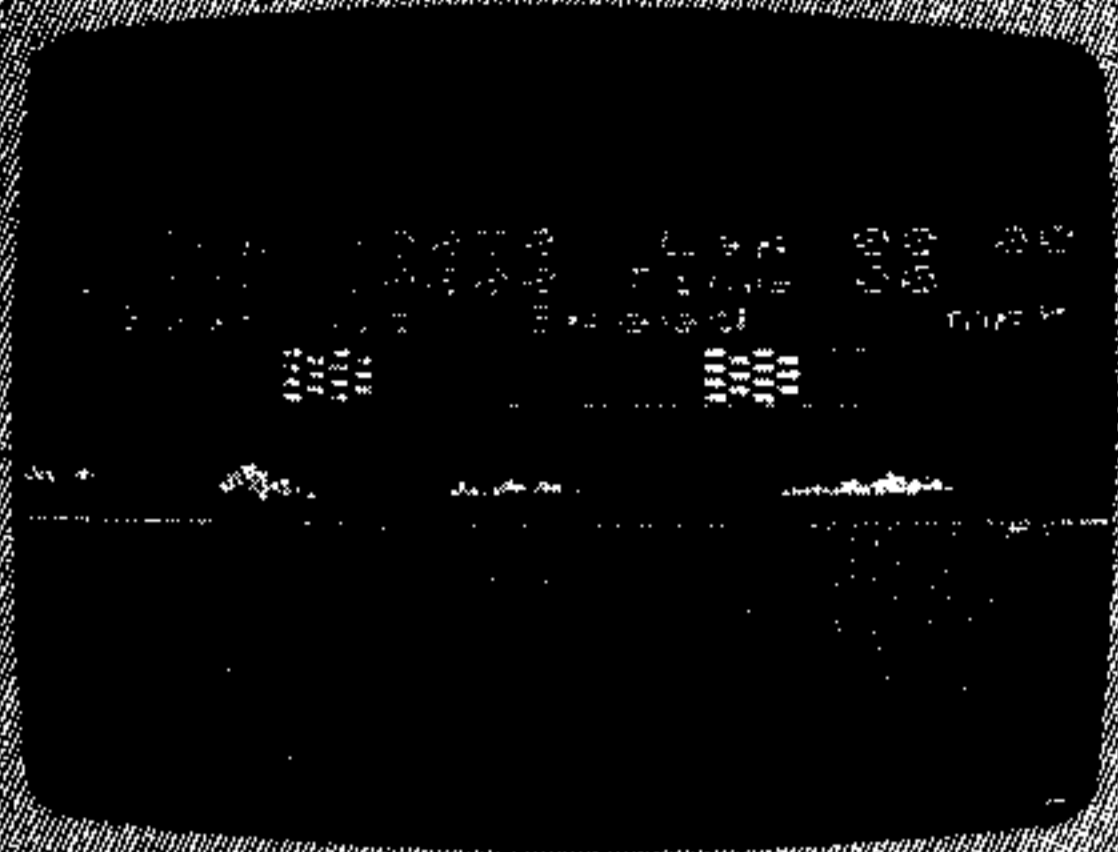
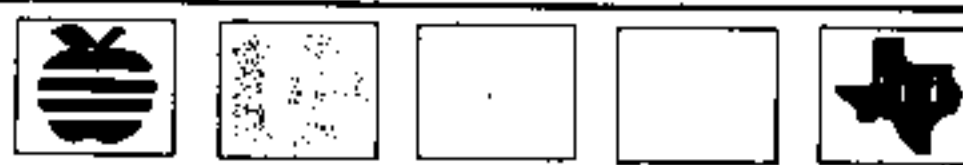
Country Club's overall performance is true to the characteristics of real golf. Real golf can be a slow game—even boring for some. But *Country Club* moves along at a satisfying pace, with no undo pauses or delays. Although adjusting all parameters before making a shot can take some time, there is nothing illogical or frustrating in the process. In fact, to the real golf enthusiast, breaking down a swing into all of its components can be very interesting and informative.

Documentation

The user's guide is easy to follow and understand and could well serve as an introductory tutorial on the many elements and strategies involved in a basic game of golf. In addition, game features on scoring, club choice and use, and insights to game play are clearly outlined.

The Editor subroutine provided for changing the formats of the different fairways lacked documentation at the time this review was written. User-Happy Simulations will be issuing a "hackers" guide supplement to the *Country Club* manual in the near future.

All in all, *Country Club* is a real golfing adventure for you TI owners; its value goes beyond the mere fun of a video game. If you're still working on that swing, this program just may teach you something.



| | |
|----------------------|---|
| Name: | Pole Position |
| Program Type: | Arcade Game |
| Machines: | TI-99/4A, VIC-20, C-64, PCjr, Apple II Family |
| Distributor: | Atari Inc. Atari Inc. P.O. Box 43687 Sunnyvale, CA 94088 |
| Price: | \$24.95 (incl. tax) \$44.95 (incl. tax) |
| System Requirements: | Joystick optional For TI-99/4A, Excellent |
| Performance: | Excellent |
| Engagement: | Excellent |
| Documentation: | Excellent |

Pole Position

A review
by Dana M. Campbell
HCM Staff

Ladies and Gentlemen, start your engines, cries the announcer, and you rev your engine, keeping a firm, gloved grip on the gear shift knob. It's the last week in May, and your moment has come to attempt to qualify for a position in the Indy 500. You wait for what seems a million seconds before the signal light turns green, and then in an explosion of dust and noise . . . you're gone.

This is the scenario that was successfully simulated by Atari in the arcade game *Pole Position*, now available for the home. The game's basic concept is easy to understand but challenging to perform: guide a turbo-charged Formula 1 racer around a track without hitting any obstacles or other cars, and do it fast enough to qualify for the coveted pole position (first row, inside lane) in the main event. Of course, various factors have been thrown in to make the game more stimulating and realistic, and it is these touches that make *Pole Position* the best of the driving-simulation games.

For instance, if you round a corner too fast, your car will skid, slowing you down while the clock ticks away. Also, unlike similar games, the other cars on the road do not remain stationary—they will move around on the track as you approach, and ruthlessly attempt to cut you off as you pass them. Nerves of steel and a steady hand are all that will get you through some of those tight squeezes.

Staying On Track

Although a keyboard can be used to play the game, the ease of a joystick is akin to power steering. Pressing a (fire or keyboard) button will toggle you from low to high gear and back again. Three levels of difficulty test

"Unlike similar games, the other cars on the road do not remain stationary—they will move around on the track as you approach, and ruthlessly attempt to cut you off as you pass them."

your skill, with the only apparent difference between them being that the number of curves and cars on the track increases with each succeeding level. Accumulate points by completing a lap, by passing a car, and for each second of time left on the clock after you've crossed the finish line. If you finish the qualifying race in 73 seconds or less, you receive a position (from one through eight) in the main race. Finishing the big race within specified lap times will garner you some Extended Play.

The differences between the TI-99/4A and the Commodore versions of *Pole Position* are quite noticeable. On the TI, players can choose from one to eight laps per race, but are limited to a maximum speed of 195 miles per hour (mph). Players on the VIC-20 or C-64 can accelerate up to 244 mph, but must contend with a set number of laps every race.

The TI game is also not as well crafted as the Commodore games. In the TI version there is no signal light to begin the race—players must rely on the sound alone. The car responds slowly on turns, and loses speed a bit when shifting from low to high gear—even when shifting at the recommended 100 mph. The background scenery changes in jerky movements which supposedly depict motion. This "motion", however, reminded me more of oldtime movies than a high-

speed race. In addition, cars passed on the track aren't actually "passed"—they simply disappear just before your car is opposite them. I actually ran over a few such cars in my path and did not crash. Too bad I can't do that on the highway . . .

The Commodore versions feature brighter, more detailed graphics, and the other racers on the track provide more of a challenge—avoiding them requires some pretty intricate weaving among the cars. The traditional signal light is there to start the race, as are the appropriate engine sounds. Shifting and motion occur smoothly. The C-64 game even offers an option that allows players to pause during their race. You can't pass on the grass in these versions, but you're better off not trying to pass on the outside anyway—you'll be quickly cut off and will likely crash.

Too Much of a Good Thing

The game's instruction manual is simple but complete. In fact, the Strategy Tips almost tell you too much, for if you stay on the center stripe as suggested, you can bypass everything, taking away most of the game's

challenge and fun. (You can't get as good a lap time though by staying on the middle line. It's faster to hug the curves.)

The home game's sound effects exactly mimic the original arcade game, from the starting tune to the crashes.

As mentioned above, *Pole Position* is a game that is easy to learn, which is also its main drawback. Once you've taken a few spins around the track, the track and its scenery become predictable and boring. Teenagers on up may tire of the game after a few plays, while younger kids may spend more time with this simulation of something they are not yet allowed to try in real life.

Pole Position would be more interesting if a few slick spots were placed on the track so that they couldn't be seen from a distance. Adding another gear or two would also keep things hopping, and why not require some quick pit stops after every few laps? Finally, a few scenery changes would help keep the race lively. The arcade version of *Pole Position II* features this last suggestion, so perhaps we will see it added to a new home version.

In the meantime, practice hugging those curves, and try not to hit any billboards this time . . .

Versions for IBM PCjr and Apple II Family had not yet been released at press time. HCM



| | |
|--|--|
| <p>Game: Buck Rogers Program Type: Action game Platform: TI-99/4A Developer: Atari Year: 1983 Price: \$29.95 System Requirements: Atari 9900, 9940, 9900X, 9940X, 9900XL, 9940XL Minimum Requirements: Atari 9900, 9940, 9900X, 9940X Recommended: Atari 9900XL, 9940XL Documentation: Manual Playability: Excellent Value: Excellent</p> | <p>Game: Star Trek Program Type: Action game Platform: TI-99/4A Developer: Atari Year: 1983 Price: \$29.95 System Requirements: Atari 9900, 9940, 9900X, 9940X, 9900XL, 9940XL Minimum Requirements: Atari 9900, 9940, 9900X, 9940X Recommended: Atari 9900XL, 9940XL Documentation: Manual Playability: Excellent Value: Excellent</p> |
|--|--|

Buck 'n' Kirk

A review of
BUCK ROGERS—PLANET OF ZOOM and STAR TREK
 by Steve Nelson
 HCM Staff

Over the years there have been several attempts to launch a new sci-fi series on prime-time television, but other than *Star Trek*, only *Buck Rogers in the 25th Century* and *Battlestar Galactica* were able to make any kind of impact. Now that these series are gone from the airwaves (except for reruns), sci-fi fans have been forced to turn to video arcades to see any interplanetary adventure outside of movies. But wait! Thanks to some like-minded video game programmers, the *Enterprise* and *Buck Rogers* once again are on the TV screen—this time with you at the helm.

Hoppers And Klingons?

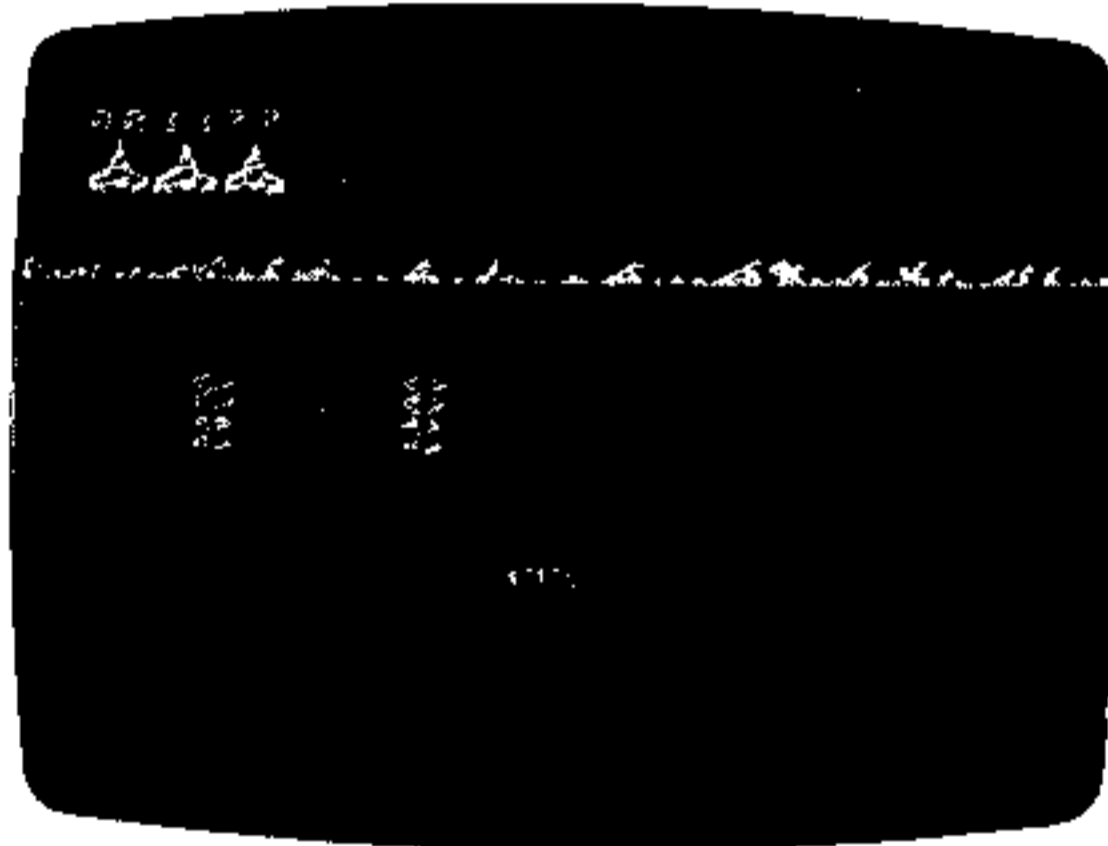
Like most space-theme games, these two provide lots of alien spaceships and other neat things to blast, but they do it in different ways. On the *Enterprise*, the familiar front viewscreen is your window to deep space. Alien saucers, Klingon battle cruisers, and Nomad (who scatters bombs and mines in all directions) all attack you furiously as soon as you begin play. In *Buck Rogers*, you must pilot a spaceship over the three-dimensional surface of the planet Zoom, negotiate a series of tall electron posts (one touch is instant destruction), shoot your way through waves of creatures called hoppers—who, by the way, are very difficult to hit—then leave the planet and attempt to fight your way through flying saucers defending their mother ship. Whew!

Once you destroy the mother ship, you advance to the next level of play and begin the game again. As you move from one level to the next, the game's speed increases.

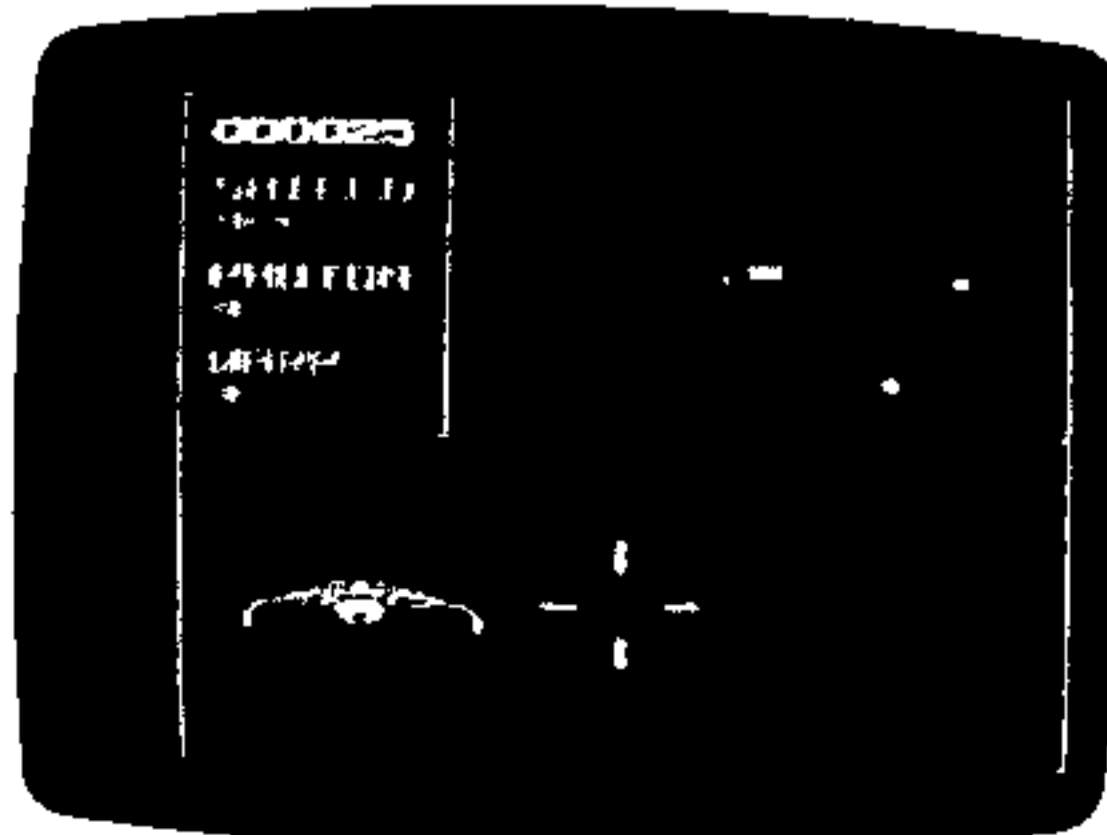
Buck Rogers is somewhat disappointing in terms of its scenario, as the game seems to rely more on its 3-D effect to keep you interested, rather than create anything more than a picture on the screen. The spaceship itself, complete with shadow on the planet

Zoom's surface, is an excellent model, but the hoppers and the flying saucers are not nearly as well depicted. Once I left the planet to attack the mother ship, I expected a spectacular battle with a huge spaceship; instead, the mother ship looked tacky, and it was much too easy to destroy.

The ship's response to your joystick is spectacular. You can bank to the right or left, and the planet's surface curves away beneath you. The closer to the ground you fly, the slower your ship moves, giving you more control as



"The ship's response to your joystick is spectacular. You can bank to the right or left, and the planet's surface curves away beneath you."



you maneuver between electron posts, but causing you to burn more fuel. When you fly higher, the opposite happens—your ship burns less fuel, but you lose some control because it is more difficult to negotiate the electron posts at faster speeds. And, at higher speeds, the hoppers and flying saucers attack much faster. You will have to have incredibly fast reflexes, or luck, or both, in order to play this game at full speed.

Watch For Energy Bolts

I also tested *Buck Rogers* on the Commodore 64. The graphics on the C-64 version aren't as crisp as those on the TI version, but the game is more difficult, with a few more obstacles. (For instance, when you advance to the second level of difficulty, the electron posts begin emitting bolts of energy that can destroy your ship unless you fly between them.)

Star Trek is a much less graphic game, but more challenging. You must keep your mind on more than just blasting the aliens. The *Enterprise* viewscreen is split into three sections: a view of the *Enterprise* out in space with the Klingons, flying saucers and Nomad around you; a close-up view of what is in front of you, along with your gun sight; and indicators that keep track of how many shields, photon torpedos, and warp drive units you have left. You must monitor all three, keeping in mind that your main worry is losing your shields.

Each time the *Enterprise* suffers a hit, it loses a shield. To replenish, you must rendezvous with a green starbase which will give you one shield, one photon torpedo, and one warp drive. Your starship comes armed with phazers and photon torpedos, energy shields, and can maneuver at either warp speed or on impulse power. Ten levels of play challenge you, each level continuing at a faster pace.

The only drawback I ran into while playing *Star Trek* was that after about five minutes of playing time my hand went numb from holding the joystick.

It's Them Or You

The documentation for both games is adequate, and you really don't need to know much to play either game: just blast them before they blast you.

Both of these games for the TI-99/4A have a speech option. If you have a speech synthesizer hooked up to your computer, you can hear a limited amount of speech.

I really liked the response of the starship in *Buck Rogers*, but the game itself tended to get boring after a few minutes of play, and the 3-D effect is very hard on the eyes. *Star Trek* is a more challenging game, but less exciting visually. While neither one of these games are as good as a *Star Trek* rerun, they are both fun to play, and can hold their own against the hordes of other star-struck video games now on the market.

SST BASIC COMPILER SYSTEM

A review
by Tom Green
HCM Staff

SST Software's new *Expanded BASIC Compiler System* enhances TI BASIC by giving the BASIC programmer an expanded instruction set, while greatly increasing program execution speed. This, however, doesn't come without a price: With the *SST System*, programs have to be re-written in a highly structured format, and carefully edited and compiled—resulting in greater program development time. As is the case with nearly all microcomputers, the TI-99/4A's resident BASIC is an "interpreted" language. When the Central Processing Unit (CPU) RUNs a BASIC program, it's like trying to read a book in a foreign language—looking up every word in a dictionary to find out what it means—instead of reading a translation. A BASIC compiler translates the program into the CPU's "native tongue" (machine language) *before* the program RUNs—so that it can RUN many times faster.

SST Expanded Compiler

Three principal steps are involved when using the *SST Compiler System*.

The flow chart in Figure 1 demonstrates the entire process: **(1)** Create and edit an SST BASIC program by modifying an existing TI BASIC program, or writing a new one to conform to SST BASIC format. RUN-test it using one of the two editor programs and prepare it for compiling. **(2)** Compile and SAVE the program to disk using the compiler program. **(3)** LOAD and RUN the program using one of the two loader programs.

Last year SST Software broke the Sound barrier with a cassette-based compiler system for the 99/4A. Now the expanded system goes to Mach 2.

Preparing To Compile

The first step is to prepare your program in SST format. Figure 2 (taken directly from the SST manual) shows the outcome of such a conversion. The "at" symbol (@) after each of the variable names designates the variable as an integer. For the inexperienced programmer, this next requirement will be the most difficult part of using the *SST System*: Any variable or constant you wish to use in your program must be defined at the beginning as a one- or two-character variable name. No actual numbers are allowed in the body of the program, so if you want to add two numbers together, they must be defined. These requirements defeat the easy (although unstructured) features that make BASIC such an attractive language. This drawback, however, is somewhat offset by the fact that novices will be introduced to structured programming in the familiar BASIC environment.

Still, we do feel that defining constants (something not required in even a highly-structured language like Pascal) is a little hard to justify. Once you've written the SST BASIC version of your program, it is combined with one of the two editor programs. The *BASIC Editor* requires either the Mini-Memory or Editor/Assembler Module, and the

Editor/ex requires the TI Extended BASIC module.

The major advantage of the *Editor/ex* program is that it allows use of the Merge option to combine a program with the editor. To use the *BASIC Editor* your program must be keyed in with the *Editor* in memory. Next, RUN-test your programs from

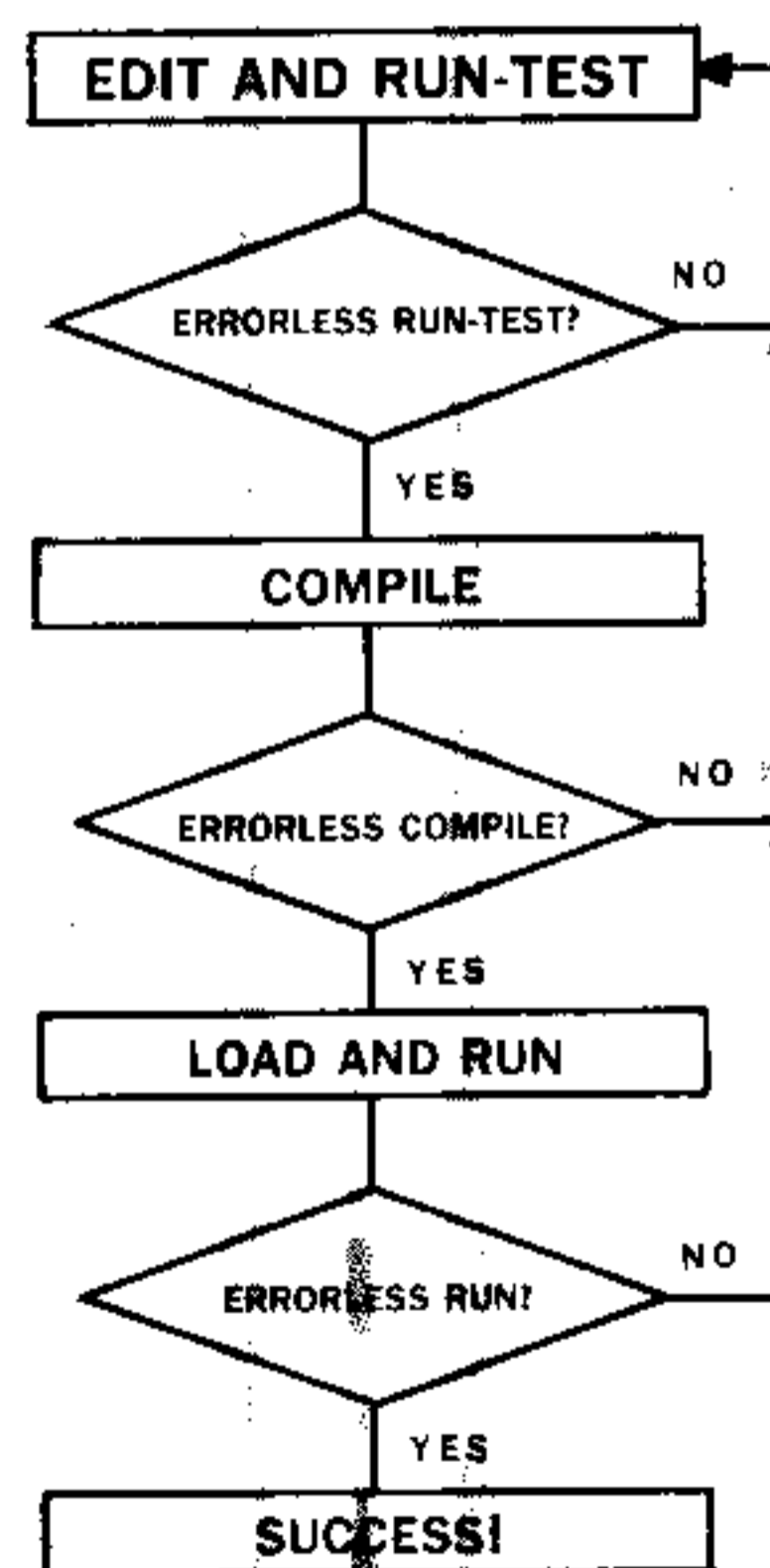


Figure 1



either editor. Every instruction is checked for correct syntax, and errors are noted so that modifications are relatively easy to locate. You modify errors found by the editor by using regular TI BASIC or Extended BASIC commands in the conventional manner. The *Expanded SST Compiler System* also includes many new commands. During the RUN-test these SST BASIC commands are printed to the screen as a means of tracking your program.

Compiling, LOADING and RUNing

Once you have completed the Editor phase of debugging, you RUN the editor to create your source file, and you're ready to begin compiling. Program debug-

| | |
|----------------------|--|
| Name: | SST Expanded BASIC Compiler System |
| Program Type: | Utility |
| Machines: | TI-99/4 or TI-99/4A |
| Distributor: | SST Software, Inc. P.O. Box 26 Cedarburg, WI 53012 |
| Price: | \$95, Disk; \$85 User Group fee; \$50 with proof of purchase of SST BASIC Compiler System. |
| System Requirements: | Memory Expansion, Disk Drive, and Editor/Assembler or Mini-Memory Command Module. |
| | Poor Fair Good Excellent |
| Performance: | ██████████ |
| Ease of Use: | ██████████ |
| Documentation: | ██████████ |

ging is a trial-and-error process between the *Editor* and *Compiler* programs. The *Compiler* will catch errors that get by the Editor. Yet, it is not 100% effective—a program may successfully pass through the *Compiler* and not RUN properly, leaving the developer with "run-time" errors. This "hit-and-miss" method

of editing can be frustrating, and is always time-consuming. For instance, the process of editing, compiling, and running the program in Figure 1 took approximately 15 minutes to complete. You can see that a longer, more complicated program would require much more time and patience to develop. The increased performance of the program, however, makes up for the frustration that arises from implementing these procedures. (Try writing machine code from scratch sometime—you'll appreciate the power of the *Compiler*.)

When you finish compiling, the *Loader* program then accesses code generated and saved to disk by the *Compiler*, and places it into memory for execution. You may link and RUN several compiled programs using the *Loader*. There is also a Fast Load option that prepares programs so they don't require the *Loader* each time they RUN. Once prepared, the programs RUN directly using the commands CALL INIT, CALL LOAD, and CALL LINK, all of which are available from TI BASIC with either the Mini-Memory or Editor/Assembler modules. This creates an executable file that can be RUN on any TI computer with Memory Expansion and the particular command module used during the Fast Load procedure.

Pixels Accessable

One of the more powerful commands (unique to SST BASIC) is the hi-resolution screen graphics mode, specified as PLOTMODE (Bit-Map Mode). This function allows access to each of more than 49,000 pixels (dots) on the monitor screen. You have the ability to place or remove a character anywhere on the screen, designate foreground and background color, or apply the same parameters to individual pixels. This is the first time such access has been available to programmers on the TI-99/4A, outside of assembly language or TI Forth.

The *SST BASIC Compiler* can only use the TI Extended BASIC module when running the Editor/ex program. But no matter which editor you use, the SPRITEMODE commands give you access to sprites. Generally, these commands are similar to those available in TI Extended BASIC, with only minor variations.

"A BASIC compiler translates the program into the CPU's "native tongue" (machine language) before the program RUNs—so that it can RUN many times faster."

These new enhancements, however, do not come without certain limitations. Because of differences in the TI-99/4A's video-processor memory modes, SPRITE and PLOT modes are not compatible in the same program. Either command can be CALLED only once from a program, and both have further restrictions on accessing certain codes.

Expanded SST Is Expandable

Another plus of the *SST System* is access to user-defined routines with the CALL USERA . . . E command. Up to 6 assembly language or compiled BASIC routines may be included in the *Loader* program as a sort of user-defined "library" of functions. Thus, as you become more adept at programming your TI-99/4A on an assembly language level, you can customize the *SST System* to include your routines.

Documentation

The first word that comes to mind when describing the user's manual of the *SST Expanded BASIC Compiler System* is "beefy." Guidelines for program development are thoroughly outlined in step-by-step fashion, and include remedies for common pitfalls in the procedures. This product however, is not for the casual programmer. But, those who are willing to invest the time to learn the *SST System* will be rewarded with programs that run many times faster than ordinary BASIC programs.

HCM

The next program is similar to one which appeared in the March, 1980 BYTE Magazine. It is a program designed to generate prime numbers, and is often used as a benchmark. The program was originally run in Basic on the TRS-80 computer. It took 7 hours, 12 minutes to check the first 10,000 integers for prime numbers. The program written here checks only the first 1,000 integers.

```

100 LET L@=6
110 LET E@=1
120 LET M@=1000
130 LET Z@=5
140 LET A@=1
150 LET N@=10
160 LET D@=1
170 LET B@=2
180 LET C@=2
190 FOR A@=L@ TO M@
200 A@=A@+E@
210 D@=A@/C@
220 FOR Z@=B@ TO D@
230 Z@=Z@+E@
235 REM FOR T.I. BASIC LINE 240 SHOULD BE
236 REM           N@=INT(A@/Z@)
240 N@=A@/Z@
250 N@=N@*Z@
260 N@=A@-N@
270 IF N@<=0 THEN 300
280 NEXT Z@
290 PRINT A@
300 NEXT A@
310 STOP

TIME, BASIC:           1535 seconds
TIME, SST COMPILER 18 seconds

```

If line 120 is changed from M@=1000 to M@=10000, the program will check the first 10,000 integers. The *SST EXPANDED COMPILER* completes the program in 11 minutes, 20 seconds. In T.I. BASIC, it took 4 hours and 15 minutes to check the first 5500 integers. The *SST EXPANDED COMPILER* took 4 minutes to check the first 5500 integers.

Figure 2

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.



