Software Giveaway with Purchase of this Magazine
See bound-in card behind front cover

HOME COMPUTER™ magazine

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

August, 1984

$3.00 in USA
$3.75 in Canada

- FREE Spreadsheet Software for the Apple II Family, C-64, IBM PC/PCjr, and TI-99/4A
- Early-Learning Color Fun on the VIC-20
- A Graphic Adventure in Learning for PCjr & Apple
- How to Write Games on C-64 & TI-99/4A

Continuing 99'er Magazine's Complete Coverage of the TI-99/4A

PRODUCTIVITY COMES HOME

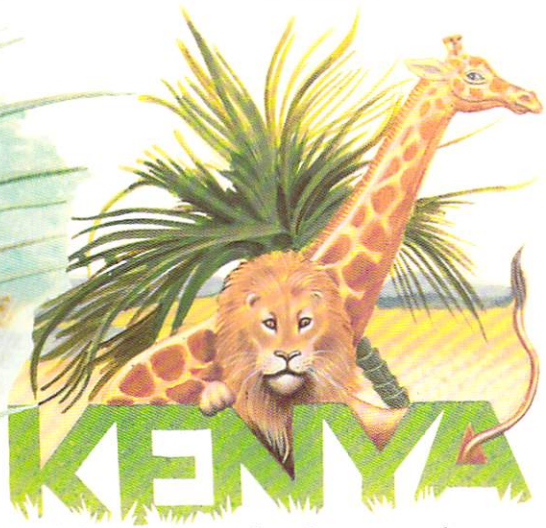
- Reviews Galore— Word Processing, Database, & Spreadsheet Programs
- Computer Gaming with Terrifying Tigers & Rampaging Robots
- Fascinating Explorations in LOGO

Ready-To-Run Apple IIc Programs for the Now with

0 12509 08
477465

ENTER THE ELEPHANT™ SAFARI SWEEPSTAKES.

GRAND PRIZE
(1 winner)



An exciting two week adventure for two to a wild game preserve in Kenya, Africa. The trip includes airfare, luxurious accommodations, meals, tips, and taxes.

SECOND PRIZE
(25 winners)

A Bell & Howell 35mm camera. The 35J complete with fine Lumina lens completely eliminates complicated focusing.

FIRST PRIZE
(5 winners)

A Deluxe Camping Package featuring an 8′ x 10′ Wenzel Cabin Tent, four Wenzel sleeping bags, plus a Coleman lantern, stove and cooler.

THIRD PRIZE
(100 winners)

Camouflage Nylon Duffle Bag. This handsome bag is water repellent and double reinforced at all stress points.

And thousands of Elephant Safari camouflage T-shirts featuring the Elephant logo.

HOW TO ENTER

No purchase necessary. Just come into a participating Elephant Safari Sweepstakes dealership where you'll find free entry blanks and official rules. While you're there, check out our full line of quality Elephant memory disks and accompanying products. Entries must be received by July 31, 1984. Void where prohibited.

Dennison

For the Elephant dealer nearest you, call 1-800-343-8413. In Massachusetts, call collect 617-769-8150.