Need help in forming a Commodore user group? If so, call the **New Mexico Commodore Users Group** in Albuquerque. They will send copies of *On-Line* (their newsletter), their bylaws, and any other pertinent information upon request. They will also answer questions regarding becoming an affiliate of NMCUG. This is just one of many aspects of this group which offers monthly meetings, hardware and software reviews, classes, a public-domain software library, and a newsletter. Individual membership dues are \$12 per year, \$15 for family and out-of-state individuals. If you would like to volunteer some time helping the group, you can earn credit toward the membership fee. For more information, write: NMCUG, P.O. Box 37127, Albuquerque, NM. 87176.

The **San Luis Obispo Commodore Computer Club** was formed a year ago with just a handful of interested people, and today they count a membership of more than 130 members. The group has a large club library containing programs for the VIC-20 and Commodore 64, produces a newsletter, and operates a bulletin board service. According to Alan Heminger, outgoing president of the group, the club meetings are well-attended and informative. The club even has its own computer and disk drives, which allows them to provide many of their own services. If you live in the San Luis Obispo area and are interested in finding out more about this club, contact: Gary Bissell, 1766 Ninth St., Los Osos, CA. 93402, (805) 544-2924.

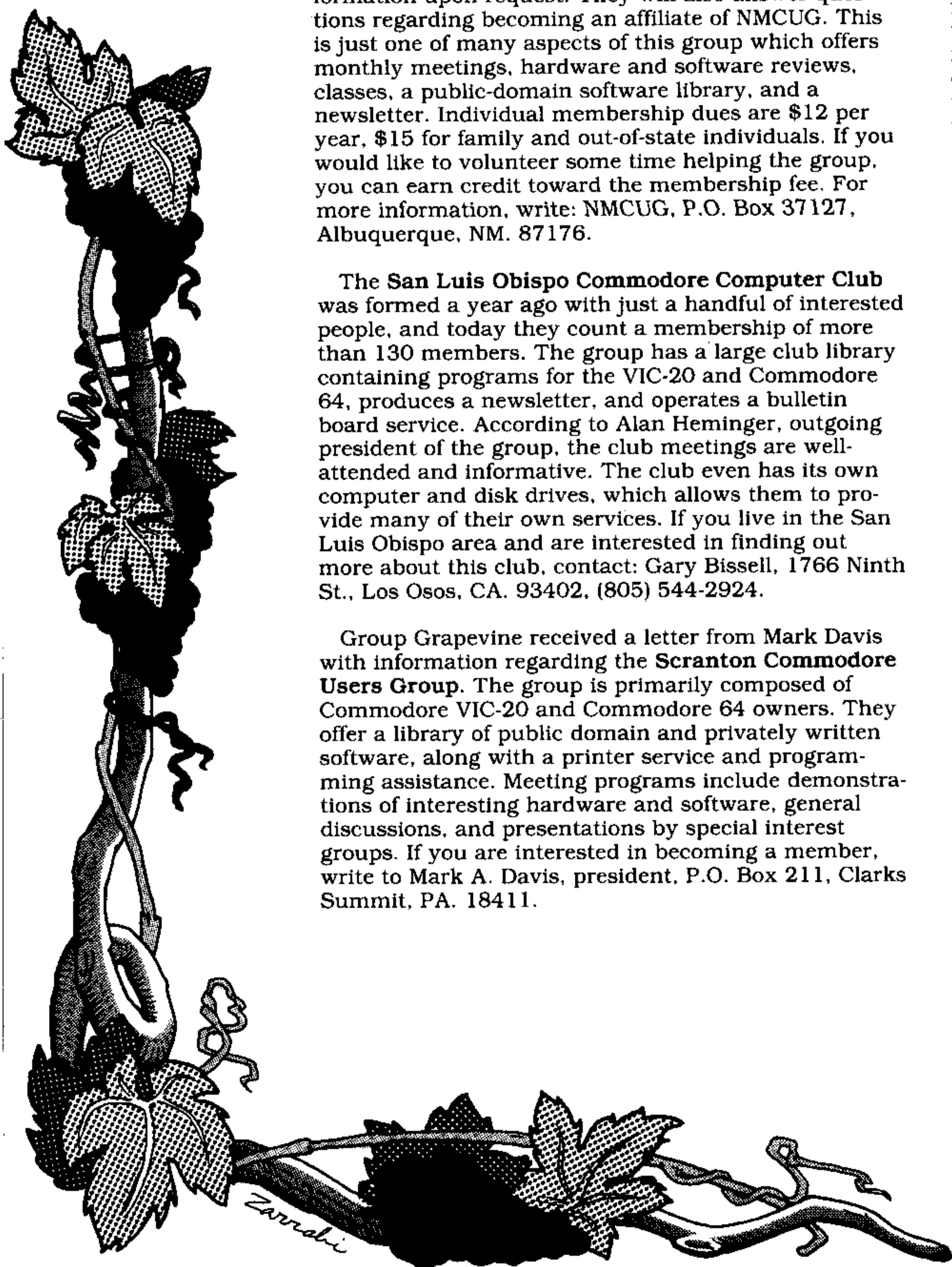
Group Grapevine received a letter from Mark Davis with information regarding the **Scranton Commodore Users Group**. The group is primarily composed of Commodore VIC-20 and Commodore 64 owners. They offer a library of public domain and privately written software, along with a printer service and programming assistance. Meeting programs include demonstrations of interesting hardware and software, general discussions, and presentations by special interest groups. If you are interested in becoming a member, write to Mark A. Davis, president, P.O. Box 211, Clarks Summit, PA. 18411.



David Frye, a 16-year-old high school sophomore from Alden, New York, has initiated a TI-99/4A users group at his high school. The idea seems to have caught on well—new members from all over the Clarence School District are joining up, and so far no one is over 17. The group's library consists of 78 BASIC and 31 Extended BASIC programs. They are now looking for other users groups run by kids with which to correspond and exchange ideas. If you are interested in becoming a member, contact David Frye, 1132 Boncliff Drive, Alden, NY. 14004.

Congratulations! **TISHUG** (Texas Instruments Sydney Home computer User's Group) in Australia celebrated their third birthday in May. **TISHUG** founder, Shane Andersen, has watched the club grow to an unbelievable 700 members since that first meeting in May 1981. The group's dream of establishing a bulletin board service has finally come true. **TISHUG**'s bulletin board is for downloading software, up-to-date news and views, electronic mail, and programming hints. This electronic bulletin board will soon become the very first bulletin board in the Southern Hemisphere to have clear, spoken text as displayed on the screen. The group is also making plans to provide low-cost RS232 interfaces and modems so that as many club members as possible can take advantage of the bulletin board. In addition, each month **TISHUG** conducts a software competition open to everyone. Prizes are awarded for the best in the following divisions: Best Award of the Month; Junior Award of the Month; and Rookies Award of the Month. There must be many TI-99/4A users "down under" who would like to join up and share their talents. Upon glancing through the Sydney News Digest (**TISHUG**'s newsletter) one notes that there are several active regional groups affiliated with **TISHUG** in Sydney—Blaxland, Newcastle, Illawarra, Mosman, Nepean, Bulkem Hills, and Marrickville/Ashfield. If you are interested in becoming a member, contact: John Robinson, P.O. Box 149, Pennant Hills, NSW, Australia 2120.

Group Grapevine received its first issue of **Channel 99 User Group**'s newsletter from Tom Arnold, coordinator. This group of 180 members—several of whom live as far away as 200 miles—hails from Hamilton, Ontario (Canada). They are interested in exchanging newsletters and user-written programs with other groups. Channel 99 recently held a *Munchman* competition at their monthly meeting, and they are currently planning for a lending library consisting computer-related books. Also in the planning stage is a programming tutorial to be held at the meetings and augmented with lessons and problems in their newsletter. If you think you live too far away from a users group to become a member, this is the group to contact. Write: Tom Arnold, 77 Lavina Crescent, Hamilton, Ontario L9C 5S8.



Group Grapevine just received some unhappy news from Ed York of the **Cin-Day Users' Group** in Cincinnati. In Vol. 4, Issue 2 of Home Computer Magazine, Ed announced that the group was preparing to sponsor the first annual Midwest Computer Fest in the fall. According to Ed, the group's new officers feel that it is no longer feasible for the **Cin-Day Users Group** to sponsor this project. Therefore, the first Midwest Computer Fest has regrettably been cancelled. If you have any questions, contact Ed York, P.O. Box 519, West Chester, OH. 45069, (513) 777-0010.

A small group of TI-99/4A owners in Dimona, Israel is very interested in joining with another user group. Kitlaru Beny writes Group Grapevine that the group is extremely hungry for all kinds of information regarding software, peripherals, etc. If you would like to contact this faraway group, write: Kitlaru Beny, P.O. Box 565, Dimona 86104, Israel.



Wow! Here's music to Group Grapevine's ears! An IBM PCjr computer club is forming out there in the Washington, DC area. The **Capital Area IBM PCjr Computer Club** is the place to learn more about hardware and software for Junior, including: keyboards, memory expansion, magazines, games, and business software, as well as IBM compatibility, DOS 2.10, BASIC programming, home/office applications, telecommunications, word processing, spreadsheets, and much more. The club is open to all persons who have PCjrs, or who are merely interested in computers. The club is geared toward less technically-experienced people who want to know more about home computing. For more information, contact: Lowell Denning, 12611 Beechfern Lane, Bowie, MD. 20715, (301) 262-8275 or (202) 566-4801.

The **Portland IBM PC Club** of Portland, Oregon has a very active group, with special interest groups in the following areas: spreadsheet, business applications, C language, word processing, data base management, novice, and hackers. The club's efforts to establish a bulletin board have finally come to fruition and it should be in place and operational by the time Group Grapevine goes to press. The Portland club has members who are willing to rent their personal computers for a fee set by each individual member. If you live in the Portland area and would like more information about this very active and interesting group, contact: Rich Rohde, P.O. Box 2068, Beaverton, OR. 97075, (503) 620-6862.

According to the **Quad-City Personal Computer User's Group's** newsletter, this group provides a forum for the exchange of information by PC users, a software library, outstanding public-domain and user-supported programs, educational programs, and group purchasing of hardware and software. If you live in the Bettendorf, Iowa area and are interested in becoming a member, contact: John Dannenfeldt, P.O. Box 464, Bettendorf, IA. 52622, (319) 752-0245.

Group Grapevine just received a news release from IBM with information regarding support to PC user groups. User groups can now receive support and information directly from IBM's Entry Systems Division. The support department offers a newsletter distributed on a diskette that includes selected technical articles from user group publications,

items of general interest to user groups, and information about recent IBM PC product announcements. The division also offers a bulletin board to assist user groups in communicating with other groups and to provide general product information. It also includes answers to questions frequently asked on the support department phone line. Officers of PC user groups can call the support department for general information about user group activities, IBM PC products, and forming a new user group. If your group would like to register with IBM, write to Gene Barlow, IBM PC User Group Support (2900), P.O. Box 3022, Boca Raton, FL. 33432.



A group focusing on practical applications for Macintosh has taken root in New York City. **New York MacUsers' Group** meets monthly and anyone can become a member—not just New Yorkers. The meetings include software exchanges, speakers, demonstrations, and many more topics of interest, as dictated by the membership's interests. If you're a new Mac owner and would like more information, contact: Cheryl Sandler, NYMUG, P.O. Box 6686, Yorkville Station, New York, NY. 10128, (212) 535-1943.

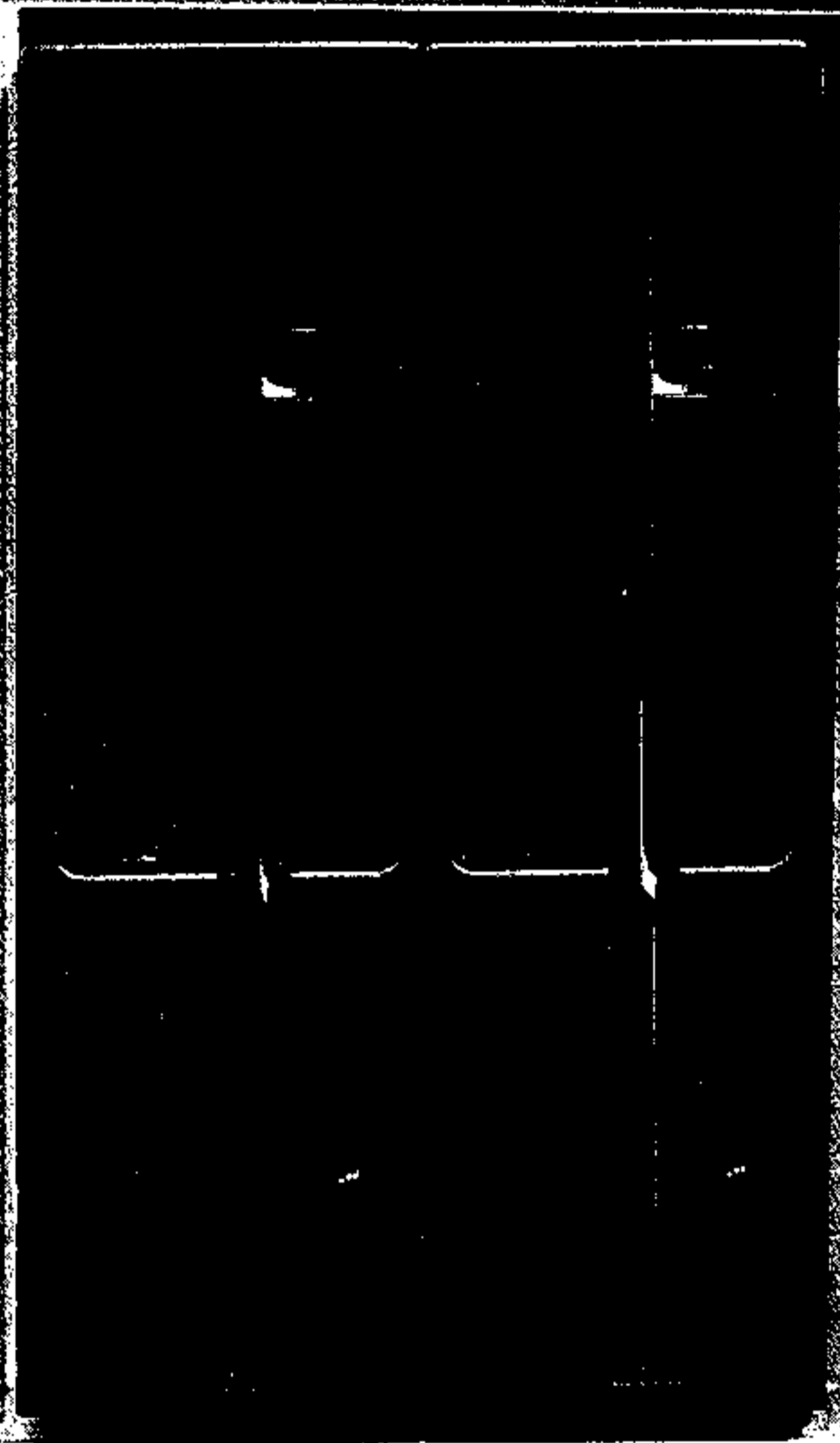
Group Grapevine just had a phone conversation with Dave Hoffman, founder of the **Wenatchee Valley Apple User Group**, established in September, 1981. The group consists of 67 members from many professional fields, including engineering, medicine, law, and education. The group library consists of approximately 1,000 titles, which can be sent out by mail to any interested club member—as well as other users groups. They have donated about one-half of the library titles to the regional library. This group also helped an Apple group in the Tri-Cities area get off the ground, and because there are more and more Macintosh users becoming interested in the club, they are considering forming a Mac group. If you are interested in this group or its possible offshoot and live in the Wenatchee area, contact: Dave Hoffman, 535 Highland Drive, Wenatchee, WA. 98801, (509) 662-7317.

Some news comes from a relatively new Apple group, the **Roseville Apple Users Group** in Roseville, CA. This group has been in existence for only four months and has 55 members. After talking with Otto Haiungs, president of the group, it sounds as if they have plans for several community projects, as well as in-house club projects. Activities that include the handicapped in the community is one idea they plan to pursue, and any help from people knowledgeable in this area would be greatly appreciated. The group meets at the Roseville Main Library the second and fourth Tuesday of each month (the meeting on the fourth Tuesday is a Special Interest Group meeting). Currently, the group has SIGs covering the Macintosh, business, and games. Membership dues are \$12 per year, and include a subscription to the Roseville Apple Core Bulletin, the group's monthly newsletter. If you are interested in becoming a member, or have information regarding computers and the handicapped, please contact: Otto Haiungs, P.O. Box 1377, Roseville, CA. 96551, (916) 783-0364.

Okay Apple users, it's tree-shaking time! We really would like to hear from all of you Apple groups via letter or newsletter (or even a phone call), so we can keep up with all the exciting things that are blossoming out there in Apple land. Some Macintosh seedlings to plant amongst the other Apple varieties in this orchard would be extremely welcome.

HCM

Imagine, 360K bytes of diskette storage in the TI Expansion Box—magic made possible by installing two, half-height, dual-sided disk drives in the TI box.



CompuAdd Corporation is marketing two Shugart SA455s as a dual-disk drive that slips nicely into your peripheral expansion box with no modifications. Although two disk drive units would overtax the power supply, they claim to have solved this problem by utilizing drives that require half the normal power. So while you are in fact running two drive motors, they require approximately the same amount of power as one regular drive motor—with no loss of performance. Because these items are sold separately by the distributor, we felt our readers would appreciate some added documentation explaining installation procedures.

Getting Ready

When your dual-disk drive arrives, the first thing you will need to do is take the dual-disk drive unit out of the box and make certain that all the necessary parts are there (see Photo 1). Next, you must turn off the power to your peripheral expansion box and wait a couple of minutes before removing the old disk drive unit. The only tool required is a Phillips screwdriver.

Removing The Old Disk Drive Unit

First, remove the screws holding the old disk drive in place. I found it helpful to first disconnect the signal cable and remove the disk controller card from the peripheral expansion box before removing the drive unit. Gently slide the disk drive unit out until the back is visible. Detach the power cable, the ribbon cable, and slide the unit free. Once the disk drive unit is removed, you are ready to install the Shugart dual-disk drive assembly in its place.

Installing The Shugart

You must be aware of two important things when installing the disk drive system. First, be sure you understand which drive is Drive 1 and which is Drive 2. Second, determine whether a special resistor pack is in its proper position. The manual (which is a bit on the sparse side and

could be more specific) explains what position the resistor pack should be in according to the number and type of drives you are using.

Start by opening the package of cables (34-pin signal cable and power cable). Refer to the instructions in the

| | |
|--------------|--|
| Product: | Shugart 455 Disk Drive |
| Machine: | TI-99/4A |
| Distributor: | CompuAdd Corp. 13010 Research Blvd. #101 Austin, TX. 78750 |
| Price: | \$189 per drive \$15 2-drive cable kit |

System Requirements: TI *Disk Manager 2* command cartridge, TI disk controller card, TI Peripheral Expansion System

Poor Fair Good Excellent

| | |
|-----------------|------------|
| Performance: | ██████████ |
| Ease of Set-Up: | ██████████ |
| Documentation: | ██████████ |



“While you are in fact running two drive motors, they require approximately the same amount of power as one regular drive motor—with no loss of performance.”

documentation, which show you the proper set-up of the jumper plugs on the jumper block at the back of the dual-disk drive unit. Drive 1 requires the plug to be in position 1. Drive 2 requires the plug to be in position 2 (see Photo 2). Plug the two ends of the 34-pin cable into the connectors on the drive units, and do the same with the power cable. Because there are two drives, both units need to be connected (see Photo 3). Place the Shugart disk drive unit in front of the opening in the peripheral expansion box and thread the 34-pin cable back through this opening, connect it to the disk controller card, and re-install the card into its proper slot. Now connect the power cable to the supplied adaptor and to the drives. Gently slide the Shugart disk drive assembly back into the peripheral expansion box until it is flush with the front of the box. (Note: the new unit cannot be fastened in place—the screw holes do not align.) Now you are ready to turn on the power and test it.

How It Checked Out

After making certain that everything was plugged in correctly, I turned on the power and tried to load a program from Drive 1. My first attempt failed; neither indicator light came on, and the program didn't load. Needless to say, everything *wasn't* connected properly—I had the 34-pin ribbon cables reversed.

“One important thing isn't mentioned in the documentation: The dual drive system can read both sides of a disk.”

As I mentioned earlier, the documentation is not as specific as it could be. I turned off the power, switched the cables, and tried again. This time the program loaded fine. Next, I tried to save the program to DSK2 (the second drive). The program saved perfectly. Since then, I have used the dual-disk system every day for the past few weeks with absolutely no problems. It works great, and saves me a lot of time.

The dual-disk drive unit fits nicely inside the TI peripheral expansion box, and CompuAdd recommends that you leave the units loose in there. However, you may want to slide a thin piece of cardboard between the drive units and the side of the box to give it a “friction” fit. I found that unless you plan on regularly moving the peripheral expansion box, there is no reason to worry about securely bolting in the drives—a piece of cardboard works fine.

One important thing isn't mentioned in the documentation: The dual drive system can read *both* sides of a disk. However, if you are using a *Disk Manager 1* command module, you will be able to access only *one* side of a disk (for a total of 180K bytes of storage between the two drives). In order to access both sides of a disk, you will need the *Disk Manager 2* command

Photo 1:

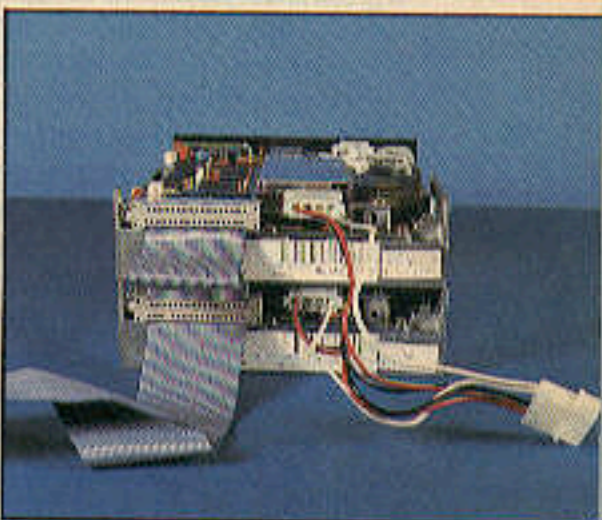
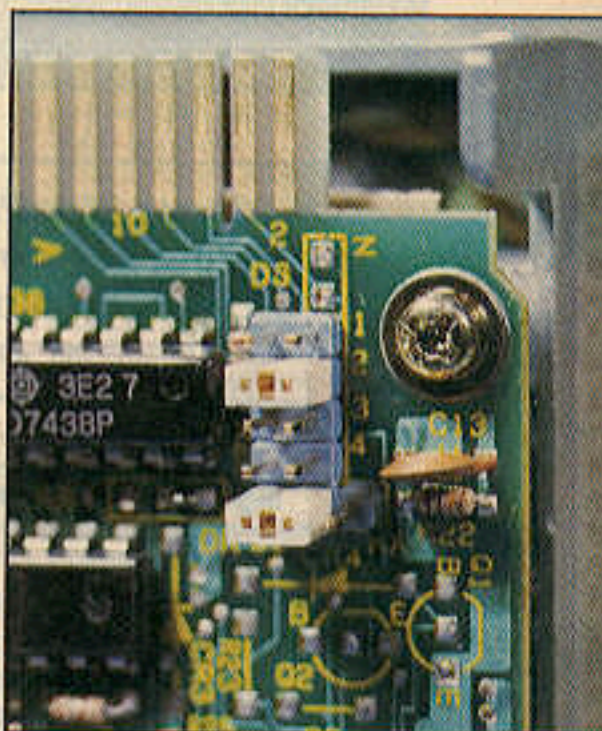
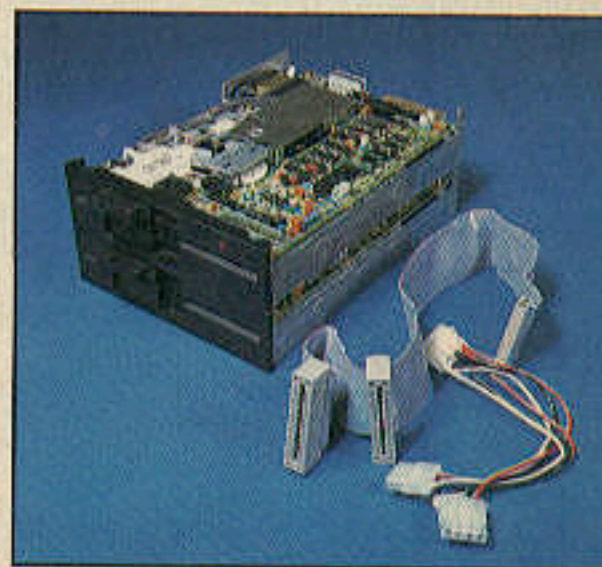
Complete kit, showing both drives, power cable and signal cable unassembled.

Photo 2:

Close up of Drive 2's jumper block with jumper plug shown in position 2.

Photo 3:

Rear of unit showing power cable and signal cable properly connected.

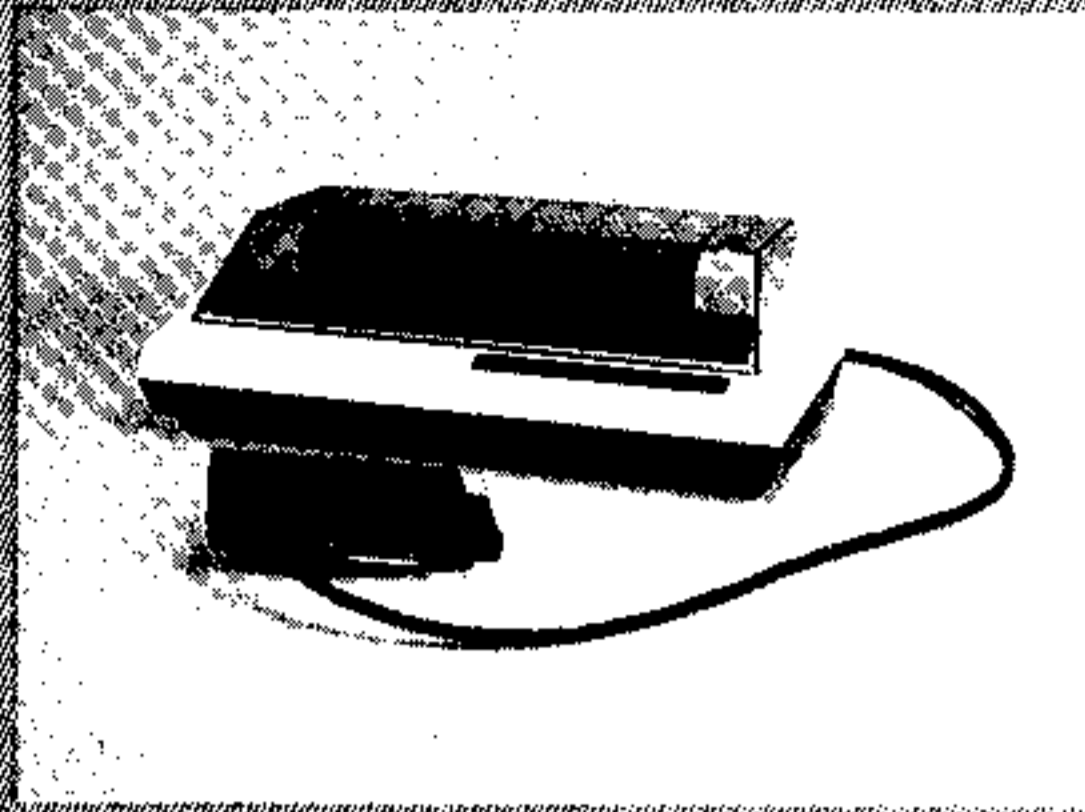


module. I recently called Texas Instruments, and they informed me that *Disk Manager 2* is still available. If you call them at 1-800-842-2737, they will refer you to a distributor who can get one for you.

A Few Complaints

The documentation should be more thorough and specific to be of any real assistance, but because the system is so easy to install, I'll say that the documentation is marginally adequate. CompuAdd has informed me that they will be amending their documentation to discuss the *Disk Manager* cartridges. Hopefully, they will make the installation procedure a little clearer at the same time. CompuAdd does provide you with a toll-free number 1-800-531-5475 for assistance if you have problems installing the drives. There is one other thing: Even though the dual-disk drive unit fits snugly, you really ought to be able to securely mount the unit in the peripheral expansion box.

Otherwise, CompuAdd's system performs great with either *Disk Manager 1* or *2* (giving you 180K or 360K respectively). This system does what it is supposed to do, and does it well. It's easy to install, and best of all, it's priced right. I recommend it. **HCM**



Name: Axiom GP-100 TI II Graphic Printer
Product Type: Impact Printer
Machine: TI-99/4A
Distributor: Axiom Corporation
 1014 Griswold Avenue
 San Fernando, CA. 91340
Price: \$299 for Printer with Parallax TI Expansion Interface. \$99 for Parallax TI Expansion Interface alone.

System Requirements: None

| | |
|------------------------|--------------------------|
| | Poor Fair Good Excellent |
| Performance: | ██████████ |
| Ease of Set-up: | ██████████ |
| Documentation: | ██████████ |

A Shortcut to 99/4A Printing:

A Review of the Axiom GP-100 TI II Graphic Printer

by **Tom Green**
HCM Staff

The Axiom Corporation—which markets a high-tech line of color graphic printers—has not turned its back on older, yet still-functional systems. A case in point is the Axiom GP-100 TI II—a printer equipped with a module (the Parallax TI Interface) that plugs directly into the side expansion port of the TI-99/4A. The module's function is threefold: **(1)** it is a quick-connect parallel interface for the Axiom GP-100 TI II (without requiring the TI Peripheral Expansion System), **(2)** it serves as a parallel print interface for "any" parallel-input printer for the TI-99/4A when ordered as a separate unit, and **(3)** it features a built-in edge connector for other standard peripherals, such as the TI Peripheral Expansion System.

Print Control Options

The general specifications for the Axiom GP-100 TI are listed in Figure 1. Interface (software) control options for text formatting include suppression of line feeds or carriage returns (thus overriding these automatic functions), added line feeds for double (or triple, or . . .) spacing, line length designation, and left margin setting.

Special printout formatting features (executed from BASIC using ASCII codes) give the programmer some powerful options. For instance, the TAB functions of TI BASIC operate with the printer, augmented by a POSition function that allows the printer to type at any column width (00-79), in any sequence. This means that you can enter characters at the end of a line, and backtrack to print more characters on the same line. All text can be changed to double-width characters; for bold type, there is an overprint option.

So far, we've concentrated on text options, but the real strength of the

Axiom GP-100 TI II is its graphics printing capabilities. The dot matrix impact head is arranged as 7 rows (or 7 needles). Each needle is addressable and can be controlled to print at any column using the ESC POSition

command. To join dot patterns between lines, there is a control code that compresses the line spacing. With these options, you can customize printouts with special symbols or high-resolution pictures.

A Trouble-Free Package

The Axiom GP-100 TI II is marketed as an inexpensive printer for the TI-99/4A, operating with or without the TI Peripheral Expansion System. It fulfills its promise, operating trouble-free and installing easily. In addition, when daisy-chained to the Peripheral Expansion System, the RS232 option is still functional.

Printer-specific errors (indicated by a light that appears on its front panel) include detection of abnormal timing between machines, and carriage return malfunctions. It's too bad there are no error detection messages like "Out of Paper" or "Change Ribbon" for those of us who don't pay attention to such details.

As indicated in Figure 1, lower- and upper-case characters are standard with the Axiom GP-100 TI II. The lower-case print, however, is of a style that some people find annoying—there are no "descenders"—that is, the let-

ters g, j, p, q, and y are printed level with the baseline. According to one representative of the Axiom Corporation, there is a trade-off between offering printers with lower-case descenders and minimizing the final retail price. They chose to keep the price at a minimum, and still offer a competitive package. And, technically, the strongest element of this printer is its graphics production, not its text.