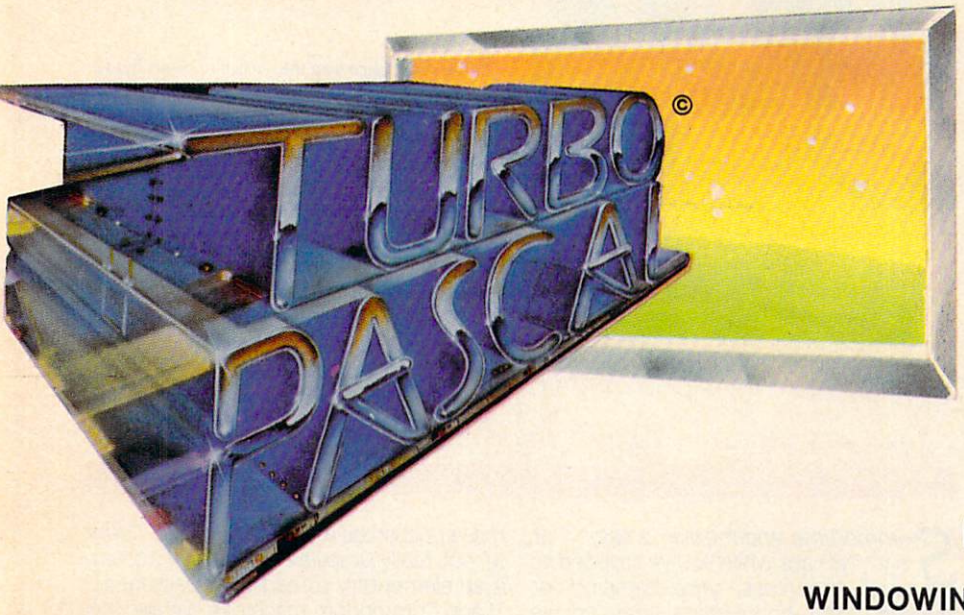
ELEPHANT MEMORY SYSTEMS™

ELEPHANT NEVER FORGETS

Circle 1 on Reader Service Card.

ANNOUNCING . . .
VERSION 2.0

EXTENDED PASCAL FOR YOUR IBM PC, PC jr., APPLE CP/M, MSDOS, CP/M 86, CCP/M, OR CP/M 80



NOW . . .
WITH
WINDOWING
$49.95

"What I think the computer industry is headed for: well documented, standard, plenty of good features, and a reasonable price."

Jerry Pournelle,
Byte, February 1984

"The Perfect Pascal"

Alan R. Miller,
Interface Age, January 1984

If you already own Turbo Pascal version 1.0, you can upgrade to 2.0 for $29.95. Just send in your old master with your check. (Manual update included of course).

NEW FEATURES

WINDOWING!

. . . This is a real shocker. On the IBM PC or PC jr. you'll now have a procedure to program windows. . . . Any part of the screen can be selected as a window and all output will automatically go to this part of the screen only. As many windows as you please can be used from the same program.

AUTOMATIC OVERLAYS!

. . . No addresses or memory space to calculate, you simply specify OVERLAY and TURBO PASCAL will do the rest.

GRAPHICS, SOUND AND COLOR SUPPORT

. . . For your IBM PC or JR!

FULL HEAP MANAGEMENT!

. . . via dispose procedure.

OPTIONAL 8087 SUPPORT!

. . . Available for an additional charge.
If you have a 16 bit computer with the 8087 math chip—your number crunching programs will execute up to 10X faster!

ORDER YOUR COPY OF TURBO PASCAL VERSION 2.0 TODAY

For VISA and MasterCard orders call toll free: **1-800-227-2400 x968**
In CA: **1-800-772-2666 x968**

(lines open 24 hrs, 7 days a week)
Dealer & Distributor Inquiries welcome
408-438-8400

CHOOSE ONE (please add $5.00 for shipping and handling for U.S. orders)

_____ Turbo Pascal 2.0 $49.95
_____ Turbo Pascal 2.0 with 8087 support $89.95
_____ Update (1.0 to 2.0) Must be accompanied by the original master $29.95
_____ Update (1.0 to 8087) Must be accompanied by the original master $69.95

Check _____ Money Order _____
VISA _____ MasterCard _____
Card #: _____
Exp. date: _____ Shipped UPS

BORLAND INTERNATIONAL

Borland International
4113 Scotts Valley Drive
Scotts Valley, California 95066
TELEX: 172373

My system is: 8 bit _____ 16 bit _____
Operating System: CP/M 80 _____
CP/M 86 _____ MS DOS _____ PC DOS _____
Computer: _____ Disk Format: _____
Please be sure model number & format are correct.
NAME: _____
ADDRESS: _____
CITY/STATE/ZIP: _____
TELEPHONE: _____

California residents add 6% sales tax. Outside U.S.A. add $15.00. (If outside of U.S.A. payment must be by bank draft payable in the U.S. and in U.S. dollars.) Sorry, no C.O.D. or Purchase Orders. D5

Circle 2 on Reader Service Card.

HOME COMPUTER magazine

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 "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.

Membership in ABC applied for and pending.