The printer is sold with a one-year warranty, including parts and labor. If a printer requires repair, Axiom will fix and return the printer within 48 hours, or send you a new machine free of charge. Normally, the print head and ribbon wear out first on a dot-matrix printer. The replacement cost of the print head is \$49.50. Under normal usage, it has an estimated life span well beyond one year. The ribbon will last an average of 500 pages of type, and costs \$9.95 to replace.

Documentation

The Axiom GP-100 TI II manual includes set-up instructions, testing sequences, and explanations of all print execution commands. This user's guide comes complete with detailed diagrams of the interface process, and program examples of all command features. The only documentation error that came to my attention is on

Axiom's GP-100 TI II printer with parallax interface is a triple bypass operation—providing lowcost printing without the usual expansion system.

page 24—a reference is made there to the replacement of character sets, which does not apply to this printer.

Considering its dual-purpose print capabilities, its expedient installation procedure, and the compactness of its interface module, the Axiom GP-100 TI II proves to be a valuable addition to the TI home system.

HCM

| Figure 1. Axiom GP-100 TI II Printer Specifications | |
|--|--|
| 1. Character matrix | 5 by 7 dot matrix standard, with double width capability. |
| 2. Characters | Full upper/lower case characters, numerals and symbols. |
| 3. Graphics | Dot addressable 7 vertical dots per column, 480 columns maximum. |
| 4. Print speed | 30 characters/second—left to right, unidirectional. |
| 5. Maximum width | 80 columns |
| 6. Character spacing | 10 characters/inch. |
| 7. Linefeed spacing | 6 lines/inch—Character mode. 9 lines/inch—Graphic mode. |
| 8. Linefeed space | 5 linefeeds/second—Character mode. 7.5 linefeeds/second—Graphic mode. |
| 9. Paper feed | Pin feed with manual advance for fanfold-style paper. |
| 10. Paper width | 4.5 to 10 inches acceptable. |
| 11. Multiple copies | 2 including original. |

Industry Watch

SUNWARE PROVIDES CARTRIDGE CONVERSIONS FOR TI SOFTWARE

In an effort to increase software development for the TI-99/4A, two ex-TI employees have formed a firm, SunWare, to convert disk-and cassette-based software into cartridges. The firm will either market them or sell them back to the developers to market themselves. SunWare will produce two types of cartridges—a Peripheral Port cartridge which can duplicate the functions of a floppy disk (up to 48K ROM or RAM), and a Command Port cartridge, which has 32K ROM or RAM memory.

APPLE OPENS MACCOLLEGE FOR SOFTWARE DEVELOPERS

A new program has been put together by Apple Computer to assist experienced, independent programmers in developing software for the Macintosh. Dubbed MacCollege, it will be located at the company's headquarters in Cupertino, California and will provide resources and instructions to programmers certified under Apple's Certified/Registered Developers program—a support program for Apple-compatible products.

LEADING EDGE GOES AFTER IBM WITH A PEANUT OF ITS OWN

Going after the Apple //c/IBM PCjr market, Leading Edge Products of Needham, Massachusetts is reportedly preparing to mount its attack using—Peanuts. The name chosen for the PC-compatible manufactured for Leading Edge by Matsushita of Japan is (coincidentally?) the PCjr's common nickname. Declining to give specific details, Leading Edge did say that its Peanut will have a better keyboard than the Junior, and that it will be aimed at first-time home computer buyers.

TEXAS INSTRUMENTS RELEASES DEBUGGER

Texas Instruments recently announced that it has relinquished to the public, for free use, any and all proprietary claims to the Advanced Assembly Language Debugger. Copies have been mailed to all 99/4A user groups. TI further stated that this software is not covered by warranty in any fashion, and that it assumes no responsibility for its use.

RADIO SHACK GOES DOOR-TO-DOOR

Banking on the assumption that there is a large number of people who are intimidated by the entire computer-buying process, Tandy/Radio Shack is planning to bring their Color Computer 2 right into prospective buyers' homes for a demonstration. Included in this package is the Color Computer 2, educational software, joysticks, modem, and disk drive. Beginning with 13 markets, using a direct mail and magazine advertising campaign as well as 250 salespeople, Radio Shack reportedly plans to cover the entire nation within three years.

COMMODORE PLANS TO INTRODUCE A NEW COMPUTER

Relying on their flagship, the Commodore 64, Commodore International Ltd. is predicting substantial sales and profits for 1984. The company is also planning to introduce a new home computer with the power of Apple Computer's \$2,495 Macintosh at a price under \$1,000. The computer will be based on a more powerful 32-bit microprocessor, giving it better graphics and a friendlier user interface. Next year, Commodore will offer expanded memory and a higher resolution 80-column display for the Commodore 64.

Any Questions?

The present global ignorance of computing may come, in part, from our natural aversion to asking simple questions—for fear of revealing only a shallow knowledge of vital topics.

Why not let someone else ask the questions while we sit back and benefit from the reply? That's the purpose of this column.

Q. I keep hearing the term "buffer." What is it and what does it do?

A. A buffer is an interface between the Central Processing Unit (CPU) of a computer and any peripheral equipment such as a printer, disk, plotter, etc. The term is most often used to refer to a memory area that is used as temporary storage space for data until the peripheral device is ready for it. For example, printers operate at a much slower speed than microcomputers; to assure that the data to the printer is not lost, it is stored in a buffer and transmitted at a speed that the printer can handle. Similarly, most printers have buffers (memory storage areas) as part of their circuitry, so they can store data while the actual printing operation is in progress. In some instances, buffering also refers to taking care of any voltage level differences between a computer and a peripheral device.

Q. What is the difference between parallel and serial transmission?

A. Serial transmission means transmitting data one bit at a time over one signal path (wire). With parallel transmission all of the bits (usually as an 8-bit byte) are transmitted simultaneously using a separate signal path for each bit of data.

The major advantage of serial transmission is that it is generally less susceptible to electronic noise than parallel. It works better over long distances and is the method used to interface to "modems" for transmission over telephone lines.

The major advantage of parallel transmission is that it can be much faster because several bits are transmitted simultaneously. This method is very flexible and is excellent for use with digital to analog (D/A) and analog to digital (A/D) applications. Although the parallel method is faster, as the length of cable gets longer, the data transmitted may pick up electronic noise and become garbled.

Q. What is the difference between machine language and assembly language?

A. There are two basic types of languages used in microcomputers—low-level and high-level languages. Low-level languages communicate more directly with the hardware of the computer. High-level languages are easier for people to understand and act as translators between programmers and a machine's hardware. The lowest-level language is machine language. It tells your computer what to do using binary numbers. But working with binary numbers is time-consuming and error-prone. Programmers developed assembly language to speed up writing machine language programs.

Assembly language is also a low-level language, but it uses symbols called mnemonics (two or three letter abbreviations) that stand for machine language instructions that are closer to English than binary numbers. It is, therefore, much easier to work with. In order for the computer to use the assembly language program, it must be translated into machine code by a program called an "assembler."

Q. I'm thinking about buying a home computer. How much memory do I need?

A. This is a good question. The answer will vary depending on how you plan to use your computer. If you are planning to use it to store large amounts of data, run memory-intensive applications, or play complicated games like chess, then you will need more memory than someone who plans on using his or her computer to begin learning simple programming or to play video games. The amount of memory space required is directly related to the number of instructions needed to do the job assigned.

Many small computers come with less than 64K of memory. This can be enough if the computer has cartridge slots for games or specific applications, and the user is just starting out in computing—as long as the computer can be expanded to meet future needs. Examples of this type of computer are the TI-99/4A or VIC-20. Computers with 64K such as the Commodore 64, or the unexpanded Apple IIe or IBM PCjr, allow for more extensive programming applications, and should prove quite adequate for most home applications.

Finally, those computers that contain 128K or more memory are best for people who wish to use fancier word processors and more extensive business applications. One thing to remember is that as you become more proficient at computing and programming, you will eventually need more memory. You could either buy it now in the form of a more powerful computer, or add on memory as needed.

Q. Why won't programs written for the TI-99/4A work on any other machine and vice versa? If all microcomputers use BASIC as their language, why aren't programs interchangeable between systems?

A. The main reasons have to do with the different Central Processing Units (CPUs) in the various computers, and the way memory is managed by the different operating systems.

All of the data, commands, and functions needed for operation of a computer are controlled by the CPU. Because different computers use different CPUs, they require different machine-level instructions to process, store, and channel information. Even though two machines use BASIC, how a CPU interprets the BASIC instructions depends upon the way the computer translates the instructions into machine-level commands. Computers with different CPUs will interpret and even store identical BASIC statements in different formats best suited to their particular method of processing.

Another reason is in the way that memory is allocated. Even though two computers may use the same CPU, their operating systems can differ drastically. Some systems have more memory than others, and some may require a specific amount of memory to be set aside for specific functions. These differences all contribute to non-interchangeable programs.

Machine-specific functions and commands are one more reason. Every machine has commands and functions that do not work on other systems. Because of these differences, most programs are not interchangeable.

HCM



TAX DEDUCTION FILER

Tax time, 1984, has come and gone—but it's not too late to get organized for 1985. Here's a program that can help you get ready for next year's income tax preparation right now.

by Roger Wood
HCM Staff

If you itemize your deductions using Schedule A, often you can save yourself a lot of money. But keeping track of a year's worth of deductible items can indeed be a tremendous chore—especially if you, by default, rely on the “Shoobox Method” of data management (alternately referred to as “Pitch Now—Worry Later”). But hark! The Mighty Mouse of 1040 Land has come to save the day—the day's receipts, that is . . .

By saving your deductible data with the *Tax Deduction Filer* every few weeks, you can make itemizing a more pleasant, manageable, and profitable task. You should be aware that IRS forms tend to change yearly, so some minor modifications of the program may be necessary for 1984 taxes. This program is not a substitute for reading the IRS instructions, but it will help keep your records straight.

Tax Deduction Filer was inspired by a program entitled *Schedule A*, submitted by Marty Casado of Eugene, OR.

Using The Program

Here's how the program can help simplify your record keeping. Let's say you send a check for \$50 to your local public television station. Tax deductible, right? By the time next year's taxes are due though, you may have trouble remembering in which month you sent the check, let alone its amount. So you have to plow into your old checkbook registers and cancelled checks to find that deduction. With this program, however, you can sit down with your computer every few weeks, add new deductions to your files, and have all of your deductions neatly compiled for Schedule A when tax time rolls around.

The program is easy to use and totally menu-driven. The first menu asks you which part of the program you wish to access:

1. ADD DATA
2. CHANGE DATA
3. DISPLAY DATA
4. TOTALS
5. PRINT REPORT
6. LOAD DATA FILE
7. SAVE DATA FILE
8. EXIT PROGRAM

If you select any of the first three options, you are presented with a menu of the 17 deduction categories available:

1. MEDICINE AND DRUGS
2. DOCTORS, DENTISTS, ETC.
3. MEDICAL TRANSPORTATION
4. OTHER MEDICAL
5. STATE AND LOCAL INCOME TAX
6. REAL ESTATE TAX
7. MOTOR VEHICLE SALES TAX
8. OTHER TAXES
9. HOME MORTGAGE INTEREST
10. CREDIT CARD INTEREST
11. OTHER INTEREST
12. CASH CONTRIBUTIONS
13. OTHER CONTRIBUTIONS
14. CASUALTY AND THEFT
15. UNION AND PROFESSIONAL DUES
16. TAX PREPARATION FEES
17. MISCELLANEOUS

If you examine Schedule A, you will find that these categories generally follow the schedule's format. This makes it convenient to transfer your data from the program to the form at tax time.

The fourth option on the first menu displays the totals of each of the 17 categories. If you have a printer connected to your computer, the Print Report option (number 5) gives you a hard copy of your deductions.

Options 6 and 7 are for loading and saving your data files for long-term storage, and Option 8 is for exiting the program. This last option has a built-in safety feature, which inquires whether you are ready to halt the program and wipe out all the data in memory, or, if you want to first save the data before halting.

Program Implementation

When you enter data into the program you are asked to give a description of the item and an amount. The description could include to whom you paid the expense, as well as the date; its only limitation is that it cannot exceed 27 characters. (This keeps the displays easy to read.) Any time you press [ENTER] or [RETURN] without making an entry, the program returns you to the main menu to select another option.

After you have made an entry, the program stores the information in one member of the string array A\$(). This array is DIMensioned to 500, which should be plenty for most households.

Data is stored in the array in a unique fashion. When a category is selected, the program takes its number (between 1 and 17, see list above) and adds it to 100. This value is stored as the initial ASCII character in the

array element (the value would be between 101 and 117). Next, the length of the description is found using the LEN function, and 100 is added to it. This ASCII value (somewhere between 100 and 127) is placed in the array as the second character. The description and the STR\$ of the amount are "concatenated" with the first two ASCII characters to form the total string value of the element. The program can then identify the type of data in any element by simply taking the ASCII function of any member. In addition, the ASCII value of the second character is used to index the length the description and the starting location of the amount in the string.

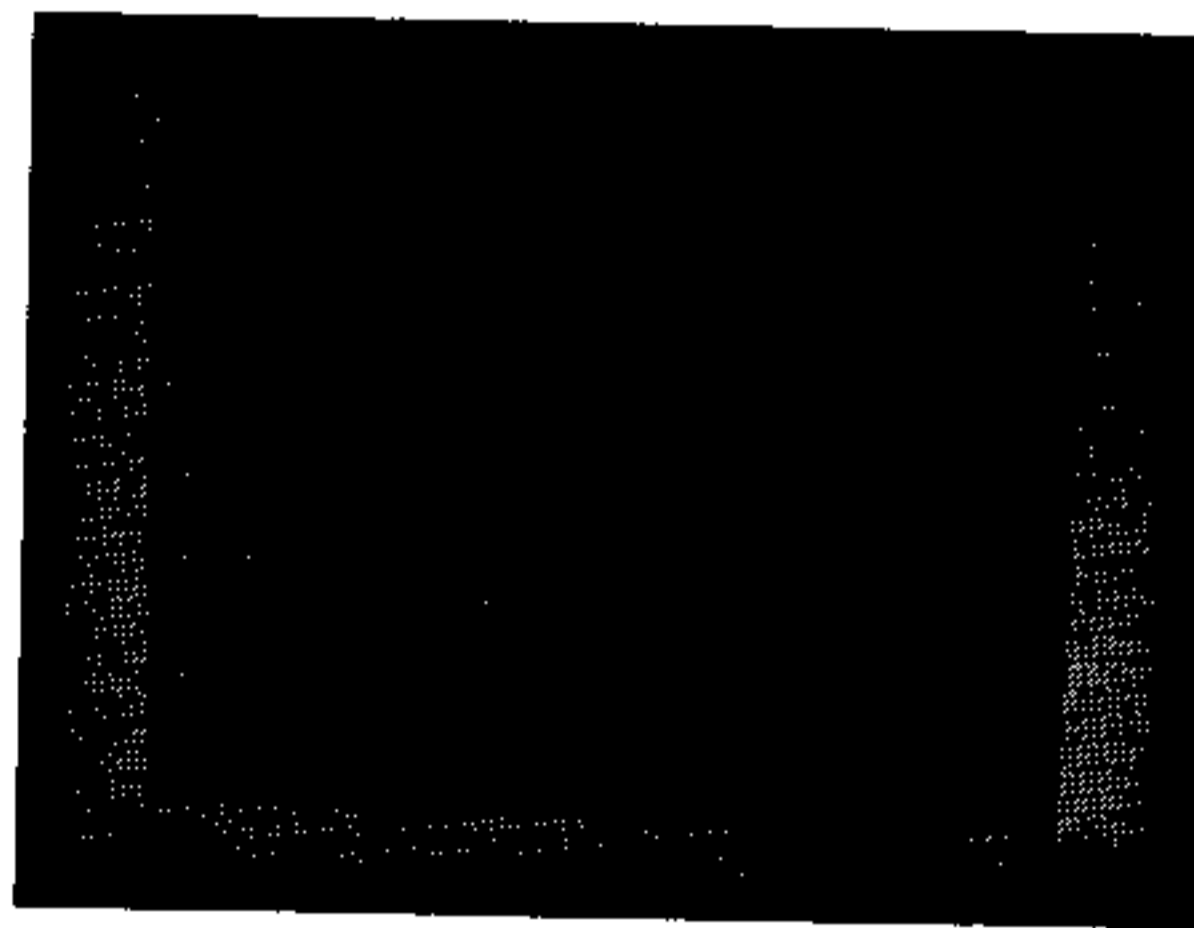
Whenever you use the program, you don't need to worry about how many items will eventually be entered in a given category, because all of the different categories

are stored in one array. The total number of items is stored in the variable R—itsself being the first one stored in the data file.

Because the program adds each item as the next available array element, it automatically keeps track of the order in which the items are entered.

For C-64, IBM PC & PCjr, and TI-99/4A see next page.

"By saving your deductible data with the Tax Deduction Filer, you can make itemizing a more pleasant, manageable and profitable task."



Apple's new ProDOS operating system removes certain bugs that were present in the old DOS 3.3 system. The Convert program that comes on the ProDOS master disk makes converting a program from one system to the other relatively easy—but the old DOS 3.3 bugs don't make it totally fool-proof.

Whenever you access the printer in a program from ProDOS, you must include PRINT CHR\$(4); in front of the PR#1 statement. Preceding PR #1 with PRINT CHR\$(4) under DOS 3.3 doesn't work with many printer interfaces (VIDEX and Grappler+, for example). However, a PR #1 without the additional CHR\$(4) works just fine.

To ensure that this program can be loaded under one operating system and then be converted to the other without difficulty, we included a prompt (see lines 200-230) so the user can tell the computer under which system the program is running. A flag (OP) is set at this point in the program so that when the printer is accessed later, the proper commands are selected (see lines 1070 and 1090).

Tax Deduction Filer (Apple) Explanation of the Program

| Line Nos. | Explanation |
|-----------|---|
| 100-170 | Program header. |
| 180-190 | DIM arrays and initialize variables. |
| 200-230 | Title screen and operating system prompt. |
| 240-280 | Main menu and keyboard input. |
| 290-300 | Display categories subroutine. |
| 310-320 | Extract one record from array subroutine. |
| 330-400 | Display category subroutine. |
| 410-480 | Add data to array. |
| 490-660 | Change data in array. |
| 670-730 | Subroutine to find a specific record. |
| 740-800 | Display a category. |
| 810-860 | Figure and display totals. |
| 870-1100 | Send data to printer. |
| 1110-1190 | Load data file. |
| 1200-1300 | Load data error trapping routine. |
| 1310-1420 | Save data file. |
| 1430-1520 | Save data error trapping routine. |

For the Key-in Listing see HCM PROGRAM LISTINGS Contents on page 93.



The C-64 version of *Tax Deduction Filer* printer routine is designed to use the VIC-1525 printer. This serial printer is treated as a sequential output file using serial channel 4. The printer does not support the regular TAB function that you use to format data on the screen. Instead, a special control code, CHR\$(16) is used. A good example of this occurs in line 1180:

```
1180 PRINT#4,"CATEGORY",CHR$(16)"10DESCRIPTION",
CHR$(16)"40AMOUNT",CHR$(13)
```

Here, the word CATEGORY is printed at the left edge of the paper. CHR\$(16) tells the printer that the next two-digit number will specify the column where the next entry is to be printed. Notice that this two-digit number must be part of a string. In this instance, the word DESCRIPTION will begin in column 10. Likewise, the word AMOUNT will begin in column 40. Notice that the numbers 10 and 40 will not be printed, even though they appear in the same quotes as the words DESCRIPTION and AMOUNT. Because the strings are preceded by CHR\$(16), the printer knows that these numbers are for formatting purposes.

For further information on this and the rest of these printer control codes, refer to the VIC-1525 manual. If you are using an RS232 interface for your printer,

then the above control codes may not be supported by your printer, and you will need to modify the Print Report section (lines 1060-1380) of the program to conform to your printer's specifications. [Would anyone who tries this please send in a Letter-to-the-Editor with specific details. It would be helpful to other readers who wish to follow in your footsteps.—Ed.]

Tax Deduction Filer (C-64) Explanation of the Program

| Line Nos. | Explanation |
|-----------|---------------------------------------|
| 100-170 | Program header. |
| 180-190 | DIM arrays and initialize variables. |
| 200-240 | Title screen. |
| 250-290 | First menu. |
| 300-310 | Display categories subroutine. |
| 320-340 | Extract one record subroutine. |
| 350-450 | Add data to array. |
| 460-710 | Change data in array. |
| 720-810 | Subroutine to find a specific record. |
| 820-970 | Display a category. |
| 980-1050 | Figure and display totals. |
| 1060-1380 | Send data to printer routines. |
| 1390-1550 | Load data file. |
| 1560-1730 | Save data file. |
| 1740-1800 | Disk error routine. |
| 1810-1870 | End program routine. |
| 1880-1950 | Data statements for menus. |

HCM

For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.



IBM's BASIC file management system is so versatile that disk and cassette options can share much of the same code in the program. For example, because both cassette and disk options use the basic DOS input and output scheme, we used the same OPEN statement for both options (see lines 1020 and 1130). F1\$ is determined by the user's input (lines 980-100 or lines 1090-1110), and no other special disk or cassette instructions are required. Most other computer systems (e.g., TI, Apple, C-64) require separate routines for each type of device. If you run the program on an IBM PC, be aware that your data disk must be in Drive A when you save or load.

Line 500 contains a useful trick for centering text on the screen—such as titles, headings, etc. It is based on a 40-column screen.

```
500 CLS:LOCATE 2,(20-LEN(N$(C))/2):PRINT N$(C).
```

Here, the horizontal location of the cursor (the start-

ing column) is determined by the LENGTH of N\$(C), where N\$(C) is the category to be displayed. Specifically, half of the length of the category name is subtracted from one-half of the width of the screen. This causes the title of the particular category to be centered on the screen.

Tax Deduction Filer (IBM PC and PCjr) Explanation of the Program

| Line Nos. | Explanation |
|-----------|--------------------------------------|
| 100-160 | Program header. |
| 170-180 | DIM arrays and initialize variables. |
| 190-230 | Title screen. |
| 240-290 | Main menu and first input. |
| 300-450 | Add data routine. |
| 460-610 | Change data routine. |
| 620-730 | Search for record. |
| 740-780 | Totals routine. |
| 790-950 | Print report routine. |
| 960-1060 | Load data file routine. |
| 1070-1170 | Save data file routine. |
| 1180-1200 | End program. |

HCM

For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.



The menu-driven nature of *Tax Deduction Filer* demands quick changes between text screens. Extended BASIC's DISPLAY AT and ACCEPT AT statements make it much more suitable to this application than TI BASIC.

In TI Extended BASIC, we use the SEG\$() command to extract the characters to help implement *Tax Deduction Filer's* unique file structure. You can see the string constructed in line 430. The subroutine in line 810 takes the string apart, returning the value of the category in AC, the item name length in AL, the item name in D\$, and the item value in V\$. This technique is very useful in simplifying the code for the program, and in reducing the amount of memory required for data storage.

This same technique is handy for a multitude of purposes: You could insert a character that represents the previous item or the next item in a search chain, thus eliminating the need for manipulating whole records when you sort. You could also use several characters as qualifying categories when searching for items. Now see if you can come up with some original

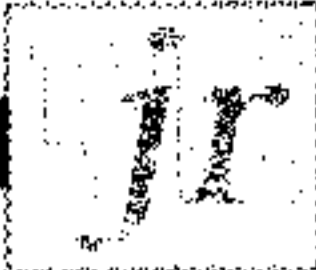
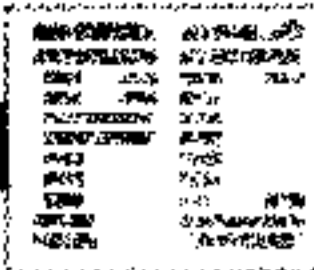
ideas for this technique. And don't forget to let us know about them in a Letter to the Editor.

Tax Deduction Filer (TI-99/4A) Explanation of the Program

| Line Nos. | Explanation |
|-----------|---|
| 100-170 | Program header. |
| 180-190 | Initialize program variables and arrays. |
| 200-220 | Display the main menu and input choice. |
| 230-280 | Add data routine. |
| 290-440 | Change data routine. |
| 450-500 | Display data routine. |
| 510-520 | Calculate and display totals. |
| 530-680 | Print reports. |
| 690-740 | Load data into memory from cassette or disk. |
| 750-800 | Save data to cassette or disk. |
| 810 | Routine to retrieve information from the data string. |
| 820 | Display mode option screens. |
| 830-860 | Input subroutines. |
| 870 | Data for the main menu screen. |
| 880-930 | Option screen data for display. |
| 940-950 | Subroutine to catch the user when trying to exit the program. Give opportunity to back out and save data first. |

HCM

For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.



COMPUTER

by **Melody Covington**
and the HCM Staff

I recently came across a kaleidoscope in a store. Fascinated by the twirling imagery, I played with it for several minutes. "I could do that on my computer," I mused. The symmetry could be mathematically controlled, and a feeling of movement would be easy to generate. I had formed rough ideas for the programs, even before I got home.

Little Stars

My first Kaleidoscope program, *Little Stars*, wasn't very difficult. My main task involved creating a symmetrical pattern, which was fairly easy to do because the characters are placed on the screen with coordinates. If I knew one set of coordinates, then three more could be calculated to produce a mirror image in each corner. A short subroutine accomplished that. The next step was to randomly select a location for the first set of coordinates.

The only task left was to design the algorithm (the method by which a task is solved). The problem was simple: For every location selected on the screen, find three more sets of coordinates that define mirror image locations of the first. By mirror image, we mean for every character in the upper left corner of the screen, there will be a similar character in the corresponding lower left, lower right, and upper right corners of the screen.

For example, let's look at a screen with 32 columns and 24 rows. If a character is placed in column 2, row 2, then we will also need a character at column 2, row 23, and so on. Now it's easy to come up with a quick solution. To find the mirror of a column, subtract the column number from 33. To find a mirror row, subtract the row number from 25. You can then use just the new column (and old row), just the new row (and old column), and both the new row and new column to find the three mirror positions. The values you subtract from the row and column will vary with the number of rows and columns on your monitor.

Now that I had an algorithm for obtaining the mirror images, I needed to find a spot on the screen to place the first character. The random number generator does this simply; its command varies slightly from machine to machine, but it always performs the same function. On most machines the `RND` or `RND(1)` function will return a number between 0 and 1. If you multiply that number times your screen width (or number of lines), it will generate a random number between 0 and your screen width (or number of lines). This way you can generate random screen coordinates. Once the coordinates for one character are generated, you can then use the mirroring algorithm to plot the other characters.

Color Kaleidoscope

This program is an extension of *Little Stars*. I just added color and graphics shapes to the previous program—the same algorithm is used to find the mirror images. Color is an important factor in graphics design. As you compare the first program with the spectacular effects of this one I'm sure you will agree. *Continued*

Linear Kaleidoscope

In this program I deviated slightly from the previous two, and added an enhancement. Instead of simply plotting random dots all over the screen, *Linear Kaleidoscope* shoots graphics characters across the screen in lines of color. The increased complexity is minor compared to the fantastic effects we can create.

This program is actually a very primitive line-drawing routine in BASIC.

The algorithm used to draw a line is much simpler than you might suspect. I just needed to generate two random numbers, which serve as the horizontal and vertical slope of the line. An inner loop is responsible for the length of the line. Each time the loop repeats, the horizontal and vertical slope factors are added to the current position on the screen. If the same factors are added every time, the line will continue in the same direction. Remember, all we need to do is draw one line, and let the mirror-plotting routine draw the others. The result is four lines racing across the screen to create a beautiful symmetrical pattern.

"Instead of simply plotting random dots all over the screen, Linear Kaleidoscope shoots graphics characters across the screen in lines of color. The increased complexity is minor compared to the fantastic effects we can create."

APPLE II Family

```

100 REM *****
110 REM * LITTLE STARS *
120 REM *****
130 REM BY MELODY COVINGTON
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.4.1
170 REM APPLE II FAMILY APPLESOFT
180 REM
190 HOME
200 REM PLACE 25 SETS OF * THEN PLACE
25 BLANKS
FOR X = 1 TO 2
220 IF X = 1 THEN MS = "*"
230 IF X = 2 THEN MS = "."
240 FOR I = 1 TO 25
250 GOSUB 290
260 NEXT I
270 NEXT X
280 GOTO 210
290 REM SUB ROUTINE
300 REM RANDOMLY GENERATE COORDINATES
310 R = INT (RND (1) * 23 + 1)
320 C = INT (RND (1) * 32 + 1)
330 REM PLOT CHARACTER PLUS 4 REFLECTI
ONS
340 VTAB R: HTAB C: PRINT MS
350 VTAB 24 - R: HTAB 33 - C: PRINT MS
360 VTAB R: HTAB 33 - C: PRINT MS
370 VTAB 24 - R: HTAB C: PRINT MS
380 RETURN
390 END
    
```

HCM

COMMODORE 64

```

100 REM *****
110 REM * LITTLE STARS *
120 REM *****
130 REM BY MELODY COVINGTON
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.4.1
170 REM C-64 BASIC
180 REM
190 PRINT CHR$(147): SC=1023: CS=55295: J=1
3
200 POKE 53280, 0: POKE 53281, 0
210 REM MAIN LOOP
220 FORM=42 TO 32 STEP -10
230 FOR I=1 TO 25
240 GOSUB 280
250 NEXT I
260 NEXT M
270 GOTO 210
280 REM SUBROUTINE
290 REM RANDOMLY GENERATE COORDINATES
300 R=INT(RND(1)*12)+1
310 C=INT(RND(1)*20)+1
320 REM PLOT CHARACTER + 3 REFLECTIONS
330 Q1=40*R+C
340 Q2=40*R+(40-C)
350 Q3=40*(24-R)+C
360 Q4=40*(24-R)+(40-C)
370 POKE SC+Q1, M: POKE CS+Q1, J: POKE SC+Q2, M
: POKE CS+Q2, J
380 POKE SC+Q3, M: POKE CS+Q3, J: POKE SC+Q4, M
: POKE CS+Q4, J
390 RETURN
    
```

HCM

VIC-20

```

100 REM *****
110 REM * LITTLE STARS *
120 REM *****
130 REM BY MELODY COVINGTON
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.4.1
170 REM VIC 20 BASIC
180 A=7680
190 PRINT "SHIFT CLR"
200 POKE 36879, 104
210 FORM=42 TO 32 STEP -10
220 FOR T=1 TO 15
230 X=INT(RND(1)*11)
240 Y=INT(RND(1)*11)
250 P=2*(10-X)
260 Q=2*(10-Y)
270 POKEA+X+22*Y, CH
280 POKEA+X+P+22*Y, CH
290 POKEA+X+P+22*(Y+Q), CH
300 POKEA+X+22*(Y+Q), CH
310 FORTD=1 TO 50
320 NEXT T, D, CH
330 GOTO 210
    
```

HCM

IBM PC & PCjr

```

100 *****
110 * LITTLE STARS *
120 *****
130 BY MELODY COVINGTON
140 AND THE HCM STAFF
150 HOME COMPUTER MAGAZINE
160 VERSION 4.4.1
170 IBM PCjr WITH CASSETTE & CARTRIDGE
BASIC
180 IBM PC WITH CASSETTE BASIC & BASIC
A
190
200 SET UP
210 RANDOMIZE TIMER
220 CLS: SCREEN 0: WIDTH 40
230 PLACE 25 SETS OF * THEN PLACE 25
BLANKS
240 FOR M=42 TO 32 STEP -10
250 FOR I=1 TO 25
260 GOSUB 300
270 NEXT I
280 NEXT M
290 GOTO 240
300 REM
310 RANDOMLY GENERATE COORDINATES
320 R=INT(RND*23)+1
330 C=INT(RND*39)+1
340 PLOT CHARACTER PLUS 4 REFLECTIONS
350 LOCATE R,C: PRINT CHR$(M)
360 LOCATE 24-R,40-C: PRINT CHR$(M)
370 LOCATE R,40-C: PRINT CHR$(M)
380 LOCATE 24-R,C: PRINT CHR$(M)
390 RETURN
    
```

HCM

TI-99/4A

```

100 REM *****
110 REM * LITTLE STARS *
120 REM *****
130 REM BY MELODY COVINGTON
140 REM HOME COMPUTER MAGAZINE
150 REM VERSION 4.4.1
160 REM TI BASIC
170 REM
180 REM SET UP
190 RANDOMIZE
200 CALL SCREEN(2)
210 CALL CLEAR
220 CALL COLOR(2, 16, 2)
230 REM PLACE 25 SETS OF * THEN PLACE
25 BLANKS
240 FOR M=42 TO 32 STEP -10
250 FOR I=1 TO 25
260 GOSUB 300
270 NEXT I
280 NEXT M
290 GOTO 240
300 REM SUBROUTINE
310 REM RANDOMLY GENERATE COORDINATES
320 R=INT(RND*24)+1
330 C=INT(RND*32)+1
340 REM PLOT CHARACTER PLUS 4 REFLECTI
ONS
350 CALL VCHAR(R,C,M,1)
360 CALL VCHAR(25-R,33-C,M,1)
370 CALL VCHAR(R,33-C,M,1)
380 CALL VCHAR(25-R,C,M,1)
390 RETURN
    
```

HCM



Little Stars

Two FOR-NEXT loops in this program control the action. The first (outer loop) starts in line 240, and determines whether an asterisk or a space is to be plotted. The second (inner) loop starts in line 250 and dictates the number of characters plotted. These two loops will first plot 25 asterisks, then 25 blanks, then 25 more asterisks, and so on.

Color Kaleidoscope

Here, the TI computer assigns color to groups of eight characters so that the program can draw with up to eight different colors at the same time. Each group can have its own foreground and background color. (They are assigned in lines 400 through 470.) The first (outer) loop has been altered slightly from the first program. It still decides which characters are plotted, but the characters now have an ASCII value in the range of 96 to 152. These are the characters which were assigned colors earlier.

APPLE II Family

```

100 REM *****
110 REM * COLOR KALEIDOSCOPE *
120 REM *****
130 REM BY MELODY COVINGTON
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.4.1
170 REM APPLE II FAMILY APPLESOFT
180 REM
190 HOME
200 GR
210 POKE -16302,0
220 FOR Y = 40 TO 47
230 FOR X = 0 TO 39
240 COLOR = 0: PLOT X,Y
250 NEXT X: NEXT Y
260 REM LOOP TO CONTROL COLORS
270 FOR A = 1 TO 15
280 COLOR = A
290 REM LOOP TO CONTROL NUMBER OF LINE
    S DRAWN
300 :
310 FOR D = 1 TO 25
320 REM RANDOMLY SELECT BEGINNING COOR
    DINATES
330 R = INT (RND (1) * 39)
340 C = INT (RND (1) * 47)
350 REM PLOT SPOT PLUS 3 REFLECTIONS
360 PLOT R,C
370 PLOT R,47 - C
380 PLOT 39 - R,C
390 PLOT 39 - R,47 - C
400 NEXT D
410 NEXT A
420 REM DELAY ROUTINE
430 FOR M = 1 TO 1000
440 NEXT M
450 GOTO 200

```

HCM

VIC-20

```

100 REM *****
110 REM * COLOR KALEIDOSCOPE *
120 REM *****
130 REM BY MELODY COVINGTON
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.4.1
170 REM VIC 20 BASIC
180 REM
190 A=233: B=223: C=105: D=95: K=7680: J=384
200 PRINT "SH: FT CLR"
210 POKE 36879,104
220 FOR T=1 TO 15
230 X=INT(RND(1)*11)
240 Y=INT(RND(1)*11)
250 P=(2*(10-X))+1
260 Q=(2*(10-Y))+1
270 POKEK+X+22*Y,A
280 POKEJ+X+22*Y,CC
290 POKEK+X+P+22*Y,B
300 POKEJ+X+P+22*Y,CC
310 POKEK+X+P+22*(Y+Q),C
320 POKEJ+X+P+22*(Y+Q),CC
330 POKEK+X+22*(Y+Q),D
340 POKEJ+X+22*(Y+Q),CC
350 FOR TD=1 TO 50
360 NEXT TD, T
370 CC=INT(RND(1)*8): IF CC=6 THEN 370
380 IFA=32 THEN A=233: B=223: C=105: D=95: GO
    TO 210
390 A=32: B=32: C=32: D=32: GOTO 210

```

HCM

Linear Kaleidoscope

We now add the final stage to this program. A slope to draw the line is determined in lines 370 and 380. The loop in line 400 causes the line to continue drawing for 15 characters. Lines 410 and 420 add the slope values to the current position of the line, to continue its path.

TI-99/4A

```

100 REM *****
110 REM * COLOR KALEIDOSCOPE *
120 REM *****
130 REM BY MELODY COVINGTON
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.4.1
170 REM TI BASIC
180 REM
190 CALL SCREEN(16)
200 CALL CLEAR
210 RANDOMIZE
220 REM DEFINE CHARACTERS
230 S$="0103070F1F3F7FFF"
240 T$="80C0E0F0F8FCFEFF"
250 U$="FF7F3F1F0F070301"
260 V$="FFFEFCF8F0E0C080"
270 FOR A=96 TO 152 STEP 8
280 CALL CHAR(A,V$)
290 NEXT A
300 FOR B=97 TO 153 STEP 8
310 CALL CHAR(B,T$)
320 NEXT B
330 FOR C=98 TO 154 STEP 8
340 CALL CHAR(C,U$)
350 NEXT C
360 FOR D=99 TO 155 STEP 8
370 CALL CHAR(D,S$)
380 NEXT D
390 REM DEFINE COLORS
400 CALL COLOR(16,14,15)
410 CALL COLOR(15,12,10)
420 CALL COLOR(14,9,16)
430 CALL COLOR(13,7,11)
440 CALL COLOR(12,15,14)
450 CALL COLOR(11,10,12)
460 CALL COLOR(10,16,9)
470 CALL COLOR(9,11,7)
480 REM THESE LOOPS PLACE 25 SETS OF E
    ACH COLOR COMBO
490 FOR M=96 TO 152 STEP 8
500 FOR I=1 TO 25
510 GOSUB 550
520 NEXT I
530 NEXT M
540 GOTO 490
550 REM SUBROUTINE
560 REM RANDOMLY GENERATES COORDINATES
570 R=(RND*12)+1
580 C=(RND*16)+1
590 REM PLOT CHARACTER PLUS THE REFLECT
    IONS
600 CALL VCHAR(R,C,M)
610 CALL VCHAR(25-R,C,M+1)
620 CALL VCHAR(R,33-C,M+2)
630 CALL VCHAR(25-R,33-C,M+3)
640 RETURN

```

HCM

IBM PC & PCjr

```

100 *****
110 * COLOR KALEIDOSCOPE *
120 *****
130 BY MELODY COVINGTON
140 AND THE HCM STAFF
150 HOME COMPUTER MAGAZINE
160 VERSION 4.4.1
170 IBM PCjr CASSETTE & CARTRIDGE BASI
    C OR
180 IBM PC CASSETTE BASIC & BASICA WIT
    H
190 COLOR/GRAPHICS MONITOR ADAPTER
200 AND COLOR MONITOR
210 REM SET UP
220 RANDOMIZE TIMER
230 CLS: SCREEN 0: WIDTH 40
240 REM PLACE 25 SETS OF *, THEN PLACE
    25 BLANKS
250 FOR M=219 TO 32 STEP -187
260 COLOR INT(RND*16)
270 FOR I=1 TO 10
280 GOSUB 320
290 NEXT I
300 NEXT M
310 GOTO 250
320 REM
330 REM RANDOMLY GENERATE COORDINATES
340 R=INT(RND*23)+1
350 C=INT(RND*39)+1
360 REM PLOT CHARACTER PLUS 4 REFLECTIO
    NS
370 LOCATE R,C: PRINT CHR$(M);
380 LOCATE 24-R,40-C: PRINT CHR$(M);
390 LOCATE R,40-C: PRINT CHR$(M);
400 LOCATE 24-R,C: PRINT CHR$(M);
410 RETURN

```

HCM



THE BOOLEAN BRAIN

by **W.K. Balthrop**
HCM Staff

*Captured by your computer,
you wind up exploring
its inner workings.*

[Editor's Note: In the August issue of Home Computer Magazine we featured "The Boolean Brain" for the IBM PCjr and Apple II family. We bring it back this month for those readers with TI-99/4A and Commodore 64 computers.]

It is late. Another night of blasting aliens draws to a close. You reach to turn off the computer—but it's not through playing! Suddenly, before your hand can touch the switch, the screen flashes bright red. Then the message CPU Error appears briefly and vanishes, leaving the screen totally blank.

What has happened? Has your computer died? You pounce on the keyboard, hoping to save your system before it's too late. Instead, a tingling surge of electricity grabs and holds your arms fast. With horror, you realize something is pulling you in, in . . .

This must be a nightmare you think. But when you open your eyes, there's a new shock awaiting: You are in a strange, brightly lit room—a room that looks remarkably like the inside of a computer. Thus, stranded in the *Keyboard Room*, you suddenly recall the message about a CPU failure. Perhaps if you can make it to a room with the Central Processing Unit, you can fix the problem and get out of your silicon cell.

The Program

The *Boolean Brain* program is a combined adventure game and "logical" learning experience. Your goal is to find the computer's CPU. To do this, you will have to open the locked doors of each room, and gain access to other rooms. Each door is secured with a logic lock.

As the game begins, you start out in the *Keyboard Room*. Here you will see a three-dimensional picture on the screen of three of the four walls. In each wall is a door, and to the right of each door is a control panel. On the wall in the center of the screen is one of four letters which indicate the direction you are facing. To move in any of the four directions, simply press either E, W, N, or S. Closed doors are red, and when you try to go through one you will be taken to another screen. This screen will display the computer logic gates that you must activate to open the door.

*"This must be a nightmare you think.
But when you open your eyes,
there's a new shock awaiting . . ."*

The two types of logic gates used in the lock look and operate quite differently. The AND gate—with its left side squared off—requires *both* of its inputs to be turned on before it will pass its output. The OR gate—resembling an arrow head—will pass its output when *either* of the two inputs are turned on. The output of the first AND and OR gates will feed the input of other AND and OR gates. The logic paths that are turned on will become green, and the lock will open when you have succeeded in completing a logic path to the right side of the screen.

On the left side of the screen are 10 input lines to the five gates. To activate an input, you simply press the number on the keyboard for the input line you desire. You want to open a path with the fewest number of inputs possible. The computer keeps track of how many inputs you use throughout the game, so if you don't learn to be a "Boolean Brain," you may not be able to escape from the computer.

(One word of warning to those who venture carelessly: There is one trap hidden in the game . . . beware bad disk sectors.)

HCM

Boolean Brain (TI-99/A) **Explanation of the Program**

| Line Nos. | Explanation |
|-----------|---|
| 100-170 | Program header. |
| 180-250 | Initialize program variables. |
| 260-270 | Input direction. |
| 280-310 | Hit a bad disk sector. |
| 320-350 | Go through open door. |
| 360-420 | Display computer rooms. |
| 430-490 | Display logic gates, and get input. |
| 500-770 | Gate control logic. |
| 780-910 | Found control room. End of game and option to play again. |
| 920 | Subroutine to read the keyboard. |
| 930-1310 | Program DATA. |

For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.

Boolean Brain (C-64) **Explanation of the Program**

| Line Nos. | Explanation |
|-----------|--|
| 100-170 | Program header. |
| 180-220 | Initialize program variables. |
| 230-910 | Draw the computer rooms, and input direction. |
| 920-940 | Check for an open door. |
| 950-1190 | Draw the logic gates, and get input. |
| 1200-1610 | Gate control logic. |
| 1620-1970 | Subroutines to draw the gates. |
| 1980-2340 | Found the control room. End of game, and option to play again. |
| 2350-2420 | Sound routines. |
| 2430-2470 | Program DATA. |

For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.



STADIUM

by Kent and Kathy Gemmel
and the HCM Staff

*This BASIC program captures
an age-old sport in a
simple simulation game.*

its right, meaning, as the horse sees right. If the horse is heading down toward the bottom of the screen, the horse's right will be your left. This may seem confusing at first, but after you have played a few games it will come as second nature.

This is no rodeo. Riders sit erect and without expression, their caps and uniforms spotless and perfect in every detail. Their well-groomed horses stand in formal posture. As a fine horse and its rider glide through the obstacle course of high fences, every movement is precise, deliberate, and subject to judgement.

Stadium Jumping is an equestrian simulation game in which you must ride your horse through a pre-planned course and jump fences—preferably without knocking down the poles or yourself in the process. You are scored on how many "faults" you have. Each time you knock down a pole you will receive four faults. A perfect score is 0, or no poles knocked down. If you fall from your horse, you won't be able to get back on to complete the course, and will have to start over again. Here are the general rules of *Stadium Jumping*:

1. All fences must be jumped in the proper order. Fences are numbered according to the order in which they must be jumped.
2. All fences must be jumped in the correct direction. A flag is posted on the side of every fence and, in some systems, this flag may be the sequence number itself. You must jump the fence squarely, with the flag to your horse's right.
3. Attempting to make the horse jump without a fence will cause you to fall from your horse. Simply missing the fence will also cause you to fall.
4. If you fail to jump a fence and instead collide with it, you will be faulted. You also run a high risk of falling from your horse.
5. After you have jumped the last fence, your score will be displayed.
6. You may run your horse around the screen as much as you like without penalty.

Three Skill Levels

Three skill levels are available in this game. The first requires only four jumps to complete a round. You have 7 fences to jump on the second level, and the third level has 11 fences and is for masters only.

Horse's Point Of View

Only two keys on the keyboard, or two directions on the joystick, turn the horse. When you press the right button or pull the joystick right, the horse will turn to

When the game begins, your horse will be in the upper left corner of the screen. To start him off, press K for Kick whether you are using the keyboard or a joystick. The horse will start running, and will not stop until the game is over.

| KEYBOARD | ACTION |
|------------------|---------------------------------|
| S..... | Turn horse to its left. |
| D..... | Turn horse to its right. |
| J..... | Make horse jump. |
| K..... | Kick horse to start the course. |
| JOYSTICK | ACTION |
| STICK LEFT..... | Turn horse to its left. |
| STICK RIGHT..... | Turn horse to its right. |
| FIRE BUTTON..... | Make horse jump. |

Stadium Jumping (TI-99/4A) Explanation of the Program

| Line Nos. | |
|-----------|--|
| 100-180 | Program header. |
| 190-290 | Initialization. |
| 300-380 | Input skill level. |
| 390-520 | Get graphics shapes and color assignments. |
| 530-1780 | Display initial arena and fences for each skill level. |
| 1790-1910 | Start of game. Wait for K to be pressed. |
| 1920-2000 | Scan keyboard, check for a jump. |
| 2010-2180 | Check for a direction change. |
| 2190-2460 | Change direction and horse shape routines. |
| 2470-2740 | Routine for fouls. |
| 2750-2880 | Rider has fallen. |
| 2890-3380 | Routine to jump the horse. |
| 3390-3580 | Finished round. |
| 3590-3700 | Option to play again. |
| 3710-3820 | Input new level, start new game. |
| 3830-3870 | Routine to print without scrolling. |
| 3880-3910 | Graphics character and color assignment data. |

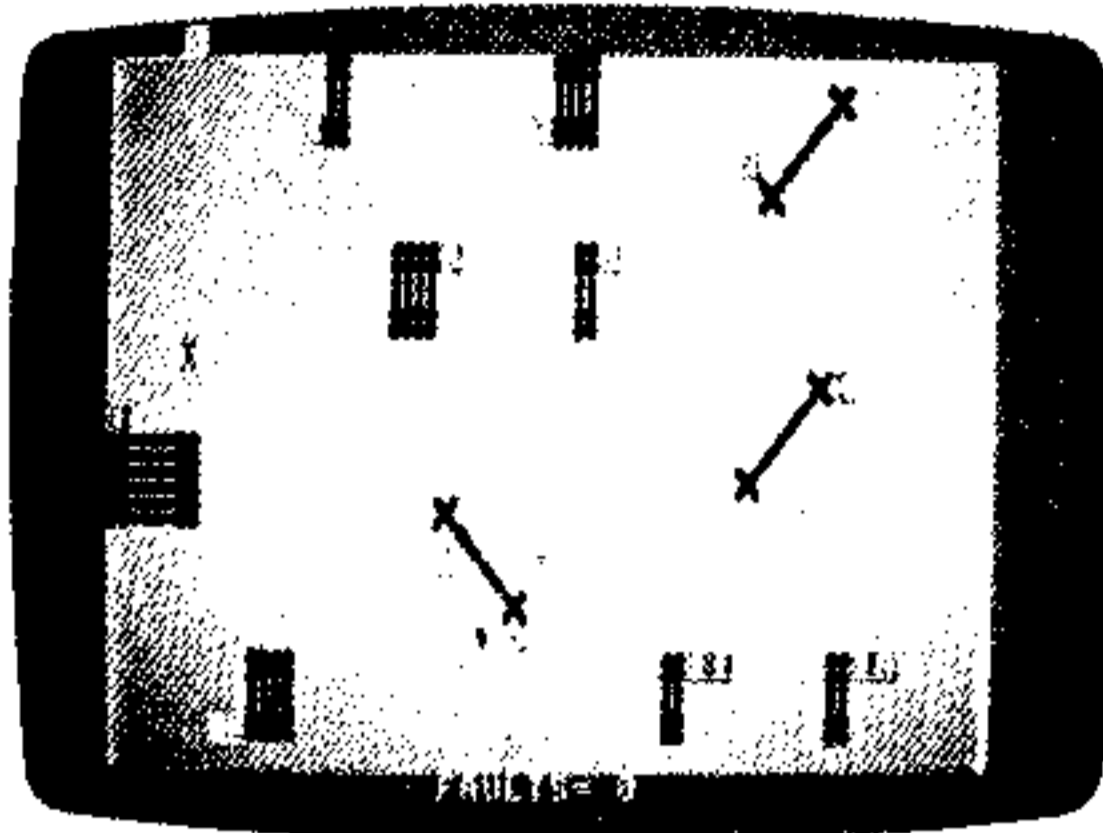
For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.

PC jr

Once you have selected one of the three skill levels, the arena will appear on the screen, with your horse waiting in the upper left corner. To start the horse, press any key—the horse will continue across the arena by itself.

Turn the horse by using the left and right cursor control keys. Once you hear the beep, you can let go of the key. However, the horse may not turn instantly, because

JUMPING



the program will still have to go through its paces before it can update the horse's shape and direction.

If you prefer to play the game without sound effects, press (F1) on the PC,

and (Fn)(1) on the PCjr. Pressing this key again will turn the sound effects back on.

| KEY | ACTION |
|--------------|---------------------------|
| LEFT Cursor | Turn horse to the left. |
| RIGHT Cursor | Turn horse to the right. |
| UP Cursor | Make horse jump. |
| (Fn)(1) | Turn sound off or on. |
| Any Key | Kick horse to start game. |

Stadium Jumping (IBM PC/PCjr) Explanation of the Program

| Line Nos. | Explanation |
|-----------|---|
| 100-200 | Program header. |
| 210-350 | Initialize program graphics and variables. |
| 360-370 | Get skill level. Branch to display arena. |
| 380 | Wait for a key to be pressed. Start the game. |
| 390-420 | Main control loop. |
| 430-460 | Interrupt routines. |
| 470-670 | Routine to display the three arenas. |
| 680-1010 | Routine to jump the horse. |
| 1020-1140 | Routine to turn the horse. |
| 1150-1250 | Move horse and check for collisions. |
| 1260-1330 | Check to see if jump went over the right fence. |
| 1340-1350 | Foul. |
| 1360-1390 | Rider has fallen. |
| 1400-1410 | Option to play again. |
| 1420-1440 | Round has been completed. |
| 1450 | Subroutine to read the keyboard. |
| 1460 | Get information for the skill level from the data statements. |
| 1470-1510 | DATA for fences. |

For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.



The Apple version of Stadium Jumping has an added feature not found in the other versions: While playing the game, you can change the speed of the horse by pressing keys 1 through 5.

| KEY | ACTION |
|--------------|---------------------------------------|
| LEFT Cursor | Turn horse to its left. |
| RIGHT Cursor | Turn horse to its right. |
| SPACE BAR | Make the horse jump. |
| K | Kick horse to start the game. |
| 1 through 5 | Change horse's speed. Default = 2. |

| JOYSTICK | ACTION |
|-------------|-------------------|
| Stick left | Turn horse left. |
| Stick right | Turn horse right. |
| Fire button | Make horse jump. |

Stadium Jumping (Apple II Family) Explanation of the Program

| Line Nos. | Explanation |
|-----------|---|
| 100-190 | Program header. |
| 200-330 | Initialize program and save machine code routines to memory. |
| 340-380 | Input options. |
| 390-440 | Get skill level. |
| 450-1040 | Set up initial arena and display fences for each level. Branch to display fences. |
| 1050-1070 | Start game. Wait for K to be pressed. |
| 1080-1180 | Read keyboard and joystick. |
| 1190-1260 | Routines to turn the horse. |
| 1270-1390 | Foul routine. |
| 1400-1420 | Rider has fallen. |
| 1430-1710 | Routine to jump the horse. |
| 1720-1740 | Round complete. |
| 1750-1770 | Option to play again. |
| 1780-1790 | Display number of fouls on the high-res screen. |
| 1800-1810 | Display a flashing border on the screen. |
| 1820 | Display the current number of gates completed. |

****NOTE:** A word of caution to anyone who may want to resequence this program in the future: A machine language routine is in use which restores DATA statements to a particular line number so that they can be read again. This routine is called RESTR, and it starts at address 2138. If you resequence this program, you will need to change any line references where CALL RESTR is used. The value passed in the routine is the line number containing the DATA statement being tested.

For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.



When the arena is first displayed, your horse will be waiting in the upper left corner of the screen. To start the game and make your horse enter the arena, press K to kick the horse. Once the horse starts moving, it will not stop until the end of the round.

| KEY | ACTION |
|-----|------------------------------------|
| S | Turn horse to its left. |
| D | Turn horse to its right. |
| J | Make the horse jump. |
| K | Kick the horse to start the round. |

| JOYSTICK | ACTION |
|-------------|--------------------------|
| Stick left | Turn horse to its left. |
| Stick right | Turn horse to its right. |
| Fire button | Make the horse jump. |

Stadium Jumping (C-64) Explanation of the Program

| Line Nos. | Explanation |
|-----------|---|
| 100-170 | Program header. |
| 180-250 | Title screen. |
| 260-310 | Get skill level. |
| 320-400 | Define graphics characters. |
| 410-1050 | Draw arena and fences. |
| 1060-1110 | Start game. Wait for K to be pressed. |
| 1120-1350 | Read keyboard and joystick. |
| 1360-1470 | Update position pointers and character shape. |
| 1480-1680 | Crash routines. Do faults. |
| 1690-2010 | Jump routine. |
| 2020-2120 | End of the round. Option to play again. |
| 2130-2180 | Enter level for a new game. |
| 2190-2270 | Graphics data for character shapes. |

For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.

HCM

M A R K E T

A

Shouts arise everywhere from the boisterous—even frantic—crowd. Many wave small sheets of paper, others yell into telephones as the numbers change on the lighted board.

This is the world of profit and loss—the instant buy and sell decisions that amass paper fortunes and shatter dreams . . .

Could you survive here?

D

by Brian Lee
and the HCM Staff

N

Sit down to a real investment—your computer—and put your market skills to the test. *Market Madness* is a game that simulates stock transactions for six companies. Up to ten people can play at one time, each with their own personal portfolio. Each player begins the game with \$5000 cash, and one turn is the equivalent of one week of time. The game can last from two to 999 weeks. Players can buy and sell stock from the exchange, at exchange rates, or they can buy and sell to each other, setting their own prices. There is even a bank that will loan you money, if you're not a bad credit risk.

E

S

S

Playing The Game

The main game screen displays the stock prices and a menu for the first player. The stock price section of the screen includes the names of the stocks, their current value, and the

number of shares of each stock you currently own. The menu at the bottom of the screen lists six options, which can be selected by typing the first letter of the desired option:

| | |
|--------|------------|
| B)uy | S)ell |
| T)rade | L)oan |
| N)ext | P)ortfolio |

B)uy Stocks

This option lets you purchase stocks at the going rate displayed on the screen. The first prompt will ask you which stock you wish to buy. Enter a number from 1 through 6 to indicate the desired stock. The only time you will not be allowed to buy a stock under any circumstances is when the company is bankrupt. (See the section on bankruptcy). The second prompt will ask you how many shares you wish to buy. If you have enough money to make the purchase, the message **TRANSACTION COMPLETE** will be displayed.

Any time you are prompted to enter a number, you can also type 0 (zero) and return to the menu screen. At that point, you may notice that there was more



money deducted from your account than what you should have paid for the stock. This extra charge is the broker's fee. The only time you don't have to pay a broker's fee for exchanging stocks is when trading with other players or cashing in stocks at the bank.

S)ell Stocks

This option works the same as the Buy Stocks option. You are prompted for the stock you wish to sell; if the company is bankrupt or if you don't own any of that stock, you will not be able to sell it. If you can sell the stock, then you are asked for the number of shares you wish to sell. You can enter all or part of your holdings in that stock, but of course, you can't sell more shares than you own. The money from the sale, minus the broker's commission, is then transferred to your cash assets.

T)rade Stocks

This option takes you to another menu screen where you can select one of three options:



1. Trade to another player.
2. Trade for another stock.
3. Cash shares into the bank.

-1. Trade to another player.

If you are the only player in the game you can't use this option. After selecting the trade option you will be asked to choose someone to trade with. Sorry, you can't trade with yourself. After selecting someone to trade with, you will be asked if you want to sell to that person, or buy from that person. When trading stock, players can set their own price for it, within certain limits. The price can't exceed twice the market value, or be less than one half the market value. After the seller enters a price for the stock, the buyer will be asked whether he or she agrees with the transaction. The players then bargain, and if they don't agree, they are taken back to the main menu.

-2. Trade for another stock.

In this option, you will be able to trade stock you currently own for any other stock straight across, as long as the company is not bankrupt.

When trading stock, the value of the stock you trade will not always be evenly divisible by the cost per share of the stock you want to receive. Thus, any money left over from the trade will be transferred to your cash assets. You must always trade enough stock to receive at least one share of the new stock.

"The old adage holds true: buy low and sell high. However, this practice is not without its price."

-3. Cashing into the bank.

You may sell your stock to the bank once per turn. You will be asked to enter the name of the stock you want to cash in, and the number of shares you wish to sell. The bank will then make an offer on the stock which may be a little higher or lower than its market value. If you decline the offer, then the program will return to the trading menu screen. If you accept the offer, the money you receive will be added to your cash assets.

L)oins

Selecting this option takes you to another menu screen where you can select one of four options:

1. Take out a loan
2. Pay back a loan
3. Compound interest on a future loan
4. Main menu (Apple and C-64)
0. Main Menu (TI and IBM)

-1. Take out a loan

If you select this option, you will be prompted for the amount you wish to borrow. You can't borrow more than your credit limit, which is calculated to be your total net worth or \$5000, whichever is greater. If your credit limit is \$5000, and you already have a loan out for \$4000, then you would only be eligible to borrow another \$1000. You will never have a credit limit below \$5000, and are not penalized if your total net worth (credit limit) drops below your current loan balance. This simply means that you will not be able to take out any more loans until your credit limit once again exceeds your loan balance.

Each week during the game you will have to pay

interest on the balance of your loan. The interest you pay is automatically deducted from your cash assets at the end of the week. If you don't have any cash assets, the interest will be tacked on to your loan balance. This is the only instance in which your loan balance can be increased above your credit limit. No automatic payments are made on your loan balance; however, it is in your best interest to pay off your loans as soon as possible. Your outstanding loans will be deducted from your net worth at the end of the game when the scores are displayed.

-2. Pay back a loan

If you are not in debt when you select this option, you will be advised that you do not owe any money. Otherwise, you will be asked for the amount you wish to repay. You can only use your cash on hand or cash assets to pay back a loan. If you overpay your debt, only the amount you owed will be deducted from your cash assets. This is a handy feature for those who may be a little overzealous to cancel their debts.

-3. Compound interest on a future loan.

This option lets you calculate how much interest must be paid on a loan for the remaining weeks in the game. The calculation assumes that you intend to keep the loan at its present balance for the duration of the game. You will not be taking out a loan with this option—it is meant as a calculator only, to estimate future interest payments.

| STOCK PLUS | CURRENT PRICE | NO. OF SHARES | ORIGINAL PRICE |
|-------------|---------------|---------------|----------------|
| 1. US STEEL | \$45 | 0 | \$8 |
| 2. FORD | \$24 | 0 | \$8 |
| 3. SANVC | \$40 | 0 | \$8 |
| 4. XEROX | \$44 | 0 | \$8 |
| 5. AT&T | \$46 | 0 | \$8 |
| CASH | \$6800 | | |
| LOAN | \$1000 | | |
| TOTAL | | | \$5000 |

PRESS ANY KEY

N)ext

This option from the main menu terminates your turn, but you will get one chance to back out first and continue your turn. When your turn is over, it will become the next player's turn; if you are the last

player in the round, then it will also be the end of the week, and the first player's turn will occur next, starting a new week.

P)ortfolio

The Portfolio screen displays a player's current financial status. Included in the display are the current market price, the number of shares invested, and the net worth of your investment in each stock. Also displayed are your current cash assets, total investment worth, loan balance, and total net worth. Your total net worth is actually your cash assets plus the value of your investments, minus any loans you may have. This value is used to adjust your credit limit when you apply for a loan.

There are two ways to display the Portfolio screen. If you press P while at the main menu, the Portfolio screen will be displayed. To return to the main menu, press either [RETURN] or [ENTER], depending on your system.

Whenever you buy or sell stock directly with the market, the Portfolio screen will be displayed after a successful transaction. If you have a TI or IBM computer, you may return to the main menu by pressing [ENTER]. On the Apple and C-64 you will get a prompt saying ANOTHER TRANSACTION (Y/N)? If you

enter Y to this prompt, you will be taken back to the menu screen. If you enter N your turn will end, and the next player's turn will start.

The Smart Marketeer

After playing the game for awhile you may notice a pattern or trend for certain stocks. The program is written to simulate real market situations as closely as possible. Stocks generally do not wildly fluctuate back and forth at random—they generally exhibit short term trends.

The two important influences in the program are the general trend adjustment, and the activity adjustment. The general trend indicates in which direction the stock is likely to move (up or down). All trends have a life span of four weeks. After four weeks, a stock's trend will take on totally new random values.

The activity factor indicates the fluctuation of a stock (how much it will change each week). The most any stock can change in one week is 20 points (or \$20 in value).

Stock Splits

When a stock reaches a value of more than \$150, the stock will split. This means that the value of the stock will be cut in half, and you will receive double the number of shares for that stock. You will lose nothing in value when this happens—in fact, it's to your advantage that it does happen every once in a while. Because the stocks are limited to how much they can

fluctuate, expensive stocks do not make very much profit. If you have 10 shares of a stock worth \$150 per share, you would have \$1500 invested.

If that stock went up \$10 per share, your \$1500 investment would earn you \$100 in profit. However, if you had 10 shares of a stock worth \$20, your investment would only be \$200. Then if the stock went up \$10, you would make \$100 in profit on a \$200 investment.

Bankruptcy

Occasionally a company will go bankrupt. This happens when a stock's value reaches \$0 (zero dollars). If you have any shares in a company at the time it goes bankrupt, they will be lost and your holding in the stock will be reset to zero shares. When a company is bankrupt you can't buy or sell its stock. It may recover from bankruptcy, but you will not get back those shares lost when bankruptcy occurred.