Publisher/Editor-in-Chief	Gary M. Kaplan
Executive Editor	David G. Brader
Managing Editor	Walter Hego
Associate Editor	Wayne Koberstein
Sr. Technical Editor	William Balthrop
Technical Editors	Tom Green, Robert Keller, G.R. Michaels, Stephen A. Nelson, Patricia Swift, Roger Wood
User Group Editor	Judy Campbell
Assistant Editor	Dana Campbell
Program Translators	Steve Cordon, Ann Dahm, Mark Edwards, Scott Kindt, Robert Paschelke, Randy Thompson, Lee & Josh White
Contributing Editors	Michael Brownsworth, William M. Goodman, Henry Gorman, Jr., Richard Haller, S. T. Holl, Roger Kirchner, Ted Martino
Editorial Assistant	Rhea Grundy
Production Manager	Norman Winney, Jr.
Creative Director	Gei-Lei Gom
Photography	Brion Marquez, Nelson Stevens, K.D. Wainsworth
Production Assistant	Rachel Knight
Administration & Finance	Pat Kaplan, Bob Karau
Accounting	Mark Andersen, Susan Stringer, Dorothy Dwyer
Customer Relations	Tel. (503) 341-1029 — Sharon Hinshaw, Denise McGuire, Kaine Thompson
Dealer Sales & Distribution	Tel. (503) 341-1032 — Ann Crenshaw, Michael Flagg, Paula Hoiland
Advertising	Tel. (503) 341-1049, 1050, 1053 — Irene Alderman, Lynn Ballheim
Main Switchboard	Tel. (503) 485-8796

Outside HCM

What better place than the kitchen to depict modern home productivity? Appliances like the food processor have revolutionized home cooking, turning it from tedious work to something more like play. Today, computers are pulling off a similar revolution in the home. But instead of processing food, the home computer processes information—freeing you for other more creative tasks, or for play. With this handy appliance, *you* do the cooking—and *Home Computer Magazine* provides the recipes.

Inside HCM

Summertime, and the livin' is easy . . . or is it? Perhaps while you've stopped to smell the roses, gone camping, or launched your home-made boat, you've been ignoring something. How about that pile of paperwork—steadily growing *inside* the house while you play *outside*? Before summer slips away, you may have to interrupt your fun to take care of business. Wouldn't it be nice if there was an invention that saved time, energy, and—most of all—work? Fortunately, there is. It's called the home computer.

Playing with computers is fun—but if you want to enjoy the remaining summer daze, it's time to put your computer to work. This issue of *Home Computer Magazine* comes just in time to get you on the road to self-efficiency. Here you will find our own programs designed to increase your personal productivity—plus tutorials and informative reviews to guide you in making those crucial software purchases.

But before you spend a lot of money on, say, a super-powerful spreadsheet program, take a look at what a simple one can do. *Snap-Calc* is yours for the time it takes to key it in—with complete versions for all popular machines, including the brand new Apple *IIc*. When you've tired of staring at figures, our *Bars and Plots* program will let you plot your expenses visually—just type in the numbers and see them graphically displayed as colorful bars.

No matter what computer you own—Apple, Commodore, TI, or IBM—you will find reading our reviews quite productive. This month *HCM* looks at four different word processors, ranging from the powerful-but-RAM-guzzling *EasyWriter II* to the compact and adaptable *Personal Editor* (both for IBM). *EasyScript* for the Commodore 64 proves its worth to a veteran writer. And for the TI faithful, we take a look at the *Companion* word processor.

Home productivity depends on wise and informed financial decisions. And we examine three programs to help you navigate those uncharted monetary streams. *Count-Sil*, not really Dracula's blood-thirsty cousin, is an elementary spreadsheet package for TI and Commodore machines. We also look at the best-seller *Home Accountant* for Apple, IBM, and Commodore, as well as *Home Budget jr.* for the IBM family.

All work and no play can make your computer a dull toy—so this issue packs in plenty of fun as well. *Colorfun* tops the list of key-in programs tailored to the very young. A somewhat older group can learn early math skills the graphic way with *Elementary Addition and Subtraction*. And more advanced students can rocket ahead with *Missile Math* multiplication.