Strategy

The old adage holds true: buy low, sell high. However, this practice is not without its price. Keep in mind that stocks with a value of \$20 or less could conceivably go bankrupt in only one week. There is a good chance that you could lose everything you invested in such a stock. When investing in speculative stock it is a good idea to diversify your investments. (Don't put all your eggs in one basket.) Then, if one stock goes under, you will still have several other investments to keep you going.

HCM

Market Madness (Apple II Family) Explanation of the Program

| Line Nos. | Explanation |
|-----------|----------------------------------|
| 100-170 | Program header. |
| 180-290 | Initialize program. |
| 300-600 | Display the main menu. |
| 610-710 | Buy stock routine. |
| 720-810 | Sell stock routine. |
| 820-920 | Main menu for trading. |
| 930-1370 | Trading with other players. |
| 1380-1480 | Cash stock into the bank. |
| 1490-1610 | Trade for another stock. |
| 1620-1730 | Main menu for loans. |
| 1740-1820 | Take out a loan with the bank. |
| 1830-1930 | Pay back a loan. |
| 1940-2020 | Calculate interest on a loan. |
| 2030-2070 | Routine for the Next option. |
| 2080-2300 | Display the portfolio. |
| 2310 | Routine for the continue prompt. |
| 2320-2350 | Update for end of the week. |
| 2360-2500 | End of the game routine. |
| 2510-2550 | Routine to PEEK the keyboard. |

For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.

Market Madness (C-64) Explanation of the Program

| Line Nos. | Explanation |
|-----------|--|
| 100-160 | Program header. |
| 170-360 | Initialize program. |
| 370-820 | Main game menu. |
| 830-970 | Buy stock routine. |
| 980-1110 | Sell stock routine. |
| 1120-1240 | Main menu for trading. |
| 1250-1820 | Trade with other players. |
| 1830-1970 | Cash stock into the bank. |
| 1980-2130 | Trade for another stock. |
| 2140-2260 | Main menu for loans. |
| 2270-2360 | Take out a loan from the bank. |
| 2370-2520 | Pay back a loan to the bank. |
| 2530-2640 | Calculate interest on a loan. |
| 2650-2690 | Routine for the Next option. |
| 2700-2980 | Display portfolio. |
| 2990-3200 | End of turn, week, and game routines. |
| 3210-3330 | Cursor routine—products rotating cursor. |
| 3340-3740 | Display instructions. |

For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.

Market Madness (IBM PC/PCjr) Explanation of the Program

| Line Nos. | Explanation |
|-----------|------------------------------------|
| 100-190 | Program header. |
| 200-290 | Initialize the program. |
| 300-320 | Main game menu. |
| 330-400 | Buy stocks routine. |
| 410-470 | Main menu for trading. |
| 480-860 | Trade with other players. |
| 870-980 | Trade for another stock. |
| 990-1080 | Cash into bank. |
| 1090-1190 | Next turn. |
| 1200-1270 | Sell stocks routine. |
| 1280-1460 | Routines for the loans section. |
| 1470-1540 | Display the portfolio. |
| 1550-1640 | End of the game. |
| 1650-1710 | Keyboard routines. |
| 1720-1780 | Display the main game menu screen. |
| 1790-1820 | Program data. |

For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.

Market Madness (TI-99/4A) Explanation of the Program

| Line Nos. | Explanation |
|-----------|---|
| 100-180 | Program header. |
| 190-290 | Initialize the program. |
| 300-310 | Main control loop for main menu screen. |
| 320-410 | Buy and sell stock. |
| 420-450 | Trading main menu. |
| 460-780 | Trade with other players. |
| 790-870 | Trade for another stock. |
| 880-960 | Cash into bank. |
| 970-990 | Loans main menu |
| 1000-1030 | Borrow from the bank. |
| 1040-1100 | Pay back loan. |
| 1110-1140 | Interest calculation. |
| 1150-1280 | End of turn, and week routines. |
| 1290-1320 | Display Portfolio screen. |
| 1330-1390 | End of the game. |
| 1400-1450 | Display main menu screen. |
| 1460 | Key input subroutine. |
| 1470 | Routine to clear part of the screen. |
| 1480 | Image format. |
| 1490 | Game data. |
| 1500-1510 | Time delay subroutine. |

For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.

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

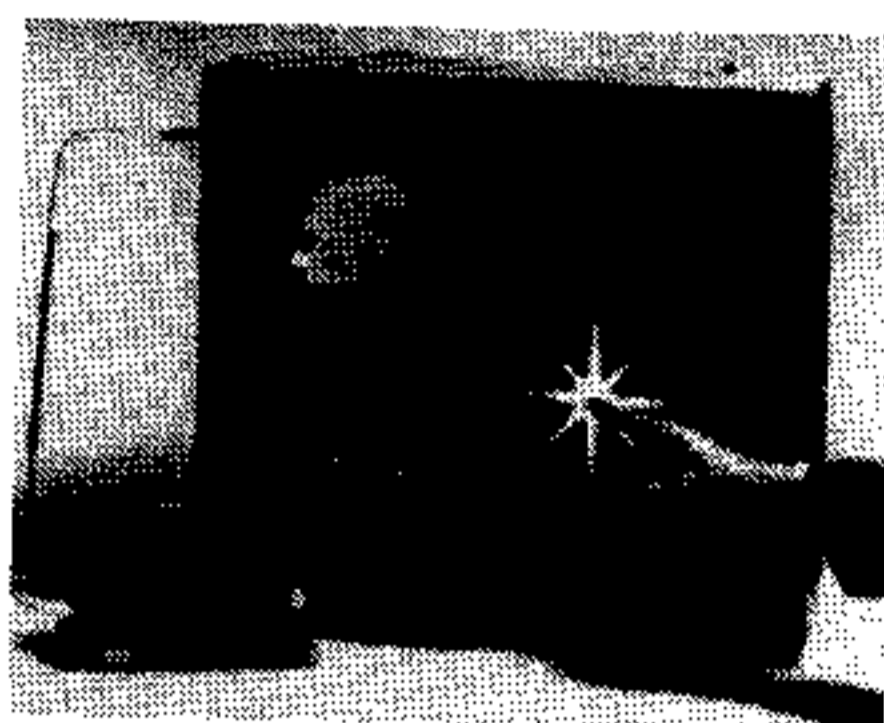


A Touch of Glass

Kit Converts Monitors to Touch Screens

Interaction Systems, Inc. has announced a digitized X-Y Touch Sensor for Sony-type CRTs and monitors. Designated the TK-2000 Series Kits, the units provide an X-Y coordinate output with 100 part resolution when a person touches the tempered glass faceplate. Output is through an RS232 serial port. The Digitized X-Y Touch Sensor is available in 12,

Interaction Systems, Inc.
24 Munroe St.
Newtonville, MA. 02160
(617) 964-5300



15, and 19 inch sizes starting at \$660.



More Utility from Your TI

Three New Assembly Language Programs for the TI-99/4A

Three high-speed assembly language programs were released by StarSoft for the TI-99/4A. Microkey provides ten user-defined function keys in TI BASIC or Extended BASIC. Each of ten control keys may be assigned a 28-character string consisting of a BASIC command that will be automatically entered when the corresponding key is pressed. Microkey's list price is \$19.95. Nibbler is a fast sector-by-sector disk copier and formatter. It

StarSoft
601 Alleghany St.
Blacksburg, VA. 24060

contains options to copy only certain sectors, to write to a different sector number than the one read from, and to format the destination disk. Nibbler's list price is also \$19.95. Unprotector allows the user to "unprotect" protected TI Extended BASIC programs while in memory. This allows users to backup and edit protected programs, and to transfer protected programs from tape to disk. Unprotector lists for \$14.95.



Increasing Your Power

Expanding Memory, I/O, & Disk Storage on TI Systems

Myarc, Inc. has released their newly developed MPES/50 systems. The MPES/50 System is a mini peripheral expansion system for the TI-99/4A, with 32K bytes of expansion memory, an RS232 serial port and parallel I/O port, a floppy disk controller, and a double-density, single-sided disk drive. It retails for \$595.

A two drive model, the MPES/50-2, comes equipped

Myarc, Inc.
P.O. Box 140
Basking Ridge, NJ 07920
(201) 766-1700

with two double-density, single-sided disk drives. It retails for \$785. A double-sided drive option is available for an additional \$50. For cassette memory storage systems, the MPES/50-RPM comes with all of the above except the disk controller and drive. This system can be upgraded later to full MPES/50 capability. It retails for \$299.



Extensions for the Home Accountant

Popular Financial Software Comes to Mac & Junior

The best-selling program The Home Accountant by Continental Software is now available for the IBM PCjr and the Apple Macintosh. On the PCjr, the program tracks up to five checkbooks, all cash and credit card transactions, and up to 100 budget categories. It will also print checks and a variety of reports. The Home Accountant jr costs \$74.95. On the Macintosh, the program will track any number of check-

Continental Software
11223 South Hindry Ave.
Los Angeles, CA. 90045
(213) 417-8031

book accounts, record 25 monthly automatic transactions, flag tax items, and enter monthly budgets for assets, credit cards, liabilities, income, and expense categories. It includes a financial calculations module which allows the user to calculate loans or determine the future value of a specific monthly investment. The Home Accountant for the Macintosh costs \$99.95.



Tiny Turtles

Cassette-Based LOGO for Bare-Bones 99/4A

Microcomputers Software has announced TINY LOGO on cassette for the TI-99/4A. No extra memory is required. Like bigger versions of LOGO, TINY LOGO uses turtle graphics to teach principles of programming. The software

Microcomputers Software
34 Maple Ave.
Armonk, NY. 10504
(914) 273-6480

package comes with a 32-page instruction booklet featuring samples of simple and recursive procedures and a summary of TINY LOGO terminology. It is priced at \$19.95. Versions for other home computers will soon be released.



A Guide *IIc* You Through

A Book for the Portable Apple

Bantam Books has published *The Apple IIc Book* by Bill O'Brien. Written for new Apple IIc computer buyers and experienced Apple users, the book answers users' questions

Bantam Books
666 Fifth Ave.
New York, NY. 10103
(212) 765-6500

about compatibility, configuring the system, and adding peripherals. It also includes information on DOS 3.3 and the new ProDOS. The paperback book is priced at \$12.95.



Muppet-Friendly Keyboards for Kids

Apple/Commodore Accessory Simulates School Desk

Children age three and up can learn letters, numbers, and colors with the assistance of the Muppets, featured on Muppet Learning Keys, Kids' Computer Keyboard. Developed by Koala Technologies Corp., the keyboard simulates the contents of a child's school desk to help children learn basic skills. The 14" x 15" three-pound keyboard connects to a computer display screen through the paddle port on the Apple IIe or IIc, or the joystick port on the Commodore 64. Each section of the desk—penmanship slate, paint set, arithmetic exercise book, etc.—can be activated by the touch of a child's finger. Miss Piggy,

Koala Technologies Corp.
3100 Patrick Henry Drive
Santa Clara, CA. 95052-8100
(408) 986-8866



Gonzo, Fozzy Bear, and Kermit help provide instruction. Muppet Learning Keys is priced at \$79.95.



Software Explosion from Great Lakes

Games and Graphing for TI Users

Super Bargraphs, Lunar Cavern, and Funhaus are three new programs for the TI-99/4A released by Great Lakes Software. Super Bargraphs features graphing of up to 15 items at a time with labels, automatic scale adjustments, accuracy to one pixel, printout capability,

Great Lakes Software
P.O. Box 241
Howell, MI 48843

and more. It is available in BASIC or Extended BASIC versions, on cassette for \$14.95 and on disk for \$16.95. Lunar Cavern and Funhaus are arcade-style games for use with Extended BASIC, and are the same prices as Super Bargraphs.



Take Stock of Your Investments

Portfolio Management Software for Home Machines

Basic Byte, Inc. has introduced a portfolio management system of three volumes that work independently. Stock Management Vol. I is for the investor's personal stock portfolio. It allows the instant update of current value of up to 100 individual stocks, calculates long and short term capital gains and losses, and records dividends. Stock Management Vol. I for the Commodore 64, VIC-20, and Atari retails for \$39.95. The IBM PC version retails for \$59.95, and an Apple version is forthcoming.

Options Management Vol. II is for the speculative

Basic Byte, Inc.
P.O. Box 924
Southfield, MI. 48037-0924
(313) 540-0655

options trader, and incorporates the record-keeping features of Stock Management Vol. I for use with an options portfolio. It also determines the fair market values of any stock option using the "Black-Scholes" model. It is available for the C-64 for \$39.95, and versions for the IBM PC, Apple, and Atari are in the works.

Graphic Analysis Vol. III, scheduled for fall release for the C-64, IBM PC, Apple, and Atari computers, is a graphics program designed to track stock market trends as well as an individual stock's performance.

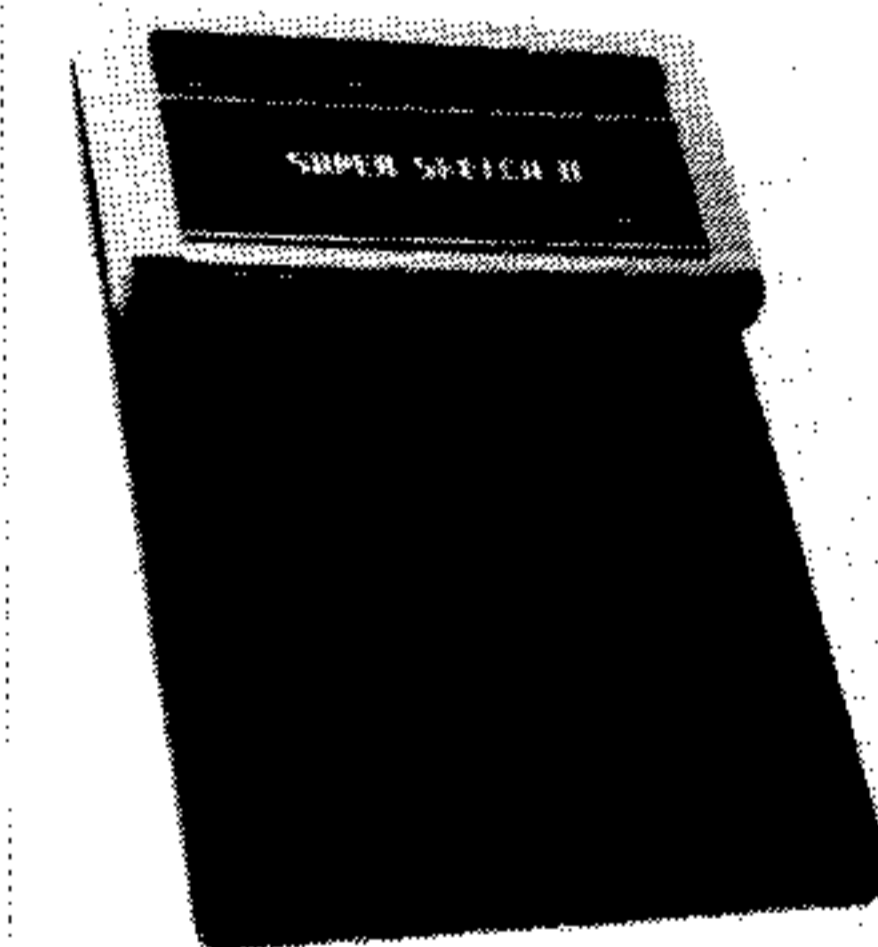


Details Sketchy On Graphic Add-On

Drawing Tablet Hooks Up To All Popular Machines

Personal Peripherals, Inc. has announced the release of Super Sketch—a graphics tablet with software cartridge—and Super Sketch II, as well as four applications software packages for Super Sketch. Super Sketch allows users to create color graphics by moving a stylus control as they would a pencil. It is compatible with the TI-99/4A and the Commodore 64 and is priced at \$59.95. Super Sketch II is physically similar to Super Sketch, but it has different styling and "a color scheme designed to blend with professional environments." It is compatible with Apple II and IBM PC and PCjr computers. Its suggested retail price is \$79.95. The applications packages, for Commodore 64 computers, include Super Music Box

Personal Peripherals, Inc.
930 N. Beltline, Suite 120
Irving, TX. 75061
(214) 790-1440



(\$19.95) for composing and performing music, Business Presentor (\$39.95) for business-related graphics, Master Home Planner (\$49.95) for creating home and commercial floor plans, and Printer Utility (\$29.95) which allows print-out capability for video graphics created with Super Sketch.



HOME COMPUTER™

product news

Junior's Desk Gets Windows

Integrated Productivity & Adventure Packs for PCjr

The jr. Series (tm), a line of software products for the PCjr, has been released by Oakwood Publishing. The series currently includes jr. DESK, an integrated package with window displays, for personal finances and small business operations; jr. FILE, a home file management system; jr. QUEST, a fantasy

adventure game that teaches players how to use logic to enhance creative problem-solving skills; and jr. WORLD TRIATHLON, an adventure tutorial with an Olympics theme that tests players' skills in typing, spelling, and memory recall. Suggested retail price for jr. DESK is \$99.95, and for the other programs, \$49.95.

Oakwood Publishing
P.O. Box 3934
Gardena, CA 90247
(213) 217-1323



Learning With Robots

Construction-Kit Game Teaches Digital Electronics

A robot construction kit that teaches the basics of digital electronics, and develops logic and hypothesis formulation skills is the basis of Robot Odyssey I by The Learning Company. Players begin by falling into a futuristic underground city inhabited by robots. The object is to escape by

designing the circuitry and chips for robots which will help them get through various levels of civilization. The program, aimed at teenagers and young adults, comes with tutorials to assist players. It will be available for \$49.95 for the Apple II family of computers.

The Learning Company
545 Middlefield Rd., Suite 170
Menlo Park, CA. 94025
(415) 328-5410



That's Entertainment!

Imagic Launches New Software Series

Imagic has launched four new entertainment product lines—Fun with Experts, Educational Simulations, Living Literature, and Time Travelers. The Fun with Experts series kicks off with Crime and Punishment. Players assume the role of judge in sentencing offenders for crimes. The Educational Simulations series that began with Microsurgeon continues with Injured Engine, where the player is provided with the technical information

and tools required to tune a car engine. The first Living Literature product brings the recently published Damiano trilogy into the realm of an interactive graphics adventure game. The Time Travelers series opens with Another Bow, a Sherlock Holmes mystery set in post-Victorian England, and The Time Machine, based on H.G. Wells' science fiction story. All five of these programs will debut this fall at \$34.95.

Imagic
981 University Ave.
Los Gatos, CA. 95030
(408) 399-2200



More Storage for Junior

PCjr Gets Dual-Disk Controller

Legacy Technologies, Ltd. has announced a two drive controller for the IBM PCjr. Legacy's new floppy disk controller provides access to two disks through Junior's operating system.

Legacy Technologies, Ltd.
4817 North 56th St.
Lincoln, NE. 68504
1-800-228-7257

The cabling provided can power one disk drive inside junior, and extend under the cover to control a second drive. The second drive can be house inside a Legacy II or positioned alongside.

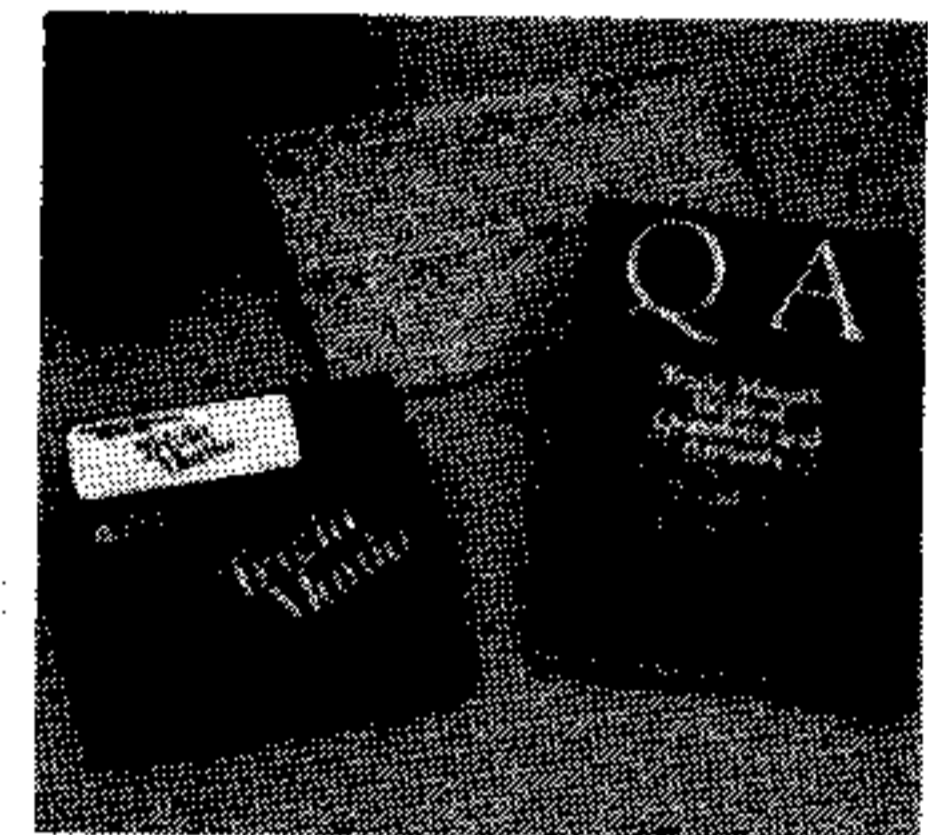


For Those With All the Answers

Trivia Games for Home Computers

Trivia Mania, a new game by Professional Software, Inc., brings the current trivia craze to Commodore 64, Apple II family, and IBM PC and PCjr users. The game consists of approximately 3,500 questions on diskette and in printed form, in three levels of difficulty and in seven categories: Science and Technology, Geography, History, Sports, Films and Entertainment, Famous People, and Nature and Animals. Trivia Mania retails for \$39.95. Professional Software is developing a series of add-on

Professional Software, Inc.
51 Fremont St.
Needham, MA. 02194
(617) 444-5224



diskette packages for the game entitled Super Sports (tm), Movie Madness (tm), What's in a Word (tm), and Educational Learning Diskettes (tm).

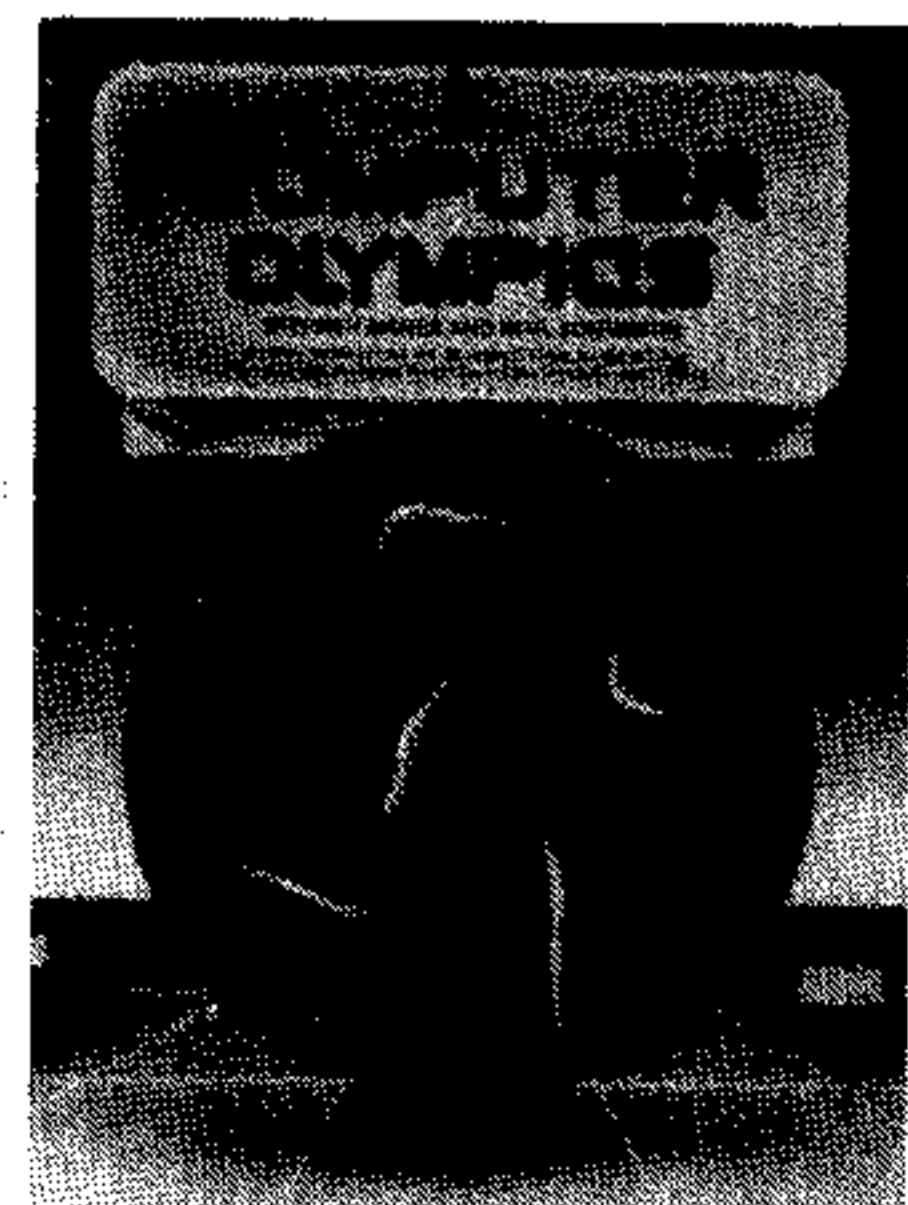


Booking An Olympic Event

Game Activity Book Teaches The BASICS

Released just in time for the Olympic Games, a new book by Stephen Manes and Paul Somerson provides ready-to-key-in listings of Olympic games and sideline activities. Computer Olympics, published by Scholastic, Inc., can be used with TI-99/4A, IBM PC and PCjr, Commodore 64 and VIC-20, and Apple II family computers. Programs include Toss the Javelin, Track and Field Record Book, and Bronze Medal Diver. The paperback book retails for \$4.95.

Scholastic Inc.
730 Broadway
New York, NY. 10003
(212) 505-3546



Cartoons and Kid Shows Go Floppy

Two Big-Name Titles from First Star

First Star Software's licensing agreements will bring MAD Magazine's cartoon strip *Spy vs. Spy* and the television classroom, *Romper Room* to the home computer this October. *Spy vs. Spy* will employ animated graphics, and, like the *Romper Room* programs, will be released initially for the Commodore

First Star Software
18 East 41st St.
New York, NY. 10017
(212) 532-4666

64 and Apple II family computers. The first program of the *ROMPER ROOM Little Learner* (tm) Series will be *Romper Room's I Love My Alphabet*. It will feature animations demonstrating different action words. The games will retail for \$34.95 for the C-64 versions, and \$39.95 for the Apple versions.



Tapes do the Teaching

Hands-On Learning Exercises for PCjr

FlipTrack Learning Systems has published a tutorial on *How to Operate the PCjr*. The self-paced tutorial is designed to teach hands-on computer operation through two audio cassettes. The first cassette guides users through start-up procedures; keyboard familiarization; BASIC programming; and the PCjr's color, sound, graphics,

FlipTrack Learning Systems
999 Main, Suite 200
Glen Ellyn, IL. 60137
(312) 790-1117

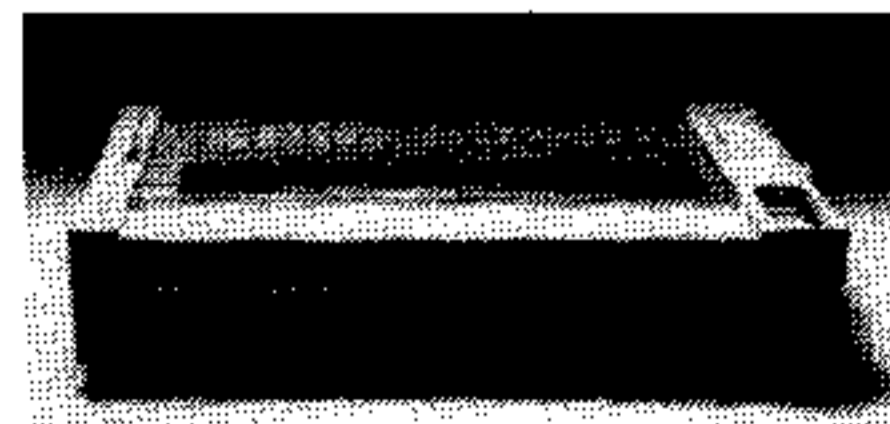
and mathematical capabilities. The second cassette details managing disk storage and files with DOS; using tree-structured directories; copying, renaming, and erasing files; and batch processing. An indexed quick reference guide accompanies the cassettes. *How to Operate the PCjr* is priced at \$39.95.



A Bigger Image for Apples

Wide-Carriage Printer Makes Debut

A wide carriage model of the Imagewriter dot matrix printer is now available from Apple Computer. The Wide Carriage Imagewriter is suited for producing documents that require wide paper such as spreadsheets, forecasting models, budgets, and data processing reports. It accommodates a range of paper sizes from three to 15 inches wide, and is compatible with Apple II and Apple III computers. Like the standard size model, the Wide Carriage Image-



writer prints in a 7 x 9 dot matrix at a rate of up to 120 characters per second. It also features eight character fonts, and provides variable resolution, pitch, and line spacing. This new Apple printer retails for \$749.

Apple Computer, Inc.
20525 Mariani Ave.
Cupertino, CA. 95014
(408) 996-1010



Make It Easy On Yourself

Word Processing & Utilities for the C-64

Educomp has announced three new low-priced programs for the Commodore 64—a word processor and two utilities. The *Quickwriter II* has over 60 editing commands, and a printer routine compatible with every printer interface and printer combination.

Power Plus adds over 40 new commands to the C-64, making it easier to send disk commands, write and debug

Educomp
2139 Newcastle Ave.
Cardiff, CA. 92007
(619) 942-3838

programs, and write code in machine and assembly languages. *Menu-driven Disk Pac* can check all sectors of a diskette for problems without losing the diskette's data. It can also unscratch data files that have been accidentally scratched or erased. Both *Quickwriter II* and *Power Plus* are priced at \$19.95, and the *Disk Pac* is priced at \$14.95.



PC Software in the Public Domain

New Directory Shows Where and How To Find It

A new directory from PC Software Interest Group lists hundreds of public domain and user-supported programs available for the IBM PC and compatible computers. The *Directory of Public Domain Software for the IBM Personal Computer* is composed of programs written by people who have chosen not to market their

PC Software Interest Group
1556 Halford Ave. Suite 130S
Santa Clara, CA. 95051
(408) 730-9291

software. The directory catalogs what programs are available and where to get them. It lists word processing, communications, data base, DOS and BASIC utilities, games with color graphics, Pascal, C and assembly language programs, and more. The directory retails for \$4.95.



Balancing a Checkbook Made Easy

Rocketman to the Rescue

A new program for the TI-99/4A has been released by *Rocketman* to assist people who hate to balance checkbooks. *Rocketman Jr.* (cassette version) and *Rocketman Sr.* (diskette version) systematically input all the information needed for reconciliation. The program provides a single screen read-out, and com-

Rocketman
4104 San Pablo Dam Rd.
El Sobrante, CA. 94893
(415) 222-1626

pares all data. Entries can be corrected until DIFFERENCE=0, indicating balance. It includes graphics and a built-in calculator, which checks the additions and subtractions of all entries in the checkbook register. *Rocketman Jr.* retails for \$24.95 and *Rocketman Sr.* retails for \$39.95.





This article continues an ongoing tutorial on the *Multiplan* software package. Newcomers to the program may wish to consult back issues of *99'er Home Computer Magazine* for previous articles in this series.

Many people feel they must justify the cost of adding a new peripheral to their home computer system. They think it is not enough to simply want to improve their machine's efficiency or increase their own computer literacy. If you are one of those who is seeking "cost-effectiveness rationalization," you will be interested in finding out how *Multiplan* can help you calculate the cost effectiveness of adding new equipment to your system. As an example, we will determine whether the cost of adding a printer to your system is justified in terms of the time and money it saves. The ideas presented here can be used for any major purchase decision.

There are two types of costs associated with equipment: initial and recurring. The purchase price of the printer is an initial (one-time) cost, while the costs of maintenance and supplies will recur over the years. In addition, recurring costs generally increase each year at some rate, such as the inflation rate.

Even though the equipment will last for several years, we must make our buying decision today. So, we need to express the projected costs and savings associated with the equipment in today's dollars. Thus, the decision to buy should also take into consideration the so-called "opportunity rate." This rate is also called the cost of money, the discount rate, or the interest rate.

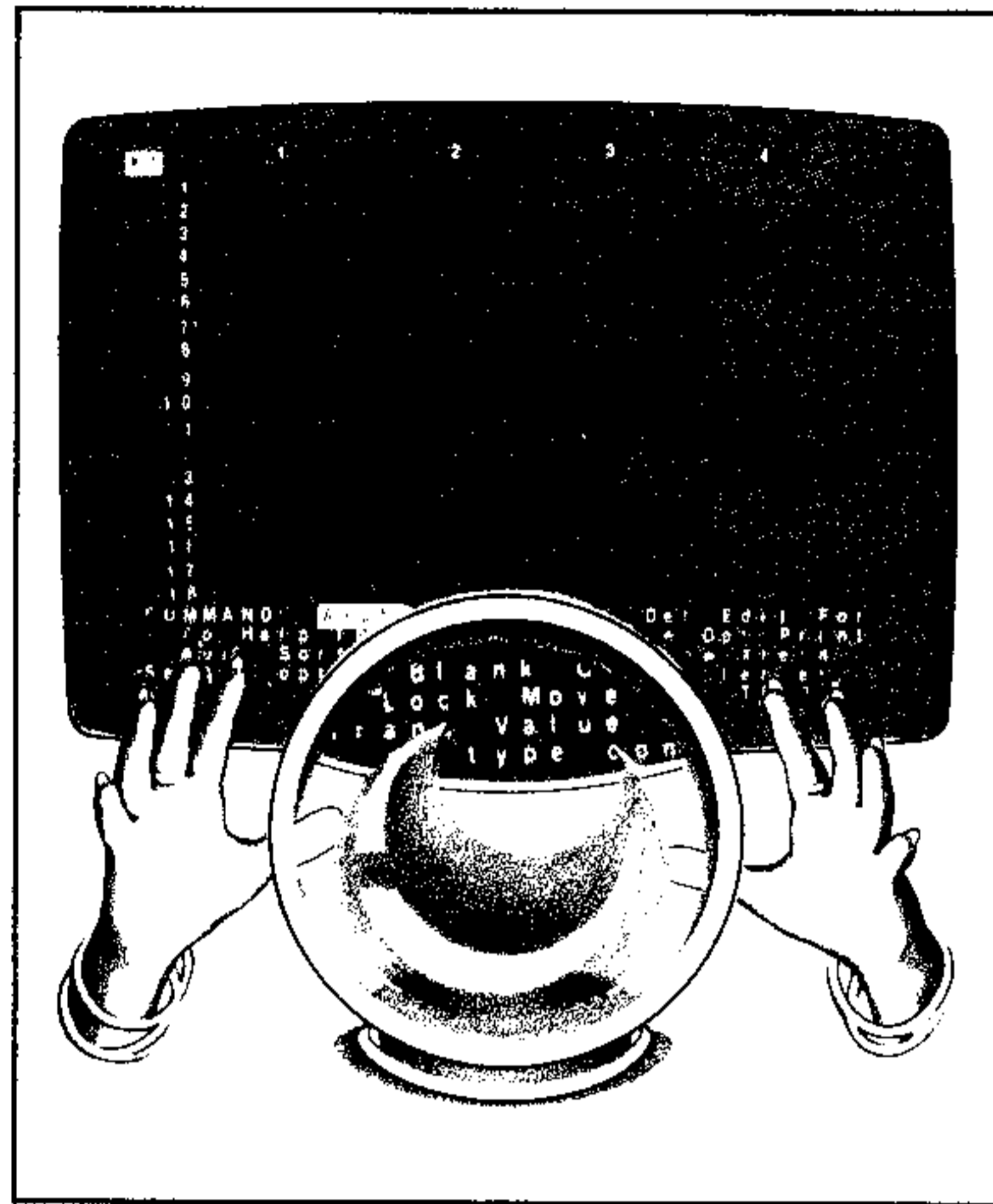
This is where Net Present Value (NPV) comes in. In mathematical terms, the total cost of something which will be purchased over n years with a discount rate of k in effect will be (in today's dollars):

$$\sum_{i=1}^n \frac{\text{cost during year } i}{(1+k)^i}$$

Multiplan's NPV function performs this calculation. We supply it with a list of costs for each year and the discount rate, and *Multiplan* returns the value in today's dollars. Thus, we don't have to be accounting wizards to make good buying decisions. Figure 4 shows the Net Present Value model to which the following discussion refers.

Factors in Equipment Cost

The initial costs for adding a printer are the cost of the equipment and the cost of the software to use the printer. Our hypothetical user already has *Multiplan*, which probably means that he or she already has whatever else the system requires to run *Multiplan* (e.g., memory expansion). Therefore, the only hardware needed is probably the printer, a cable, and an RS232



Interface Adapter. *Multiplan* can output to the printer, so no new software must be added. Since these one-time costs all occur at the beginning of the first year of use, they are already in today's dollars. So no fancy calculations based on inflation rates are needed at this point.

Two recurring costs of owning a printer are maintenance and supplies. I have set the first year's maintenance at 10% of the cost of the printer itself because my experience with printers shows that this is a reasonable figure. Some users may purchase a maintenance agreement with their printers, which would cause this figure to vary. The cost of supplies depends on exactly how the printer will be used. You should include paper and ribbons in this figure. If you are

analyzing the purchase of a daisy wheel printer, be sure to include the cost of the wheels at this point, since they wear out over time and may need to be replaced.

This model assumes that the printer's life is three years. The recurring costs in the second and third years of life have had a projected inflation rate applied to them. The model assumes a fixed inflation rate, but you can easily get fancier if you like. The figure for each year is based on the year before, so you can use two different inflation rates to arrive at year 2 and year 3.

Once the costs have been filled in, you'll want to calculate the NPV of each recurring item. I have based this calculation on the interest rate given at the top of the model. Because this rate can vary, I have set up the model so this can be changed easily. The NPV of each recurring cost is figured on this rate and on the three years of cost that are in the same row. Thus, the figures in the NPV column of the model represent costs in today's dollars.

Some users may want to include more costs than those shown. For example, if you are adding memory expansion as well as a printer, then you should probably come up with a figure for electricity usage. This would be a recurring cost, and the amounts projected for the second and third years might be based on a rate different from the inflation rate. If you are adding a large printer, you might also want to buy a word processor to make maximum use of your new printer.

Time and Accuracy Savings

Now for the other side of the analysis, the savings you expect to gain from adding a printer. If you will be using the printer to produce something which you are already producing by hand or on a typewriter, then these savings are fairly easy to measure. I have used two types of savings: time and increased accuracy.

Because the value of time saved is a fairly complicated calculation, I have used a supporting worksheet to calculate it. Figure 3 shows this supporting worksheet. If you currently type the materials that will be printed in the future, then you can use this supporting sheet

to put a value on your time saved by using a printer. You must first estimate the volume of typing you usually do. In the example, this is expressed in weeks: 4 documents of 300 words per week. The average word used is 7 characters long. You must also specify your typing speed and the speed of the printer you are considering; in the example these are 40 words per minute (WPM) and 150 characters per second (CPS), respectively. To figure the hours per week currently spent typing, use the formula:

$$\frac{\text{words/document} \times \text{documents}}{\text{speed (wpm)} \div 60 \text{ min./hr.}}$$

Figure 1—Formulae for Supporting Worksheet

| 1 | 2 | 3 |
|----|---------------------------------------|---|
| 11 | $R[-4]C^*R[-3]C/R[-7]C/60$ | |
| 12 | $R[-5]C^*R[-4]C/R[-7]C^*R[-6]C/60/60$ | |
| 13 | $R[-2]C - R[-1]C$ | |
| 15 | $R[-2]C^*52^*R[-6]C$ | |

MULTIPLAN ON THE C-64

A review
by Patricia Swift



When asked to try out *Multiplan* on the Commodore 64, I agreed eagerly even though I had never used that computer before. I have run *Multiplan* on several other microcomputers and have always found the program to be about the same as far as the human interface goes. *Multiplan* on the C-64 is no exception.

You need at least one disk drive to run this program on the C-64. Not being very familiar with this computer, I had more trouble correctly hooking up the console, monitor, and disk drive than I did using *Multiplan*. In other words, HesWare has done a good job of implementing *Multiplan* on the C-64. The manual is first-rate. It's the standard Microsoft *Multiplan* manual with tutorial and reference sections, but edited for the C-64 with information about special keys and disk handling. The "Getting Started" and "Operating Instructions" sections are extremely helpful and should definitely be read before you try to use the program.

Slow Disk Access

The start-up sequence for *Multiplan* is clumsy, although seasoned C-64 users would probably think nothing of it. You have to type LOAD "MP", 8 and then RUN to get going. After that, you have to wait for over two minutes until you can use the program. This long load time might scare new users (it worried me quite a bit), but you just have to be very patient. This slow disk performance is a characteristic of the C-64 and should not be blamed on *Multiplan*—the disk drive is connected to the console via a serial interface. Serial interfaces transmit data one bit at a time, which explains why a large program like *Multiplan* takes a long time to load into memory. Most other computers I have tried (including the TI-99/4A) use a parallel interface for the disk; parallel interfaces transmit a byte (8 bits) at a time.

Tortoise Speed

Once the program is loaded, it runs very quickly, except when the disk is being accessed. Unfortunately, that seems to happen more often than with other versions of *Multiplan*. Many of the commands require a disk read. You'd expect the HELP and TRANSFER commands to access the disk, but BLANK, FORMAT,

WINDOW, and others do it too. Although each disk access takes only a few seconds, the cumulative effect can really slow you down.

Multiplan for the C-64 comes on a write-protected disk that cannot be copied. This means that you must use a separate disk for saving worksheets—making it necessary to change disks if you are running with a single disk drive (as I was). The manual gives a step-by-step procedure for making working disks, which makes using the program quite a bit easier. This procedure copies onto a separate disk the two files that *Multiplan* needs at runtime, and even formats the working disk for you. You still have to load *Multiplan* from the original disk, but then you can switch to a working disk and leave it inserted while you work. A word of warning: This procedure was explained on a separate page that was stuck into the disk holder at the back of the manual. It's easy to miss, but definitely worth looking for.

HesWare will sell you a backup copy of the *Multiplan* disk for \$10. This sounds like a good way to protect your software investment, especially since floppy disks get quite warm (even hot) after being in the Commodore's disk drive for a while.

While running the program, I had the most trouble with the [SHIFT] key. Seasoned Commodore users already know that the [SHIFT LOCK] key gives you uppercase everything, not just capital letters. Even the [RETURN] key works differently with the [SHIFT LOCK] down. This setup really had me confused for a while—I recommend that you leave the [SHIFT LOCK] up (disengaged) when using *Multiplan*.

The display shows 40 columns with a solid border around the whole worksheet. This resulted in fairly small characters on my 10" monitor, which made it hard to read the screen. However, the border is a good idea because it prevents a badly-aligned screen from chopping off characters. Thus, if you use a TV set, you won't suffer with characters that are fuzzy around the edges.

The Key Advantage

The special keys on the C-64 are used to make *Multiplan* easier to operate. For example, the [RUN/STOP] key means cancel (it's sort of a panic button). It is extremely handy to have this often-used command as a single key, although you can still use the traditional [CONTROL] C for cancel.

Name: Microsoft/Multiplan
Program Type: Electronic Worksheet
Distributor: HesWare
150 North Hill Drive
Brisbane, CA 94005
(415) 468-4111
Price: \$99.95

System Requirements: Disk Drive

Poor Fair Good Excellent

Performance: ██████████
Ease of Use: ██████████
Documentation: ██████████

My other favorites are [F1] for tab and [F3] for delete. A small overlay is provided for the function keys so you don't have to memorize their meanings. There are still a few key sequences which must be learned or looked-up (for example, scroll down is [CONTROL] R [DOWN ARROW]).

When printing with *Multiplan* on the C-64, you can skip the usual preliminary steps of opening channels between peripherals to access your printer. Once you boot up the system and access a file, you can get printouts as easy as typing P. The P command automatically puts you in print mode, and displays the following:

PRINT: Printer File Margins Options

These are the four printer subcommands available with *Multiplan*. The first subcommand, Printer, allows you to print an entire worksheet, under the limits of preset margins. The second subcommand, File, allows you to store printed output on disk rather than send it directly to the printer. This provides the option of adding to or changing data before getting a printout. The third subcommand, Margins, lets you set margins, specify the number of characters per line, and set page lengths. The last subcommand, Options, lets you print specific areas within your worksheet.

The major differences in running *Multiplan* on the Commodore 64—as you can see—all relate to the machine itself. All of *Multiplan's* commands and functions are there, and the syntax is the same as on other machines. This means that Commodore 64 users can use the *Multiplan* models described in my *Multiplan* Medium series, and elsewhere. Just take some time to become familiar with the Commodore's special keys, and don't forget to make yourself a working disk.

HCM

Figure 2

Building the Model

1. First build the supporting worksheet. Set the Format Width for column 1 to a width of 25 characters. Then fill in the labels in column 1 (see Figure 3).
2. Fill in the speeds and their accompanying labels. Note that the format of the cell containing the hourly rate (R9C2) should be changed to \$ via the **FORMAT CELLS** command.
3. Now fill in the calculations (R11C2:R15C2) and their accompanying labels in column 3.
4. Use the **NAME** command to assign the name **TIMEVAL** to the result in R15C2.
5. Store the supporting sheet on disk via the **TRANSFER RENAME** command. Save it on a file named **TIMEWK**.
6. Now get ready to construct the main model. Use the **TRANSFER CLEAR** command to clear the screen.
7. Set the Format Width for column 1 to a width of 26 characters. Then fill in the labels in column 1 (see Figure 4).
8. Change the default format for the worksheet to Fixed with 2 decimal places via the **FORMAT DEFAULT CELLS** command.
9. Change the format of the two cells for rates (R4C2:R5C2) to percentage (%) via the **FORMAT CELLS** command. Then fill in the projected inflation rate and interest rate in those cells. Name those cells **INFL** and **INTEREST**, respectively.
10. Fill in the headings in row 7.
11. Fill in the costs of the printer, cable, and RS232 card in column 2. **SUM** these into R12C2. Then set the NPV in R12C5 to this same value by using = and picking up the total just calculated.
12. Fill in the cost of software at R14C2 and put it into R14C5 in the same way.
13. Put the formula for the first year's maintenance in R16C2. If you'll have a maintenance contract, you may want to use an actual figure here instead.
14. Fill in the formula for R16C3, then Copy it 1 cell to the right. Fill in the NPV formula at R16C5.
15. Fill in the first year's cost of supplies at R18C2. Then copy the formula for years 2 and 3 from R16C3 to R18C3:R18C4. Copy the NPV formula from R16C5 to R18C5.
16. Use the external **COPY** command to put the time value from the supporting worksheet into R22C2. Copy from sheet **TIMEWK**; name it **TIMEVAL**. Be sure to set the Link option to Yes so that if the supporting sheet is changed, then the main sheet will also be changed.
17. Fill in the value of increased accuracy at R23C2. Then **SUM** the two savings values into R24C2. Compute the savings for the second and third years by copying the formulas from cell R16C3 to cells R24C3:R24C4. Copy the NPV formula from R16C5 to R24C5.
18. Compute the overall NPV of adding the printer by entering the formula shown into R26C5.
19. Store the main worksheet on disk via the **TRANSFER RENAME** command. Save it on a file named **NPV**, or any other name of your choice.

The calculation for the hours which the printer would take to accomplish the same task is more complicated only because printer speeds are usually expressed in characters per second:

$$\frac{\text{words/doc.} \times \text{documents} \times \text{avg. word length}}{\text{speed (cps)} \div 3600 \text{ sec/hr.}}$$

The number of hours saved per week is just the difference between these two figures. (This printer example assumes the documents are already stored in a word processor.) Finally, you must assign a value to your time. Then the value of the time saved per year by using a printer will be:

$$\text{time saved/wk.} \times 52 \text{ wks./yr.} \times \text{your hourly value}$$

This supporting worksheet can supply the bottom-line time value to the main worksheet automatically. To do this, you must construct the supporting sheet

Figure 3—Value of Time Saved

| 1 | 2 | 3 |
|------------------------------|-----------|----------|
| 1 Value of Time Saved | | |
| 2 Supporting Worksheet | | |
| 3 | | |
| 4 Typing Speed | 40 | WPM |
| 5 Printer Speed | 150 | CPS |
| 6 Average Word Length | 7 | CHARS |
| 7 Average Document Length | 300 | WORDS |
| 8 Avg. Documents Per Week | 4 | |
| 9 Value of Your Time | \$10.00 | PER HR |
| 10 | | |
| 11 Current Time Spent Typing | 0.5 | HRS/WK |
| 12 Printer Time | 0.0155556 | HRS/WK |
| 13 Time Saved | 0.4844444 | HRS/WK |
| 14 | | |
| 15 Value of Time Saved | \$251.91 | Per Year |

first, name the cell containing the final result, and save the supporting sheet on disk. Then when you construct the main (NPV) model, you can ask *Multiplan* to use this "external" value by specifying the filename of the worksheet and the name of the cell. By specifying at this time that the worksheets are to be linked, you can cause changes in the supporting sheet to be automatically reflected on the main worksheet.

Let's return to our discussion of the main model. The other savings shown is in increased accuracy. This is more difficult to measure, as it can arise from many factors. For example, if your results are used for billings, then mistakes can cost you money. You may already have an idea of how much mistakes like this have cost you in the past.

You may be able to come up with other savings for your particular situation. Just remember that these will generally be recurring savings, and that you can use supporting sheets to work out the amounts.

In the model, the total savings are extended to the second and third years in the same ways as the costs, and the NPV is figured on the three yearly totals.

To decide whether buying a printer would be cost effective, you must see whether the NPV of the savings is more than the NPV of the costs. The final cell of the main model is calculated as savings minus all the costs. The sample model comes out to a positive figure here, meaning that this particular printer is cost effective for this situation.

To decide whether buying a printer would be cost effective, you must see whether the NPV of the savings is more than the NPV of the costs. The final cell of the main model is calculated as savings minus all the costs. The sample model comes out to a positive figure here, meaning that this particular printer is cost effective for this situation.

Building and Using the Model

The steps for building this model are shown in Figure 2. Once it has been built and saved, you'll want to use it to evaluate several different situations by varying some of the rates and costs. As you do this, you'll

Figure 4—Net Present Value Model

| | 1 | 2 | 3 | 4 | 5 |
|----|----------------------------|--------|--------|--------|---------|
| 1 | Net Present Value Model | | | | |
| 2 | For Adding a Printer | | | | |
| 3 | | | | | |
| 4 | Projected Inflation Rate | 6.00% | | | |
| 5 | Interest Rate | 8.50% | | | |
| 6 | | | | | |
| 7 | | Year 1 | Year 2 | Year 3 | NPV |
| 8 | Cost of Printer: | | | | |
| 9 | Printer | 400.00 | | | |
| 10 | Cable | 35.00 | | | |
| 11 | RS232 Interface Card | 100.00 | | | |
| 12 | Initial Printer Cost | 535.00 | | | 535.00 |
| 13 | | | | | |
| 14 | Cost of Software | 0.00 | | | 0.00 |
| 15 | | | | | |
| 16 | Maintenance | 40.00 | 42.40 | 44.94 | 108.07 |
| 17 | | | | | |
| 18 | Supplies | 50.00 | 53.00 | 56.18 | 135.09 |
| 19 | | | | | |
| 20 | | | | | |
| 21 | Savings: | | | | |
| 22 | Value of Time Saved | 251.91 | | | |
| 23 | Increased Accuracy | 150.00 | | | |
| 24 | Total Savings | 401.91 | 426.03 | 451.59 | 1085.87 |
| 25 | | | | | |
| 26 | Overall NPV to Add Printer | | | | 307.71 |

probably want to see the final result each time. This result appears in row 26 in the sample model, but you can use *Multiplan's* windowing facilities to keep the result visible at all times.

To do this, put the cell pointer on the rightmost cell on the result line. Then move the cell pointer up one cell. Type in **WINDOW** and press [ENTER] or [RETURN] three more times. This will give you a second window, just one row high, at the bottom of the screen. Notice that you should not choose the **Linked** option, since you don't want the new window to move around horizontally when the larger window does. Now move your cell pointer down one row to show the result. After this, go to the main window and

experiment with the model. For example, change the interest rate and watch the effect on the overall NPV.

The techniques presented here can be used to analyze the costs of buying all sorts of equipment. Even when the purchase involves no savings, it is still useful to find out how much the projected purchase will cost you in today's dollars. The model illustrates that the initial purchase price is not the only cost you should consider; maintenance and other recurring costs should also be included in a thoughtful analysis. And *Multiplan* provides the structure for translating all of this information into an accurate prediction of cost effectiveness.

HCM

Figure 5—Formulae For NPV Model

| | 1 | 2 | 3 | 4 | 5 |
|----|---|--------------------|-----------------|-----------------|---|
| 1 | | | | | |
| • | | | | | |
| • | | | | | |
| • | | | | | |
| 11 | | | | | |
| 12 | | SUM(R[-3]C:R[-1]C) | | | RC[-3] |
| 13 | | | | | |
| 14 | | | | | RC[-3] |
| 15 | | | | | |
| 16 | | R[-7]C/10 | RC[-1]*(1+INFL) | RC[-1]*(1+INFL) | NPV(INTEREST,RC[-3]:RC[-1]) |
| 17 | | | | | |
| 18 | | | RC[-1]*(1+INFL) | RC[-1]*(1+INFL) | NPV(INTEREST,RC[-3]:RC[-1]) |
| 19 | | | | | |
| 20 | | | | | |
| 21 | | | | | |
| 22 | | [TIMEWK TIMEVAL] | | | |
| 23 | | | | | |
| 24 | | SUM(R[-2]C:R[-1]C) | RC[-1]*(1+INFL) | RC[-1]*(1+INFL) | NPV(INTEREST,RC[-3]:RC[-1]) |
| 25 | | | | | |
| 26 | | | | | R[-2]C - R[-14] C - R[-12]C - R[-10]C - R[-8]C |



Have No Fear:

Assembly Language Won't Byte: Part IV

by Peter Lottrup
and the HCM Staff

Programmers use Assembly Language to create video games and other extremely fast routines. Key into a deeper level of understanding with this super tutorial.

In this fourth and final segment of our tutorial on Assembly Language for the TI-99/4A, we will look at the two remaining Assembly Language directives available with the Mini Memory: *EQU* and *BSS*. Then we will put together everything you've learned so far and write a program.

EQU and BSS Directives

The last two directives are *EQU* and *BSS*. You can think of *EQU* as an *EQU*al sign in BASIC — equating a label with any quantity two bytes in length. For example, if you wish to use a label for Video Multiple-Byte Write (*VMBW*), you enter: `VW EQU >6028`

This directive takes up no actual program space—it merely adds another symbol to the *SYMBOL* table. See Listing 1 for a short program that demonstrates how this is used.

Because we used the *EQU* directive to equate *VW* with the address of the *VMBW* utility, we could use the *VW* in line `>7DOC` (of Listing 1) in place of the address.

A new instruction is introduced in Listing 1: *LIMI* (Load Interrupt Mask Immediate). *LIMI 2* and *LIMI 0* are used in a loop at the end of the program (lines `>7D10` through `>7D18`) so that the program can be halted using (*FCTN*) (*QUIT*). This gives you a convenient way to end your program without shutting off the machine.

The *BSS* directive is used when you wish to reserve a section of memory for text, numbers, or variables. With the *BSS* directive you can set aside an area called a *buffer*. Use *AORG* to go to the desired location for the buffer area, assign the start with a label (if you wish), then enter *BSS*, a space, and the number of bytes you want to set aside.

Programming Tips

As you write longer and longer programs using the Line-by-Line Assembler, you will tend to use more and more labels to keep track of addresses and buffer areas. Each label takes up four bytes in the *SYMBOL* table. This table starts at memory location `>7CD8`, and each label you

add pushes the end of this table toward `>7FFF`. If you start your program at `>7D00` and use more than seven labels, the *SYMBOL* table will write over the beginning of your program.

The second programming tip concerns short jumps in a program. Due to the limited space reserved for the symbol table, it is often better to use `$` to specify jumps of just a few addresses. When used in an assembly language instruction, `$` stands for the current location counter. If you wish to jump back eight addresses from the current instruction, just type: `JMP $-8`.

To figure any jump, just subtract the current location counter from the location you wish to jump to. Remember that the addresses are hexadecimal (*HEX* or base 16), but the numbers you enter in the Assembler are decimal (base 10) unless you specify otherwise. For example, if you wish to jump from address `>7D58` to address `>7D74`, subtract:

```
>7D74 (Address to jump to)
- >7D58 (Address of jump instruction)
-----
>1C (length of jump in HEX)
```

You would enter 28 (decimal) because 10 *HEX* = 16 decimal, C *HEX* = 12 decimal and 16 + 12 = 28.

Once Upon a KSCAN

One of the trickiest (but most powerful) utilities available to you with the Mini Memory is the key-scan (*KSCAN*) utility located at address `>6020`. Key-scan is your program's link to the keyboard. It allows you to do all the things a BASIC programmer does with *CALL KEY* and *INPUT* statements.

To use *KSCAN* you need to understand the functions of three memory locations:

- 1) `>8374` is like the *key-unit variable* in *CALL KEY*. A 0 here means *KSCAN* will scan the whole keyboard.
- 2) `>8375` is similar to the *return-variable* in *CALL KEY*. The ASCII code of the last key pressed is here.
- 3) `>837C` is used like the *status-variable* in *CALL KEY*.

Listing 1

```

7D00 xxxx      AORG >7D00
7D00 xxxx VW   EQU >6028
7D00 0200      LI 0,392
7D02 0188
7D04 0201      LI 1,ST
7D06 7D1A
7D08 0202      LI 2,12
7D0A 000C
7D0C 0420      BLWP @VW
7D0E 6028
7D10 0300 NN   LIM1 2
7D12 0002
7D14 0300      LIM1 0
7D16 0000
7D18 10FB
7D1A 4845 ST   JMP NN
7D1C 4C4C      TEXT 'HELLO THERE!'
7C1E 4F20
7C20 5448
7C22 4552
7C24 4521
7C26 xxxx      END

```

To do a KSCAN of the entire keyboard, simply load a 0 into >8374; then branch and link to KSCAN. If all you had to do was check the character at address >8375, it would be easy. But first you must check to see if a key was pressed at all. To do this, you must check bit number 2 of location >837C. The Compare Ones Corresponding (COC) instruction is used with a register containing a mask constant. The bit we wish to test is the only place in the mask containing a 1. In our case we need to mask off all but bit 2 with zeros, so our mask constant would be >2000.

Here's how to set up a mask. In the TMS9900, the bits are numbered from 0 (far-left bit) to 15 (far-right bit) as illustrated below:

```

mask: >2000
0 1 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0
0 1 2 3 4 5 6 7 | 8 9 10 11 12 13 14 15
bit position

```

When you use the COC instruction, all bits with zeros in the mask are ignored, but any bit containing a 1 (bit 2 above) is compared to the specified memory location or register. The status flags are set according to the

Mini-Memory Run Option

To use this option, the REF/DEF table located at the high end of the Mini Memory's RAM area must be altered. This process was covered in detail in Part III of this series (HCM, Vol. 4, No. 1), but here's a quick list of what you need to do:

- 1) Use the AORG statement to go to >701C. Here, use the DATA directive to update this location with >7E0C. This is the new First Free Address in memory because it is one location beyond our buffer area.
- 2) When you have entered >7E0C in this location, change the Last Free Address in memory at location >701E. Enter >7FE8 here to make room for our addition to the REF/DEF table.
- 3) Next use the AORG directive to go to address >7FE8. Here, use the TEXT directive to identify our new program by name. Choose any name you want, as long as it is six characters long including spaces. For example, you could enter:

```
7FE8 0000 TEXT 'HELLO '
```

With the cursor at >7FEE, use the DATA directive to enter the starting address (7D2A) of the program.

- 5) Now you type the END directive and press [ENTER] twice. You will be returned to the Mini Memory main menu.

- 6) To run your program, just choose the Run option, and enter HELLO as the name of your program.

comparison of those bits with ones. If we place our mask in register 6, move the byte to be compared to register 1, and then do a COC instruction, we can find out if a key has been pressed. Here is a section of code that would do this:

```

LI 6,>2000
MOVB @>837C,1
COC 6,1

```

If the comparison shows that the two are not equal, the next instruction could jump (JNE) back to redo the KSCAN. If they are equal (i.e., both of the registers have ones in the second bit position), then the program could proceed to get the ASCII value of the key pressed from memory location >8375.

The Program

The program in Listing 2 displays a greeting on the screen and prompts the user to type in his or her name. It then accepts up to 12 characters and responds with a greeting that uses the name. The program is meant to be run using Mini Memory's Run option. We will refer to the actual memory locations of the program in the Mini Memory as we explain exactly what the program is doing.

Before we start the program itself, we enter the text we want displayed and assign a label to each section. First the initial greeting, 'HI! WHAT IS YOUR NAME?' is typed in using a TEXT directive at >7D00 and is given the label T1. To display the blanks that prompt for the user's name, we type ('.....') (twelve underline characters) at >7D16 and label this T2. Finally, we assign the label T3 to 'HELLO,' (our last message), starting at location >7D22. This brings us to the beginning of the program at >7D2A. Note this address; you will need to enter it in the REF/DEF table after keying in the program.

Because Mini Memory's Run option clears the screen automatically, we can begin by writing our first greeting. The screen is divided into 768 character locations (24 rows by 32 columns). We select location 100 (row 4, column 5) for our starting place. In lines >7D2A through >7D38, we load this location in register 0; the address of our text (T1) in register 1; and the length of the message (22 characters) in register 2. Then we branch and link to Video Multiple-Byte Write (VMBW).

Our input routine will give the user the option to Erase mistakes with [FCTN][3]. We place the label AG (for AGain) at line >7D3A so we can branch back there if Erase is used. Lines >7D3A through >7D48 place the starting address of the underlines in register 0; T2 in register 1; and 12 (the number of underlines) in register 2. Then we branch and link to VMBW.

At line >7D4A we begin our key-scan routine. We clear location >8374 to tell the KSCAN routine to scan the whole keyboard, and clear register 4 to keep track of the number of characters that have been input. Then in line >7D50 we load register 5 with the address of our buffer area (>7E00) where we will store the name input and load register 6 with the mask for checking the status of the keyboard.