All games are, in a sense, *learning* games—and we try to enhance this aspect whenever possible. How about a colorful graphics adventure that teaches you about the inner workings of your computer's *Boolean Brain*? Or do you feel like embarking on an Indiana Jones-type odyssey in the steaming jungles of the *Wild Kingdom*? We also present *Cyber-Cipher*, a "golden oldie" with a new computer twist, and *Speeder*, a deceptively simple action game. Back by popular demand, a new version of *Robochase* is now here for the Apple, Commodore, and IBM PCjr. And during breaks in game-*playing* take time to learn some game-*building*, and discover why *Programming:* (is) *The Name of the Game*.

Beyond gaming, there is fun in creativity. Those of you with an artistic eye can weave multi-colored webs on your computer screen with *Spider Graphics*. You can also creatively explore other "natural" phenomena in this month's LOGO Times section: Venture into the *Binary Forest* and learn the recursive elements of growing colorful trees; or drift through some delicate pattern designing (but don't get snowed) with *LOGO Flakes*.

So, it's *Summertime, and the livin'* is *easy*—with a little help from your home computer.

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

On Screen

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

"Etched into the granite of a Washington D.C. government building . . . is the succinct message, 'Past is Prologue.' Nowhere is this statement more appropriate today than in the home computer industry."

By the time you read this, the Summer Consumer Electronics Show (SCES) in Chicago will already have come and gone, but its indelible imprint will be with us the rest of the year. For this is the most important trade show of the home computer industry—the place where buyers worldwide get a chance to preview new hardware, software, and accessory products that will be appearing (it is hoped) on Holiday shelves during the last quarter of the year. So, as I sit here writing this editorial (in advance of the show), I'm naturally curious about what this summer's Windy City extravaganza will bring . . .

Etched into the granite of a Washington D.C. government building—I now forget which one—is the succinct message, "Past is Prologue." Nowhere is this statement more appropriate today than in the home computer industry. As a veteran of more computer and electronics shows than I care to remember, I can now see that there is finally enough "history" behind us to provide a clearly printed playbill to the second act of the Christmas marketing drama soon to unfold on the silicon stage.

Starring in the lead roles are, of course, the industry's "Big 3"—Apple, Commodore, and IBM. No longer content to win only *hardware* roles—no matter how lucrative—each of these players will now start to aggressively seek the smaller-but-more-numerous *software* roles that many other less-renowned character actors have been picking up all along. And all this *without* having to resort to building a large installed base at fire-sale prices, in hopes of making up lost profits through after-market sales.

For the third-party players, the second act will be fraught with caution and re-casting. The well of venture capital funds that has financed many a Broadway production in the past—only to have the "Golden Boys" and "Fair Ladies" close shortly after opening night—has started to run dry. Surviving players now have to financially support their own professional activities, pay ever-increasing publicity fees to gain recognition, and reluctantly accept lower price-point remuneration under competitive pressure.

The time is now ripe for employing the winning tactics of other consumer industries. So expect to see "brand name recognition" tie-ins go way beyond the past's highly volatile book, movie, and arcade blockbuster associations—to blue-chip link-ups with more of the Fisher-Prices, Walt Disneys, and Gillettes of this world. Scratch a household name that kids, parents, and adults in general can trust, and you'll be sure to find a computer industry marketing guru camped out on their doorstep shaking a freshly drawn-up joint-venture contract. Just in the burger industry alone, there's potential for such ad pitches as *Where's the byte?*, *Flame-broiled software*, and *Little Macs with special DOS*.

Expect also to see tactics borrowed from the supermarket "Cereal Wars." With inadequate retail shelf space for the horde of new computer products coming down the pipeline, the present Kelloggs and Posts of the home computer industry will protect themselves from encroachment on their hard and soft turf by "homesteading" all the temporarily vacant tracts with competitive (to themselves!) product lines. (Did you really think that all those different breakfast foods were put out on the shelves so we could have plenty of *choice*?) There will be far less "vaporware"—products that are announced with great fanfare, only to die stillborn. Instead, we'll see new products be quickly born and quickly put out to pasture (if necessary), with each generation rapidly replaced by offspring waiting in the wings for that valuable shelf space. It pains me to say that this "cereal tactic" has already been imported by the computer publishing industry. Software and hardware are next, my friends.