Next we place the label LP at >7D58, which is the beginning of our key-scan loop. Here we branch and link to >6020 (KSCAN), and at >7D5C we move the status byte into register 1. Then the COC 6,1 instruction checks to see if bit 2 is set (i.e., if a key has been pressed). If not, we branch back to LP. If the bit is set, we move the character from >8375 into register 1. Lines >7D6C through >7D82 do a series of tests. First, we see if input is complete by checking for the [ENTER] key (ASCII 13). If the input is complete, we jump to the END to display the final message. If the ASCII value is not

Listing 2

```

7D00  xxxx  AORG >7D00
7D00  4849  T1 TEXT 'HI! WHAT IS YOUR NAME?'
7D02  2120
7D04  5748
7D06  4154
7D08  2049
7D0A  5320
7D0C  594F
7D0E  5552
7D10  204E
7D12  414D
7D14  453F
7D16  5F5F  T2 TEXT '-----'
7D18  5F5F
7D1A  5F5F
7D1C  5F5F
7D1E  5F5F
7D20  5F5F
7D22  4845  T3 TEXT 'HELLO, '
7D24  4C4C
7D26  4F2C
7D28  2000
7D2A  0200  LI 0,100
7D2C  0064
7D2E  0201  LI 1,T1
7D30  7D00
7D32  0202  LI 2,22
7D34  0016
7D36  0420  BLWP @>6028
7D38  6028
7D3A  0200  AG LI 0,228
7D3C  00E4
7D3E  0201  LI 1,T2
7D40  7D16
7D42  0202  LI 2,12
7D44  000C
7D46  0420  BLWP @>6028
7D48  6028
7D4A  04E0  CLR @>8374
7D4C  8374
7D4E  04C4  CLR 4
7D50  0205  LI 5,>7E00
7D52  7E00
7D54  0206  LI 6,>2000
7D56  2000
7D58  0420  LP BLWP @>6028
7D5A  6020
7D5C  D060  MOV B @>837C,1
7D5E  837C
7D60  2046  COC 6,1
7D62  16FA  JNE LP
7D64  04E0  CLR @>837C
7D66  837C
7D68  C060  MOV @>8375,1
7D6A  8375
7D6C  0281  CI 1,13
7D6E  000D
7D70  1318  JEQ ED
7D72  0281  CI 1,7
7D74  0007
7D76  13E1  JEQ AG
7D78  0281  CI 1,32
7D7A  0020
7D7C  11ED  JLT LP
7D7E  0281  CI 1,90
7D80  005A
7D82  15EA  JGT LP
7D84  06C1  SWPB 1
7D86  DD41  MOV B 1,*5+
7D88  0420  BLWP @>6024
7D8A  6024
7D8C  0580  INC 0
7D8E  0584  INC 4
7D90  0284  CI 4,12
7D92  000C
7D94  11E1  JLT LP
7D96  0600  DEC 0
7D98  0204  LI 4,12
7D9A  000C
7D9C  0205  LI 5,>7E0B
7D9E  7E0B
7DA0  10DB  JMP LP
7DA2  0284  ED CI 4,0
7DA4  0000
7DA6  13D8  JEQ LP
7DA8  0200  LI 0,356
7DAA  0164
7DAC  0201  LI 1,T3
7DAE  7D22  LI 2,7
7DB0  0202  LI 2,7
7DB2  0007
7DB4  0420  BLWP @>6028
7DB6  6028
7DB8  0200  LI 0,363
7DBA  016B
7DBC  0201  LI 1,>7E00
7DBE  7E00
7DC0  C084  MOV 4,2
7DC2  0420  BLWP @>6028
7DC4  6028
7DC6  0300  LIM 2
7DC8  0002
7DCA  0300  LIM 0
7DCC  0000
7DCE  10FB  JMP $-8
      END

```

13, we go on to see if it is ASCII 7 (the code for [FCTN][3], which is the Erase option). If it is 7, we jump back to AG, where the underlines will be displayed and the KSCAN can begin anew. Finally, in lines >7D78 through >7D82 we check to see if the character has an ASCII value of at least 32 (a blank) and no more than 90 (capital Z). If it is outside of this range, we jump back to LP.

Putting the Moves on

As each letter is accepted, lines >7D84 through >7D8A save it in the buffer area beginning at >7E00, and print it on the screen at the underline characters. The SWPB 1 in line >7D84 moves the ASCII value of the character to the leftmost (most significant) byte in register 1; then line >7D86 moves this byte to the buffer that has its address in register 5. After each character is moved to the address pointed to by register 5, the register is automatically incremented to point at the next byte of the buffer.

After the character is placed in the buffer area, lines >7D88 and >7D8A immediately branch to Video Single-Byte Write. This echoes the keypress by putting the most recent character on the screen. Register 0 contains the starting address of the character's screen position, and register 1 contains its ASCII code.

Lines >7D8C through >7D94 INCRement register 0 (the address of the screen) and register 4 (the length of the name input), then check the status of the input. If the maximum of 12 characters has not been reached and the [ENTER] key has not been pressed, the program simply jumps back to LP to see what the next input will be. If 12 characters have been input and the [ENTER] key has not been pressed, the computer will go to lines >7D96 through >7DA0. Lines >7D98 through >7D9E prevent the user from entering too many characters. Input will stop at the 12th one, and the program will accept any new input as the 12th character. If [ENTER] has been pressed, the program will jump to the ED label (see line >7D70). If the user chose the Erase option, the program would restart the input (line >7D76).

Now we come to the final section of the program, which begins at the ED label (line >7DA2) and prints the word HELLO followed by the name accepted from the keyboard. This section is reached only if the [ENTER] key was pressed and detected (line >7D6C). Lines >7DA2 through >7DA6 make sure that at least one character has been accepted by comparing register 4 to zero. If register 4 equals zero, no characters have been entered, so the computer will branch back to LP and scan the keyboard for input. If at least one character has been accepted, then lines >7DA8 through >7DB4 display the beginning of the message, starting at screen position 356 (row 11, column 5). Lines >7DB8 through >7DC4 display the user's name that we saved in the buffer at >7E00. Because we have the number of characters in the name stored in register 4, we merely transfer this quantity to VMBW.

Lines >7DC6 through >7DCE form the loop introduced in Listing 1 that allows the processor to be interrupted by the [QUIT] function. The computer will stay in this loop until [FCTN] [QUIT] is pressed. **HCM**

Part I of "Have No Fear" ran in Vol 2, No. 12 of 99'er HCM; Part II ran in Vol. 2 No. 13 of 99'er HCM; and Part III ran in Vol. 4 No. 1 of HCM. Beginning Assembly Language programmers may also want to consult chapter 5 of The Best of 99'er from Emerald Valley Books.



The RS-232 Interface: Your Link to the Periphery

In the world of computer interfacing, a "standard" isn't always standard. Knowing what your cable "sees" at each end is the key.

by Patricia Swift

If you use a microcomputer, sooner or later you will come across the "RS-232 standard interface." This article explains what the RS-232 interface is and how it is far from standard in practice. It also gives a practical example of how to design a cable to connect an RS-232 device to a computer.

What Is An Interface?

Computers use an interface to communicate with external devices (or "peripherals") If you want to attach a printer to a computer, you must have an interface and a cable in addition to a printer and computer, as shown in Figure 1. The interface is attached to the computer, and is usually a circuit board which is installed inside the main box or peripheral expansion box. The visible part of the RS-232 interface is the connector. Between this and a similar connector on the printer runs a cable with end connectors mated to those on both the printer and the computer. The cable's wires must also be arranged so that the interface and the printer "understand" each other.

How Is An Interface Used?

The RS-232 interface can be used to attach printers, modems, and other peripheral equipment to the computer. This "serial" interface (where data is sent one bit at a time over one wire) is available for most microcomputers, and many peripherals are RS-232 compatible. Unfortunately, there are many variations of the "RS-232 standard"—you can't just hook up your RS-232 printer to your RS-232 interface using a standard cable (with the possible exception of the new "smart cables") and expect it to work. The reasons why will be detailed later.

Another common way to attach a printer to a computer is with a "parallel" interface (where data is usually sent eight bits at a time over several wires). Although there is no single standard for parallel interfaces, there seem to be fewer variations than with the RS-232 interface. When we are unable to make an RS-232 connection work, we can usually use the parallel interface with very little effort. But, because many peripherals do not have a parallel interface, this option may not be open to you.

What Does An RS-232 Connector Look Like?

The most common RS-232 connector is known as the D-type with 25 pins (see Figure 2), although variations

abound: 7-pin, 9-pin, D-, and edge connectors are just a few. Connectors can be male or female; male connectors with pins are usually part of the cable, and socketed female connectors are usually found on the printer and interface. Figure 2 illustrates how the pins or sockets in the connector are numbered. Some RS-232 connectors have tiny numbers stamped near some or all of the pins or sockets.

The EIA RS-232 Standard

The Electronics Institute of America (EIA) has established a standard for the RS-232 interface that determines the signal that belongs on each pin, and the voltage levels of the signals themselves. Figure 3 summarizes this standard.

Common Variations

If every RS-232 interface and peripheral used the EIA standard, there would be no need for articles like this. Many printers, however, use a subset of the standard, and need only some of the signals given in Figure 3. Some devices vary slightly from the standard, using different pins for some signals. Sometimes the signal levels used are in different voltage ranges. For example, 0 to 5V is not defined; but this problem is beyond the scope of this article.

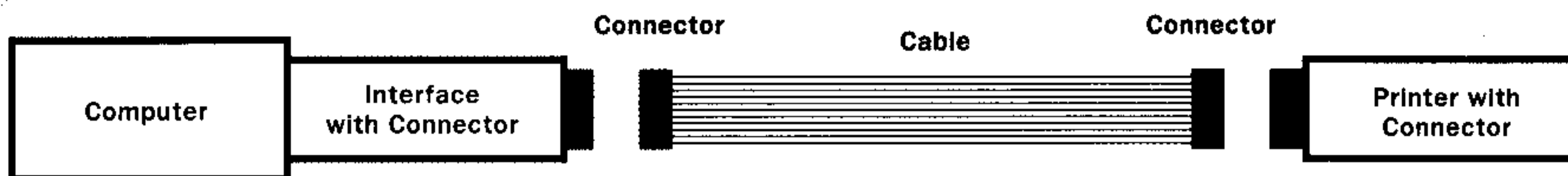
To understand what your printer and computer require to communicate, you must look at the RS-232 charts supplied by the printer and the computer interface manufacturers. We'll do that next, with a specific example, to give you some ideas of what to look for.

Example

We will look at attaching an Okidata Microline 82A printer to a Texas Instruments 99/4A computer via a TI RS-232 interface card. Both use 25-pin D-connectors like the one in Figure 2. The question is: How is the cable constructed? The cable will contain several wires, each wire connecting a pin from the interface at the computer end to a pin at the printer end.

The Okidata manual has a chart of serial interface signals and shows several possible cable arrangements for connecting the 82A to various computers (Okidata calls them "controllers"). For this example, we'll use the simplest arrangement listed. In Figure 5 (showing the TI to 82A cable wiring), the information in the column headed OKIDATA is straight out of the printer manual. The directions for the signals are also shown

Figure 1: ATTACHING A PRINTER TO A MICROCOMPUTER



in that manual. The diagram shows that only four pins need to be connected between the printer and "controller"—two grounds, one line for data into the printer, and one supervisory send data (or SSD) line for the Okidata to signal when the printer is busy or ready to receive data. These are pins 1, 7, 3, and 11 respectively. Pins 6 and 20 also need to be "jumpered" (wired together) on the printer end of the cable. The pin numbers on the computer side are not given in the Okidata manual, so we'll need to refer to the TI RS-232 interface manual for the rest of the story.

"The phrase 'RS-232 standard interface' is deceptive because so many manufacturers deviate from the standard."

The 99/4A RS-232 manual has a chart (shown in Figure 5) that describes its RS-232 connector. We will use the chart to fill in the pin numbers in the left-hand part of Figure 5. Okidata and Texas Instruments do not use the same mnemonics for similar functions—this situation is typical in the field. But we can locate the proper pins by deduction. Pins 1 and 7 are grounds; no problem there. To locate the pin for data out, we look for an output pin for data. The only candidate on the chart is pin 3. TI calls it TX while Okidata calls it TD. Finally, we have to locate an input pin for control, to be connected to the Okidata's SSD pin 11. The only input control line on the interface is pin 20, called "Data Terminal Ready."

Figure 4 shows the cable you'd need to connect the two devices. With this simple cable, the serial interface will work at speeds up to 1200 baud.

HCM

Figure 2: PICTURE OF RS-232 25-PIN D-CONNECTOR Looking at the Printer

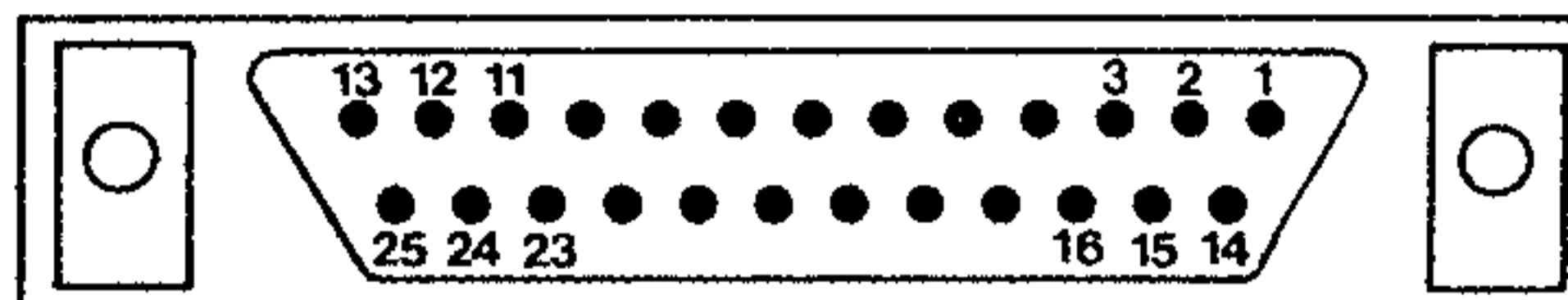


Figure 3: EIA RS-232 STANDARD, REVISION C

| PIN | SIGNAL | SOURCE |
|-----|---------------------|----------|
| 1 | PROTECTIVE GROUND | NONE |
| 2 | TRANSMITTED DATA | PRINTER |
| 3 | RECEIVED DATA | COMPUTER |
| 4 | REQUEST TO SEND | PRINTER |
| 5 | CLEAR TO SEND | COMPUTER |
| 6 | DATA SET READY | COMPUTER |
| 7 | SIGNAL GROUND | NONE |
| 8 | CARRIER DETECT | COMPUTER |
| 11 | REVERSE CHANNEL | PRINTER |
| 20 | DATA TERMINAL READY | PRINTER |

| SIGNAL LEVELS: | -25 TO -3V | -3 TO +3V | +3 TO +25V |
|----------------|------------|-------------|------------|
| DATA SIGNAL | MARKING | NOT DEFINED | SPACE |
| TIMING/CONTROL | OFF | NOT DEFINED | ON |

Figure 4: 99/4A RS-232 PIN DEFINITIONS

| PIN | DIRECTION | SIGNAL |
|-----|-----------|---------------------------|
| 1 | | GROUND |
| 2 | INPUT | DATA IN (RD) |
| 3 | OUTPUT | DATA OUT (TX) |
| 5 | OUTPUT | CLEAR TO SEND (CTS) |
| 6 | OUTPUT | DATA SET READY (DSR) |
| 7 | | GROUND |
| 8 | OUTPUT | DATA CARRIER DETECT (DCD) |
| 20 | INPUT | DATA TERMINAL READY (DTR) |

Figure 5: THE 99/4A TO OKIDATA 82A PRINTER CABLE

| COMPUTER | | | OKIDATA | | |
|-----------------|-------|-----|----------------------|-----|---------------------------|
| SIGNAL | | PIN | | PIN | SIGNAL |
| GROUND | PG | 1 | | 1 | PG GROUND |
| DATA OUT | TD/TX | 3 |▶ | 3 | RD RECEIVED DATA |
| GROUND | SG | 7 | | 7 | SG GROUND |
| DATA TERM READY | DTR | 20 | | 11 | SSD SUPERVISORY SEND DATA |
| | | | | 6 | DSR DATA SET READY |
| | | | Jumper pins 6 and 20 | 20 | DTR DATA TERM READY |

TECH NOTES



Lower Case Letters for TI-99/4A

Would you like to have true lower-case letters (rather than shortened capitals) available with TI BASIC? You can—simply by turning the shortened character set that the TI-99/4A uses into an alternate set by redefining characters 97 through 122 using the CALL CHAR command.

```
100 CALL CHAR(97,000007008384874)
110 PRINT "a"
120 GOTO 120
```

The above example shows you how to change the letter A from shortened uppercase to true lowercase using TI BASIC. Unfortunately, this solution uses some memory, and program initialization time is increased. If you have a Mini-Memory module, however, these problems can be solved by installing the following assembly language routine:

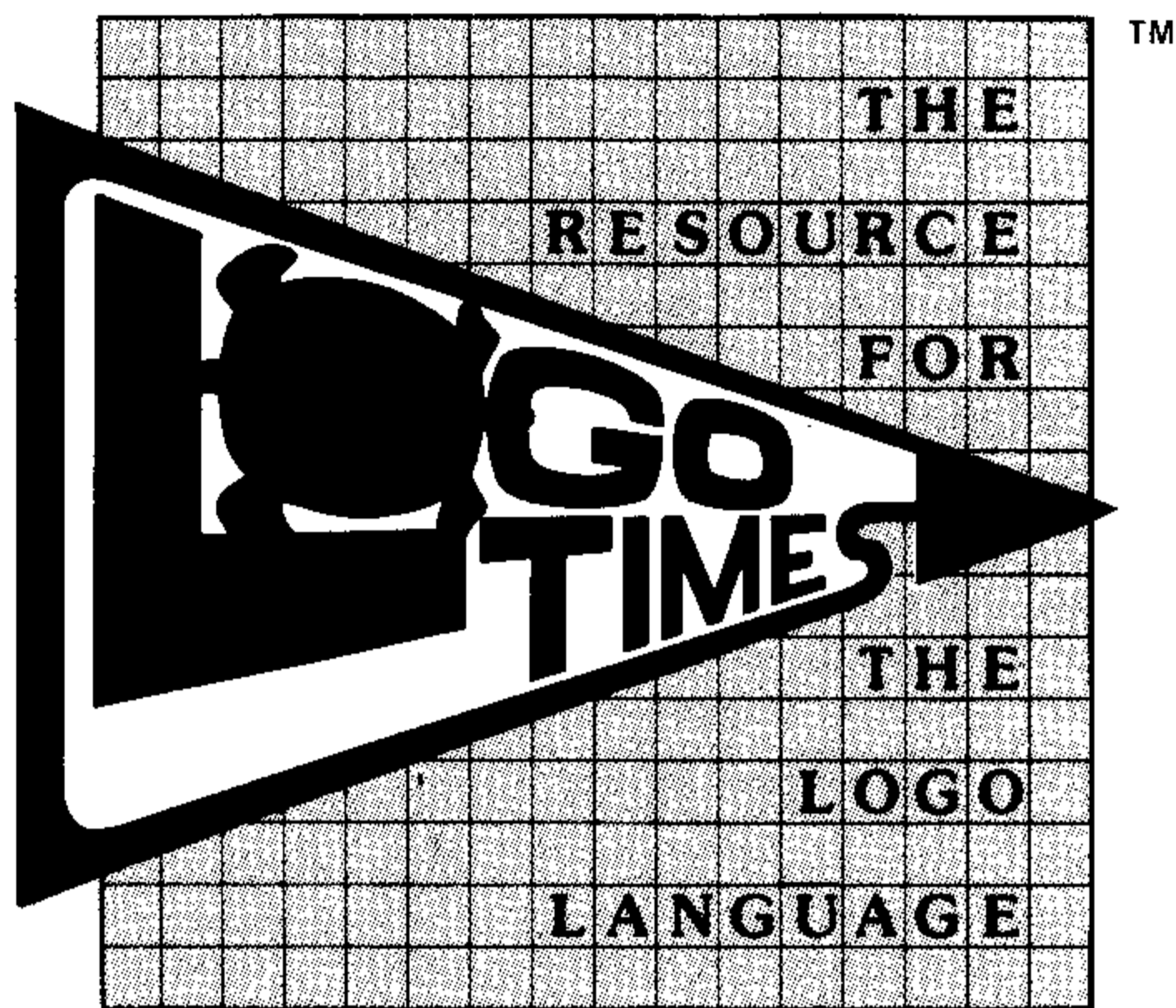
```
AORG >7D00
AL DATA >0000,>0070,>0838,>4874
DATA >0040,>4078,>4444,>4478
DATA >0000,>0038,>4440,>4438
DATA >0004,>043C,>4444,>443C
DATA >0000,>0038,>447C,>403C
DATA >0018,>2420,>7020,>2020
DATA >0000,>0438,>4438,>047C
DATA >0040,>4078,>4444,>4444
DATA >0010,>0030,>1010,>1038
DATA >0008,>0018,>0808,>4830
DATA >0040,>4048,>5070,>4844
DATA >0030,>1010,>1010,>1038
DATA >0000,>0078,>5454,>5454
DATA >0000,>0058,>2424,>2424
DATA >0000,>0038,>4444,>4438
DATA <0000,>0078,>4478,>4040
DATA >0000,>0038,>4454,>4834
DATA >0000,>0058,>6440,>4040
DATA >0000,>003C,>4038,>0478
DATA >0010,>3810,>1010,>1408
DATA >0000,>0048,>4848,>4824
DATA >0000,>0044,>4428,>2810
DATA >0000,>0044,>5454,>5424
DATA >0000,>0044,>2810,>2844
DATA >0000,>0044,>2418,>1060
DATA >0000,>007C,>0810,>207C
LWPI >70B8      SET UP WORK SPACE REGISTERS
MOV R11,R10     SAVE LINK TO BASIC
LI R0,>0608     ADDRESS OF ASCII 97 IN VDP RAM
LI R1,AL        ADDRESS OF CHARACTER CODES
LI R2,208       208 (26*8) BYTES TO WRITE
BLWP @>6028     MULTIPLE BYTE WRITE
B *R10         RETURN TO BASIC
END
```

Next, enter the following to name the routine LOWCAS, and add it and the entry point (>7DD0) to the REF/DEF table:

```
AORG >7FE8
TEXT 'LOWCAS'
DATA >7DD0
```

Then, you can call it from a BASIC program using a CALL LINK ('LOWCAS') statement. By keeping this routine permanently loaded in a Mini-Memory module, it will be available for you any time from TI BASIC. It only needs to be run once per program.

—HCM Staff



TM

LOGO SPREADSHEET

If you thought LOGO was limited to moving a turtle around the screen, this spreadsheet application will change your mind.

by Rich Haller
and the HCM Staff

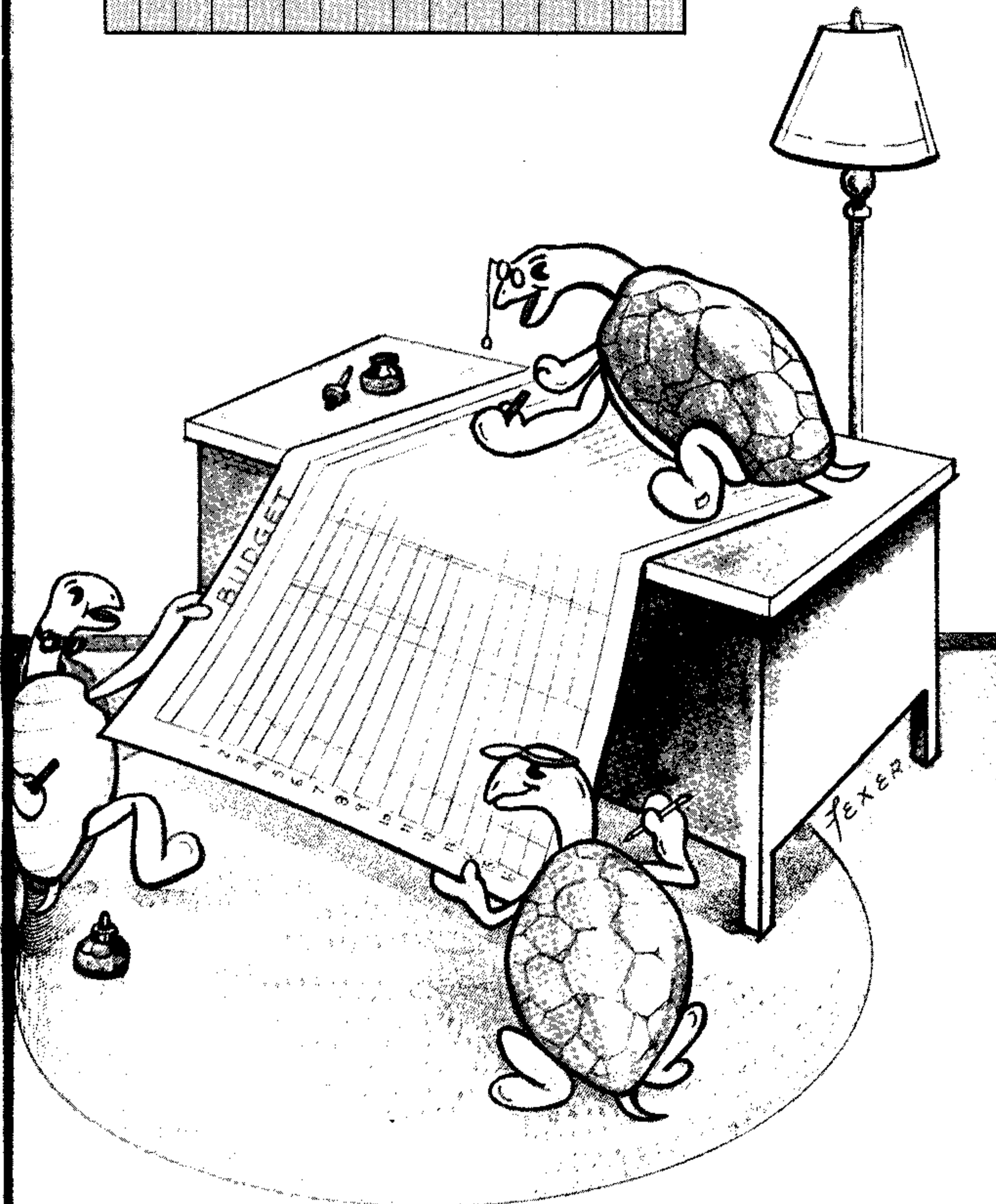
Although LOGO is perhaps best known for its graphics capability, it has other powerful features that make it a good candidate for use in writing a spreadsheet program. For example, LOGO variables can be numbers, strings of characters, or lists of values. What's more, variables are not restricted to one kind of value as they are in BASIC, where a variable must be either numeric or string. Because any cell in the spreadsheet can contain a number, a label, or a formula, the LOGO variable is ideal for representing the spreadsheet cells.

The TI Exception

The TI-99/4A home computer can't erase a variable from memory once it has been created, as the other machine implementations of LOGO can. Thus, it is impossible to store information in the spreadsheet by using variables to represent the cells. TI LOGO would never be able to distinguish between a value entered from an old spreadsheet or a new one. For this reason, a different algorithm was devised which places the values for the spreadsheet into a list. Each cell will be a list of two items in a greater list. The first item for a cell is the cell name, which will be used for searches. The second item in the list is the value for that cell. It could be a number, a string label, or another list (formula).

LOGO Runs Away

One of the most valuable aspects of writing a spreadsheet program in LOGO is the RUN command. This



command allows you to save a formula as a variable, and then **RUN** the contents of the variable. The output will be the result of the formula. Because of this feature, you will be able to enter a formula as the value of a cell, and have the result of the formula displayed on the spreadsheet. The following are samples of how procedures of a program execute equations from a list. This line can be used with the IBM LOGO, Apple LOGO II, and C-64 LOGO:

```
RUN FPUT "OUTPUT :CELL
```

This line can be used with the TI LOGO:

```
RUN (SE [ MAKE FIRST [ Y ] ] :EXPR )
```

Screen And Cell Size

The first step in writing a program like *LOGO Spreadsheet* is to develop a clear picture of how the end result should look from the user's point of view. Because of the difficulty in extending the spreadsheet beyond the borders of the screen, we decided to limit the workspace to five columns and eighteen rows, which should be adequate for simple operations. However, the cell size does vary slightly between systems:

| | |
|---------------|---|
| C-64 LOGO | Seven (7) characters, or integer numbers. |
| Apple LOGO II | Six (6) characters, or integer numbers. |
| IBM LOGO | Six (6) characters, or integer numbers. |
| TI LOGO | Five (5) characters, or integer numbers. |

The length of a formula is limited only by your system's maximum length for a string. But, a formula's result should never be a value that will exceed the space provided for it in a cell. Depending on your system, the result may be truncated, or an error message may appear in the cell.

Data Entry

Data entry takes two forms, depending on the system you are using. Both TI and C-64 LOGOs allow you to display graphics in the top two-thirds of the screen, and text in the bottom one-third. This allows the bottom lines to scroll without affecting the top of the screen where the spreadsheet is displayed. The TI and C-64 LOGOs also allow you to enter commands directly under the LOGO editor. The IBM and Apple LOGOs however, do not allow this to happen quite so easily, so the entire

screen is left in text mode. All data entry is done interactively with a procedure called **GETINPUT**.

To start the program, enter **SPREADSHEET**. You will then be asked if you want an old or a new spreadsheet. If you select **OLD**, the values assigned to the variables the last time you saved the file will be used. If you select **NEW**, the spreadsheet will be displayed with empty cells. At any time you can clear the contents of the spreadsheet by entering the command **NEW**. This erases all values stored in the spreadsheet—if you haven't saved them to disk before using the **NEW** command, they will be lost. On the IBM and Apple you can bypass the **SPREADSHEET** procedure when you want to use old data and simply enter **OLD**.

Spreadsheet Commands

A number of commands can be used to interact with the spreadsheet. You can enter numeric values (numbers), labels (words), or equations (lists) into any cell of the spreadsheet. You can then calculate one equation, or the whole spreadsheet. A detailed description of each of these commands is given below.

—ENTER :cell-name :cell-value

With this one command you can enter either numbers, labels, or equations into a cell, and the value will be displayed at the same time. If the value you enter into the cell is an equation, then the result of the equation will be displayed. Some examples of the **ENTER** command are as follows:

Numeric Values:

| | |
|------------------|--------------------------|
| ENTER "A1 110 | Place 110 in cell A1. |
| ENTER "A2 2036 | Place 2036 in cell A2. |
| ENTER "D13 98375 | Place 98375 in cell D13. |

Label values:

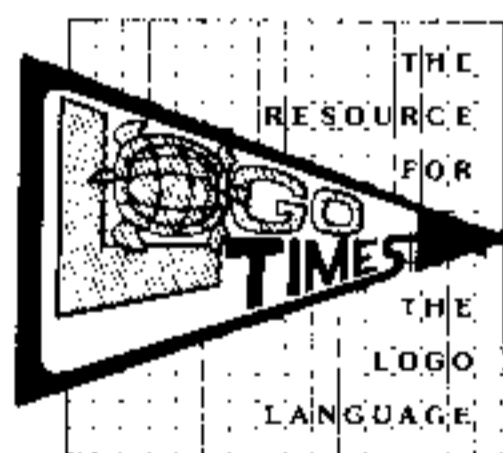
| | |
|-------------------|----------------------------|
| ENTER "B1 "INCOME | Place "INCOME" in cell B1. |
| ENTER "E16 "TAXES | Place "TAXES" in cell E16. |
| ENTER "C8 "Totals | Place "Totals" in cell C8. |

Equations: (IBM, Apple, and C-64 only)

| |
|--|
| ENTER "A12 [:A1 + :D13 - :A2] |
| ENTER "E14 [(:A1 - :A2) * (:D13 / 2000) + 1] |

It is possible to use an equation within another equation. For example, if cell A12 contained the equation shown above, you could then use A12 in the following equation:

Introduction



LOGO Times is an information resource for users who want to create their own personal languages—languages that will easily allow them to communicate with the computer in a

totally new audiovisual realm of applied imagination, exploration, and self-discovery. The articles on these pages concern the use of the LOGO language, but readers do not need any additional software or equipment (or even a computer) to understand and learn from the material presented here.


If readers want to actually experience a LOGO environment, they will need a computer, the requisite software and/or cartridges, and any additional hardware required for a particular implementation. A disk drive is required for some LOGO implementations, but in other cases, a user's work may be saved on cassette tape, or copied into a notebook (for later re-keyboarding).

The varieties of LOGO we'll consider include—but are not limited to—Terrapin LOGO for the Apple II, II+ or IIe and the Commodore 64, TI LOGO for the TI-99/4A, and LOGO Computer Systems LOGO for the IBM PC and PCjr.

- **Apple:** Terrapin LOGO requires an Apple II, II+ or IIe with 64K of RAM, one disk drive with controller, and a blank, initialized disk.
- **Commodore 64:** Terrapin LOGO requires a Commodore 64 with a VIC-1541 Disk Drive and a blank, initialized disk.
- **TI-99/4A:** TI LOGO requires the TI LOGO or TI LOGO II cartridge and a compatible 32K memory expansion unit. A cassette recorder may be used for storage, but a compatible disk system is recommended for convenience.
- **IBM PC or PCjr:** LOGO Computer Systems LOGO requires the PC or PCjr with 128K bytes of RAM, one disk drive, and a blank, initialized disk.

In each issue, one or more of the articles may refer to or build upon the topics discussed in a previous article. It is therefore recommended that for maximum benefit and understanding, new readers obtain the appropriate back issues of *Home Computer Magazine* containing *LOGO Times* articles.

LOGO Listings

As you enter LOGO statements, the last thing you do at the end of every statement is to press { **ENTER** } on the TI and IBM (the key with the  symbol), or { **RETURN** } on the Commodore 64 and Apple. This signals

the system to begin a new line. In our typeset listings, single LOGO statements may carry over from one line to the next without ending. The end of a LOGO statement is marked with a curved arrow (↷) to indicate that you press { **ENTER** } or { **RETURN** } at that point.

Notice

LOGO Times is actively soliciting articles. Manuscripts should be typed double-spaced, and accompanied by a cassette tape or disk if containing any lengthy procedures or graphics.

Send all materials to:

LOGO Times Editorial Dept.
Home Computer Magazine
1500 Valley River Dr., Suite 250
Eugene, OR 97401

All mail directed to the Letters-to-the-Editor column (*Letters on LOGO*) will be published in accordance with the conditions set forth on *Home Computer Magazine's* Masthead page.

Our Contributing Editors

Henry Gorman, Jr. Roger B. Kirchner
William M. Goodman Rich Haller

• *LOGO Times* is a trademark of Emerald Valley Publishing Co. •

“Use caution though when creating equations within equations, for if a cell appears within its own equation, or within an equation it uses, you will have problems with recursion.”

ENTER [(RUN :A12) + :A2]

In this example, the result of the equation in cell A12 is added to the value in cell A2. Use caution though when creating equations within equations, for if a cell appears within its own equation, or within an equation it uses, you will have problems with recursion. Recursion occurs when one procedure keeps calling itself in an infinite loop. This usually results in an OUT OF SPACE error.

—Equations: (TI-99/4A)

TI LOGO handles equations a little differently because of the method it uses to store the cell values. This method of building equations may seem to have its drawbacks, but as you will see later, it also has many advantages. Here are some typical equations:

```
ENTER "A12 [ (VAL [ A1 ]) + (VAL [ D13 ]) - (VAL [ A2 ]) ]
ENTER "C13 [ ((VAL [ A1 ]) - (VAL [ A2 ])) * ((VAL [ D13 ])
/ 2000) + 1 )
```

The VAL function returns the numeric value of the cell, whether it is a number or an equation. This is where the advantage comes in. In this version of *LOGO Spreadsheet* it makes no difference whether the cell is a number or an equation in extracting the cell's value. In the above example, A1 could have been either a number or an equation, and the equation using A1 would be the same.

UPDATE :cell-name

If you have a full spreadsheet, it could become quite time-consuming to recalculate every formula when only a few changes are needed. This command lets you pick and choose the formulas you want recalculated. For a cell to be recalculated, it should have a formula in it. To enter the cell desired, use the UPDATE command like this:

```
UPDATE "C12
```

This command lets you update only one cell at a time. In this example, the cell at column C row 12 will be updated.

RECALC

This command is useful if you wish to recalculate the entire spreadsheet. If any values have been entered or changed, then RECALC will display them.

OLD (IBM, and Apple only)

This command is available only with IBM LOGO and Apple LOGO II. It is necessary because in special cases, a bad command entry may cause a LOGO error message to be displayed. If this occurs, the program is no longer interactive (you are at the TOPLEVEL of LOGO). It's also possible that the error message will cause the screen to scroll, or even display the message right in the middle of the spreadsheet. If this happens, you can simply enter OLD and the spreadsheet will automatically be redisplayed. In addition, the RECALC procedure is automatically called at this time. If the error was caused by a bad cell formula, you should correct the problem before entering the OLD command. To correct a formula, you can use either the ENTER command as provided, or the MAKE primitive to reassign a value to a variable (the variable being the cell with the problem). If this isn't

satisfactory, you can erase a variable name by entering ERN and the variable name.

Complex Formulas

(Apple, IBM, and C-64 only)

Creating formulas can become a very complex task, even in simple spreadsheets. We added two procedures to help make the task easier with Apple LOGO II, IBM LOGO, and C-64 LOGO, and one procedure for the TI LOGO version.

CSUM :col-letter :beg-row-number :end-row-number

This command can be used to total up columns. You can specify the column you wish to total, the beginning row, and the ending row. A typical example might look like this:

```
ENTER "A18 [ CSUM "A 1 9 ]
```

In this example the values in column A, rows 1 through 9 are totaled up and placed in cell A18. The cell receiving the total from the CSUM procedure does not have to be in the same column as the one on which the CSUM procedure is working. The same formula above could have been assigned to cell "D7 just as easily. Notice that the column name is preceded by a quote. This is a requirement whenever LOGO works with words that are not numbers. The next two values are numbers, and do not require the single quote. There should also be a space between the column letter and the beginning row, and between the beginning and ending rows.

RSUM :row-number :beg-col-letter :end-col-letter

This procedure works the same as the CSUM procedure except that it provides row totals. A typical example looks like this:

```
ENTER "E18 [ RSUM 1 "A "E ]
```

The two procedures above can be mixed to create some very complex formulas. You also can directly insert normal cells with values, and integer numbers:

```
ENTER "E18 [ (CSUM "A 1 18) + (RSUM 4 "A "C) * :E17 + 25 ]
```

Complex Formulas

(TI-99/4A only)

The TI LOGO version of *LOGO Spreadsheet* uses one procedure to replace the two procedures used in the other versions. The procedure ADDUP lets you obtain a total of all the cells in a rectangular area of the spreadsheet. You simply supply the upper-left cell and the lower-right cell in a list, and the sum of all cells between will result. An example would be:

```
ENTER "E18 [ ADDUP [ A1 C3 ] ]
```

In this example, the value from cells A1 through A3, B1 through B3, and C1 through C3 were all added together and placed in cell "E18. Notice that the colon (:) is not required on the cell names inside the formula. This is because they are not variable names. This procedure can also be used in complex formulas of the type:

```
ENTER "E18 [ (ADDUP [ A1 C3 ]) + (VAL [ D4 ]) * 4 ]
```

Expanding The Program

You may easily expand on any of the procedures presented here, or even add your own. The beautiful thing about LOGO is that you can create your own language. Make your commands in the form of procedures, to assist with shortcuts in your own personalized spreadsheet. You may also want to define a procedure which initializes some of the cells to a predesigned set of formulas and labels. The limit to this program's expansion and capabilities is only your imagination.

HCM



MISSIONARY IMPOSSIBLE



*Crossing the river?
Beware the cannibals—they'd love
to have you drop in for lunch!
An old puzzle finds a new form in LOGO.*

by Roger Kirchner
and the HCM Staff

One of the central problems in artificial intelligence research is the difficulty of devising efficient algorithms for "intelligent" search. By this we mean the ability to assess a problem, test the consequences of specific actions, and search for a legal solution. A good algorithm would not only find a *legal* solution, but would also search for the *best* solution. A classical example is the old "Missionaries and Cannibals" puzzle:

Three missionaries and three cannibals come to a river. There a boat transports people back and forth across the river, but it can hold a maximum of only two people at a time. The problem is to get everyone across safely. Everyone is "safe" as long as the cannibals never outnumber the missionaries on either shore. There can be cannibals on a shore with no missionaries.

In order to consider a problem like this, it is necessary to represent each possible state of the game, determine the

rules which apply in each state, and devise procedures to control the order in which rules are applied in transforming a start state into a goal state.

Designing The Solution

Several logical problems must be solved before this program can be written:

1. Decide how to represent the state of the puzzle—who is where, and where is the boat?
2. For a given state, determine what the legal boat loads are.
3. Make it possible for a user to play with the puzzle. The user must be able to enter the number of missionaries and cannibals to go in the boat. Play should continue until the puzzle is solved, and the program recognizes a solution.
4. Provide an automatic puzzle solution. For each state, the program should be able to calculate a list of possible moves and then try them out according to the rules. One of four things can happen at each stage:

— We arrive at a certain state for a second time. If this happens then we should back up and try again.

— We arrive at our goal state, at which time we will have solved the puzzle.

— We have tried all possibilities without success, in which case there would be no solution.

— We arrive at a new state. In this case we might be on a possible solution track. We compute the list of boat loads possible from this state, and try the first one in the list.

Representing A State

Many possibilities exist for representing the state of the puzzle. Obviously, we could represent a state with a list. But exactly how? The following method was chosen.

```
[ S [ m1 c1 ] [ m2 c2 ] ]
```

Here, **S** stands for the side of the river where the boat is positioned. The symbols **m1** and **c1** represent the number of missionaries and cannibals on the same side as the boat. The number of missionaries and cannibals on the opposite side of the river from the boat are represented by **m2** and **c2**. For example, [L [3 3] [0 0]] represents the start position. The boat is on the left shore (L), and there are three missionaries and three cannibals on that shore. No one is on the opposite shore (the right side).

The boat loads can also be represented by a list of two values: [m c]. The number of missionaries in the boat is **m**, while the number of cannibals is **c**.

"A good algorithm would not only find a legal solution, but would also search for the best solution."

Playing With The Puzzle

To play the puzzle, type **MISSCAN** and press [RETURN] or [ENTER]. You will be asked if you would like to solve the puzzle yourself. If you want to solve the puzzle, respond by entering **Y** for yes.

The **PLAY** procedure will then initialize the game. "D (for Description) is initialized to the start position. "D will contain the current state of the game. **PLAY** then calls **PLAY1**, which does all of the work—including prompting you for the number of missionaries and cannibals to send across in the boat.

The variable "BTLD contains the contents of the boat. The **MOVE** procedure is then called using "D and "BTLD. The **MOVE** procedure will output the new state if the move is legal (safe), or it will output [] if the move is not safe. **PLAY1** will be recursive until you enter a legal boat load. Once a legal boatload has been entered, the program will call the **ACTION** procedure to actually move from one shore to the other.

Automatically Searching For A Solution

After entering **MISSCAN** to start the puzzle, you will be asked if you would like to solve the puzzle yourself. If you want the computer to solve it for you select **N**.

The key to this portion of the program is the **BACKTRACK1** procedure. This is the procedure which will find the solution, and then remember the shortest path from the starting position to the current position.

Its purpose is to search for legal paths from one position in the puzzle to another. The procedure is recursive, calling itself until it locates the proper path to the end of the puzzle.

During the **BACKTRACK1** procedure, the results of the search are displayed on the screen. The Commodore, IBM, and Apple versions display only the contents of the boat during the search. The TI version actually goes through the whole sequence of moving the boat back and forth while it searches for a solution.

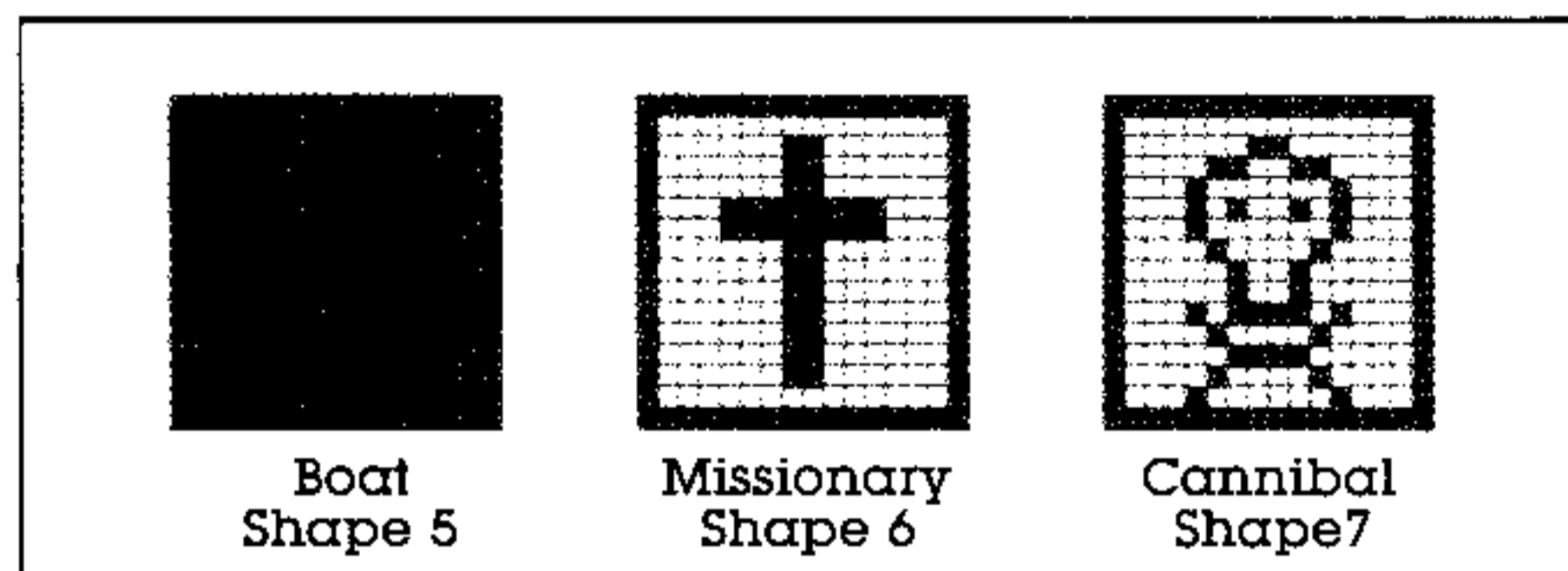
After a solution is found, control is given to **DO**, which displays each step in the solution by moving the missionaries and cannibals back and forth in the boat. After seeing the solution displayed, you can view it again or return to the main menu.

Graphics

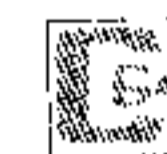
—TI LOGO II



TILOGO II is a fantastic LOGO for demonstrating the missionaries and cannibals puzzle because of its sprites. Sprites are ideal for depicting both the people and the boat. We merely put six sprites on each side of the screen and assign each sprite to carry one of three shapes: a missionary, a cannibal, or a blank for neither. The boat is made up of four sprites—two sprites on top of two others. We then assign the two sprites on top to carry one of the three shapes. The bottom two sprites make up the boat itself. The shapes for the cannibals and missionaries are shown here. They can be entered using the **MAKESHape** command.

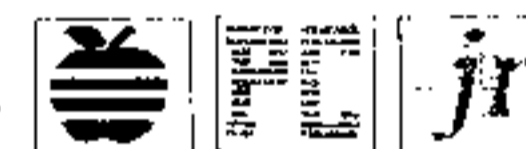


—Commodore LOGO



Commodore LOGO uses sprites to place the missionaries, cannibals, and boat on the screen. Three sprites are used for the missionaries, three for the cannibals, and two for the boat. To move around the screen, the sprites' X and Y coordinates are changed. Three shapes are used, and must be loaded from the LOGO utility disk, which comes with the LOGO language disk. After loading the program and before running it, insert the utility disk. Type **MISSCAN** and press [RETURN]. The LOGO procedure **READSHAPES** will **BLOAD** the shapes into memory for you.

—IBM LOGO and APPLE LOGO II



The lack of sprites in both IBM and Apple LOGOs makes it necessary to use normal text characters for the missionaries, cannibals, and the boat. The missionaries are represented by a capital **M**, and the cannibals are a capital **C**. The boat is constructed to look like so:

```
{ — }
```

To move the boat, it's necessary to locate the cursor with the **SETCURSOR** primitive, and then print the boat repeatedly with either a leading or trailing space, depending on whether it's traveling left or right across the screen.

HCM

For the Key-in listing see HCM PROGRAM LISTINGS Contents on page 93.

TI LOGO Versus TI LOGO II

The TI version of this procedure was written to work with TI LOGO II not TI LOGO. There aren't enough **NODES** available to enter all of the procedures with TI LOGO. If you do not have TI LOGO II you can at least *play* the game by entering the following procedures:

TI LOGO Procedures

| | |
|---------------------|---------------------------|
| TO SAVE? :P | TO DISPLAY :N :S :T |
| TO ADD :P :Q | TO SWITCH :SIDE |
| TO SUB :P :Q | TO PLAY |
| TO LEGALS :LOADS :D | TO ACTION :POS |
| TO INTVALS | TO MOVE :LOAD :SIDE :P :Q |
| TO MISSCAN | TO PLAY1 |
| TO SPLIT :N :S1 :S2 | TO NONO |
| TO MAP :X :YXS | TO SHOUT |
| TO UNLOADBOAT | TO INIT |
| TO MOVEBOAT :S | TO SETBOAT :S |
| TO LOADBOAT :LOAD | TO SHOW :D |

```

710 OPEN #1:"CS1",INPUT,FIXED::GOTO
740
720 DISPLAY AT(9,1):"FILE NAME (" :FN$:T
AB(22):" :ACCEPT AT(9,12)SIZE(-
10):FN$: :OPEN #1:"DSK1."&FN$,VARI
ABLE:60,INPUT::GOTO 740
730 DISPLAY AT(9,1):"ENTER DEVICE AND F
ILE NAME:"DEV$: :ACCEPT AT(10,1):
DEV$: :OPEN #1:DEV$,VARIABLE 60,IN
PUT
740 INPUT #1:R::FOR Z=1 TO R::INPUT
#1:AS(Z)::NEXT Z::CLOSE #1::G
OTO 200
750 DISPLAY AT(1,7):MS(7):"1) CS1 - C
ASSETTE" : "2) DSK1 - DISK DRIVE #1
" : "3) OTHER DEVICE"
760 GOSUB 860::IF K<49 OR K>51 THEN 7
60 ELSE ON K-48 GOTO 770,780,790
770 OPEN #1:"CS1",OUTPUT,FIXED::GOTO
800
780 DISPLAY AT(9,1):"FILE NAME (" :FN$:T
AB(22):" :ACCEPT AT(9,12)SIZE(-
10):FN$: :OPEN #1:"DSK1."&FN$,VARI
ABLE 60,OUTPUT::GOTO 800
790 DISPLAY AT(9,1):"ENTER DEVICE AND F
ILE NAME:"DEV$: :ACCEPT AT(10,1):
DEV$: :OPEN #1:DEV$,VARIABLE 60,OU
TPUT
800 PRINT #1:R::FOR Z=1 TO R::PRINT
#1:AS(Z)::NEXT Z::CLOSE #1::G
OTO 200
810 AC=ASC(AS(Z))-100::AL=ASC(SEGS(AS
(Z),2,1))-100::DS=SEGS(AS(Z),3,AL
)::VS=SEGS(AS(Z),AL+3,LEN(AS(Z))-A
L-2)::RETURN
820 DISPLAY AT(1,(28-LEN(MS(K-48)))/2+1
):MS(K-48)::FOR Z=1 TO 17::DISPL
AY AT(2+Z,1):STR$(Z):" :NS(Z)::N
EXT Z::RETURN
    
```

```

830 FOR Z=1 TO L-F+1::DISPLAY AT(2+Z,
1):STR$(Z):" :NS(F+Z-1)::NEXT Z
::GOTO 860
840 FOR Z=1 TO N::READ X,Y,PS::DISP
LAY AT(X,Y):PS::NEXT Z::GOTO 86
0
850 CALL CLEAR::FOR Z=1 TO N::READ
X,Y,PS::DISPLAY AT(X,Y):PS::NEX
T Z::CALL SOUND(10,440,0)
860 CALL KEY(0,K,S)::IF S=0 THEN 860 E
LSE RETURN
870 DATA ADD DATA,CHANGE DATA,DISPLA
Y DATA,TOTALS,PRINT REPORT,LOAD DATA F
ILE,SAVE DATA FILE,EXIT PROGRAM
880 DATA MEDICINE & DRUGS,DOCTORS & DEN
TISTS,MED. TRANSPORTATION,OTHER MEDI
CAL,STATE & LOCAL TAX
890 DATA REAL ESTATE TAX,AUTO SALES TAX
,OTHER TAXES,HOME MORTGAGE INT,CRED
IT CARD INT,OTHER INTEREST,CASH CON
TRIBUTIONS
900 DATA OTHER CONTRIBUTIONS,CASUALTY &
THEFT,UNION & PROF. DUES,TAX PREP.
FEES,OTHER DEDUCTIONS
910 DATA 12,6,DEDUCTION FILER,24,3,PRES
S,ANY KEY TO BEGIN
920 DATA CS1,DSK1,OTHER
930 DATA ADD NEW DATA,CHANGE DATA,DISPL
AY DATA,RETURN TO MENU
940 CALL CLEAR::DISPLAY AT(12,1):"ARE
YOU SURE YOU WANT TO EXIT THE P
ROGRAM AND ERASE THE CONTENTS OF M
EMORY?"
950 DISPLAY AT(24,1):"ENTER (Y/N):"::
ACCEPT AT(24,14)SIZE(1)VALIDATE("YN
")::PS::IF PS="N" THEN 200 ELSE EN
D
    
```

HCM

PROGRAM LISTING



HCM Program Bug

DeBUGS on Display

Bugs—problems that cause programs not to function properly—are an inevitable consequence of programming. Programs are complex systems that are difficult to test exhaustively, so occasionally bugs slip through even the best testing procedures. When such a bug in one of our programs comes to our attention, we print a correction in this column so that our readers can correct their programs as soon as possible.

In the August '84 issue of the magazine, we published a game called Wild Kingdom (the listing appeared on page 104). If you tried to play the game with joysticks, you may have run into a little bug—the program as published won't work with joysticks. However, the keyboard still functions properly. To get joysticks working, you will need to change two program lines:

```

190 IF FR$ = "Y" THEN JS = 0 :: CALL CLEAR :: DISPLAY AT(12,5)
BEEP:"PLACE ALPHA LOCK UP" ELSE JS = 1 :: GOTO 210
400 IF JS = 0 THEN CALL JOYST(2,E,F):: J = ((E + 3 * F) / 4) + 5 :: ON J
GOTO 530,480,530,420,530,460,530,440,530
    
```

The TI-99/4A version of Cyber Cypher, which appeared in the August '84 issue, does not display the low score properly. Add the following lines to the program on page 74 to correct this problem:

```

956 R = 19
957 C = 7
958 MS = STR$(L)
959 GOSUB 2850
    
```

Delete line 960

```

2465 CALL HCHAR(19,8,32)
2492 IF L < 10 THEN 2500
2494 CALL HCHAR(19,8,48)
    
```

The August 84 issue of Home Computer Magazine also contained a problem in the TI-99/4A version of Bars and Plots. Line 930 on page 69 should read:

```

930 IF M + 1 > 32 THEN 840
    
```

The TI-99/4A Sea of States program—published in Vol. 4, Issue 2—is too long to run without memory expansion included. The program will run, however, without memory expansion with the following modifications to the program listing. This "fix" will allow you to use the program with a cassette tape and requires that your system NOT have a disk drive connected. (The disk drive uses up part of the memory as a disk buffer.)

```

Change these lines, which begin on page 14, to read as follows:
190 DEF SB = SEA(S(5),S(6))
260 FOR A = 1 TO 8 :: FOR B = 1 TO 8 :: SEA(A,B) =
INT(TAN(RND * 1.37)) :: NEXT B :: NEXT A :: RETURN
270 CALL DELSPRITE(#3) :: SEA(INT(RND * 8) + 1,INT(RND * 8) + 1) = 4
    
```

```

530 DISPLAY AT(15,1):" HE SAYS "I WILL NOT LET YOU GO
EASILY WITHOUT ANSWERING THIS QUESTION:"" :: SEA
(S(5),S(6)) = 1.5
    
```

```

630 AS = "YOU HAVE FOUND A WRECK AND:" :: C = 7 :: SH = 124
:: GOSUB 660 :: SEA(S(5),S(6)) = 5 :: GOTO 480
    
```

```

820 FOR A = 5 TO 15 STEP 5 :: DISPLAY AT(17,A):"UP" :: GOSUB
1070 NEXT A :: GOSUB 260 :: DISPLAY AT(20,1):"YOU'RE
AT LEVEL 2" :: GOSUB 1070
    
```

ADD the following line to the program:

```

255 GOSUB 260 :: GOTO 270
    
```

DELETE the following lines:

```

100
120
170
    
```

New subscriber Angela McManus wrote to us wondering whether she had the entire program for Cannibals of Vol. 4, No. 2. Yes, the program is complete, but the "Explanation of the Program" for the TI-99/4A version included too many line numbers. The corrected "Explanation of the Program" on page 63 should read:

- 1830-1920 End of the game messages and restart the game.
- 1930-1970 Display the score.
- 1980-2100 Subroutine to scan the keyboard for Dr. Livingston's movement.
- 2110-2270 Move Dr. Livingston.
- 2280-2310 Subroutine to scan the joysticks.
- 2320-2350 DATA containing graphics pattern information.
- 2360 DATA containing character set color information.

In the article "3D-Ile: Apple Graphics in Three Dimensions" of Volume 4, Issue 2, there is a discrepancy between one of the menu displays and the magazine text's description of the display. Once you are in the editing mode and you have cleared an object from the screen, the menu that appears will say:

- 1) RESUME EDITING OBJECT IN MEMORY
- 2) SAVE OBJECT IN MEMORY TO DISK
- 3) RETURN TO SYSTEM MENU

This menu is different from the initial editor menu you see before you begin creating your graphics.

HCM

EXCLUSIVELY FOR TI-99/4A USERS

LIMITED SUPPLY

LIMITED SUPPLY

Special CLOSE-OUT Prices on All BACK ISSUES of



Exclusively covering the TI-99/4A home computer. Complete with ready-to-type-in-and-run program listings! The original 99'er Magazine and 99'er Home Computer Magazine were the forerunners of the current Home Computer Magazine.



ISSUE #1 ISSUE #2 ISSUE #3 ISSUE #4 ISSUE #5
 ISSUE #6 ISSUE #7 ISSUE #8 ISSUE #9 ISSUE #10

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 Concatenation • Picking the Pieces 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 Jaws Arcade Game • A Knight in TI BASIC • LOGO Has Style • ASPIC: A Language for Children • A p-System Beginner's 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)

• Tea-Scribe: A Text Editor for the Home Computer • A Christmas Computer Carol • Managing a Mailing List • The Future Way • Paradox: The Arcade Game • Plotting With the Home Computer—Pixel by Pixel • Preventing the Station—On 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 Terminate • The 99'er Gold Rush—An Arcade Adventure in the Home • Star 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

FEBRUARY 1983 (Partial Contents)

• Texas Instruments at the Winter Consumer Electronics Show • Home Computer Printers on Review • How to Create Math Databases in LOGO • Vectors in LOGO • AEPIC: A Language for Teachers • The Joy of Advertising—Part 2 • Programming Pointers with Chuck-A-Luck—Part 2 • Interview With the Voice of Paradox • Why You Need a Printer for Your Home Computer • Life on Titan: Space Game • Night Blockade: Bullfighting Game • Tower of Hanoi: Pacified 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 4A 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 Jersey—An Overview of Remote Controllers • Paradox Strategy • Converting Extended BASIC to Assembly Language • Matrix Muncher • Mini Memory Disassembler Utility • Pulling the Shads on Spillies • 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 • Text Cipher Writes and Decodes Secret Messages • Crosswords—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 Scales in Mini Memory • Colorful Word World—Reading Readiness for Preschoolers • Gameware: Buffet's

MAY 1983 (Partial Contents)

• A Consumer's Guide to Word Processing • Word Processing Market Basket • A Generalized Filing Program for VIPs • The Multipan Medium Balances Your Checkbook and Budget • Activity Account 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 TI99/4A • The Wonders of Diskette Storage • Beezies—a Multi-Screen Strategy Game • Lost Ruins—an Archeological Dig Game • 3-D Illustrations 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 Asarok Game • Probal Your Station in the Space Zapper Game • What Multipan 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 1822 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-Roo • A Book Review of Learn BASIC for CC-40 Users • 3-D Animation with the TMS 9918A Video Chip • Game Reviews • Group Grapevine • and much, much more.

AUGUST, 1983 (Partial Contents)

• The Home Computer Goes To Work • Bill One, Part Two at the Fashion Factory • Better Business Bu

Graphs in Graphic Persuasion • An Ensemble of Assemblers • Cashflow Helps Money Management • Keystrokes for Trifly Folks—a Review of Typewriter • Game Reviews of Career Quest and Starprobe '89 • Counting Fun for Preschoolers • Peripheral Vision '89 • Mean Machines and Small Potatoes • Multipan Medium Groups Gets into Rectangles • Turtle Text, a LOGO Word Processor • Group Grapevine • Hello, Little Brother—CC-40 Speaks to 99/4A • Pate's Progress Looks at Student Assessment • Pocket Sunrise • A Little Investigator • Gameware Buffet's Jungle Jim, Success Formula, and much, much more!

SEPTEMBER, 1983 (Partial Contents)

• Adding on to Your Home Computer • Offer Directory of Commercially Available Software, Accessories and Peripherals • Peripheral Vision '89 Hardware Reviews • Pocket Sunrise, Part Two in Extended BASIC • Byte Lightning—Mini Memory Plays a Game • T-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 No Win and Crime and Punishment • Group Grapevine • 99'er Hall of Fame • 99'er Digital Update on New Products and much, much more!

OCTOBER, 1983 (Partial Contents)

• Adventures and Fantasy with Your Home Computer • Do-It-Yourself Adventure • Flopprocessor Emulator, a 4-Bit Microprocessor • Have No Fear: Assembly Language Won't Bite, Part 1 • Make Your Mark • Les Izmore and Doug Computing Cartoon • Once Upon a Tortoise Shell—A LOGO Adventure • Turtle Strut • Number Nibbler for Children • Lots of Pits on Your CC-40 and HX1000 Four-Color Printer • Multipan Medium—Bartender • A Grizzly 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 Bite, Part 2 • Squeezing the Most Out of TI BASIC • T-WRITER Tutorial • LOGO Lesson • Interview with Gene Osborn • T-WRITER: At Home in the Office • PLATO's Progress • Les Izmore and Doug • The Multipan Medium • Computer Assisted Instruction—99'er Interviews the Kids • Gameware Buffet: Taco Man and Robo Chase • Game Reviews of Jail Break and Arithmetics • Hall of Fame • 99'er Digital • and much, much more.

DECEMBER, 1983 (Partial Contents)

• The Home Computer Goes To Work • Bill One, Part Two at the Fashion Factory • Better Business Bu

CLOSE-OUT SPECIAL

Hurry, Supplies Are Limited!

Any 3 back issues for only \$5.95 postpaid in the U.S.

Any 6 back issues for only \$10.95 postpaid in the U.S.

NEVER BEFORE OFFERED!

The Best Programs Available On Disk or On Tape From Each Issue Above.

only \$3.95 per issue, postpaid in U.S.

—NOT FOUND ON ORDER FORM—Write on order form or on separate sheet issues desired and choice of tape or disk.

*Requires corresponding magazine back issues for program documentation.

* SUPER BONUS *

This offer not found on center bind-in order card. Please write "Super Bonus" on card when ordering and indicate your 12 choices.

Any 12 for only \$21.90 PLUS a FREE Simon's Saucer Flipper/Snapper™

- A quality, ready-to-run game on cassette tape or diskette.
- A complete, easy-to-use programming lesson on a deck of colorful flip cards.
- A durable and attractive collector's case for your software library.



All prices include postage in the U.S. To Order, Use Order Card In Center Of Magazine.

ALL PROGRAMS IN THIS MAGAZINE



ONLY \$3.95 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 ON DISK™ or ON TAPE™ to newsstand purchasers or subscribers of this magazine.

For only \$3.95 (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.