Enter the dragon. Inscrutable Japan Inc. is finally ready to flex its samurai muscles. Until now, we've only been *hearing* about the MSX software standard for home computers among the Japanese consumer electronic giants. But now that the American computer market is more mature, and the major price wars are over, its time for Tokyo to use its collective consumer-appliance clout to promote "integrated electronic home systems." Expect to see color TVs with home computers *built in*, and home computers with VCR and video disk interfaces *built in*. By 1985, Japan will be offering fully integrated systems—with the compact laser disk migrating over from stereo music to data storage—including everything but the kitchen sink (unless, of course, the sink happens to be a Sony Washman).

Getting back to Act Two of our soon-to-be prime-time marketing drama, the "herd instinct" is unfortunately starting to overpower the industry at large. After all, the proverbial 40 million Frenchmen—from the country that gave us Bardot and Perrier—can't be wrong . . . Or, can they? I'm alluding to the powers-that-be in the software and peripheral industry who are exclusively chasing after the presently small, installed bases of new 128K machines—the *IIc*, Mac, and PCjr—and ignoring the over 5-million VICs, C-64s, and 99/4As that have been sold. There's a golden opportunity here for players who aren't blinded by the lights and lure of Broadway, and are ready to accept the less-flashy-but-steady roles that the summer-stock theater has to offer.

Letters to the Editor

Dear Sir:

Just received my issue of HCM and it's great. I am glad to see you included other top computers in your magazine. Maybe now we will see some cooperation between the computer literate of the other brands.

I was one of the first subscribers of 99'er when it was in it's infancy and I'm glad to see it grow as it has.

I now am the present owner of an APPLE II+. I would now be lost had I not gained the experience from programming my TI-99/4A. The Apple manuals were not as informative as the one I got with my TI-99/4A. Because I couldn't afford a disk right away I spent hours converting TI programs to Apple. Boy, the looks I got whenever I would ask for some help from other Apple users on how to change TI stuff to Apple. I hope this attitude toward users of different systems stops so we can all learn from each other's experiences.

Joseph C. Manning
Bloomsburg, PA

Thank you, Joseph, for your kind words about HCM—they are greatly appreciated. We tend to share your hopes about the cross-pollination of ideas between the users of different brands of computers. After all, it is the same melting-pot concept that has made America great in the first place, right?

An Apple Greeting

Dear Sir:

The program listing with this letter is a HELLO program that I made, and wanted to know if you would like to print in your fine magazine. If you decide to print it, the next paragraph explains what it does. By the way, the program is for an Apple computer.

This HELLO program makes it possible to press one letter such as A instead of typing out RUN HELLO. The program also locks, unlocks, loads, and deletes programs. Good luck and I hope your readers find it useful.

```
10 HOME
20 REM BY MAURICE PATRICK TESSIER
30 REM 28 MONTGOMERY ST.
40 REM SAUGERTIES,N.Y. 12477
50 REM FOR APPLE IIE
60 PRINT CHR$ (4);"NOMONOC,I,O"
100 TEXT : HOME : D$ = CHR$ (4): PRINT D$"CATALOG": B = PEEK (37) - 2: IF B > 22 THEN B = 22
110 T = 0: CH = 4: FOR CV = 0 TO 23: GOSUB 1000: IF C < > 160 THEN POKE P - 1,219: POKE P,193: POKE P + 1,221: T = T + 1: S = CV
115 VTAB 1: HTAB 17: FLASH : PRINT "PAT'S DISK": NORMAL
120 NEXT CV: VTAB 24: A$ = "*TYPE LETTER TO RUN, OR LOAD=1,LOCK=2,UNLOCK=3,DELETE=4,EXIT=5"
130 B$ = "RUN": HTAB 1: PRINT LEFT$ (A$,39);: A$ = MID$ (A$,2) + LEFT$ (A$,1): K = PEEK ( - 16384): IF K < 128 THEN FOR K = 1 TO 75: NEXT K: K = FRE (0): GOTO 130
140 POKE - 16368,0: K = K - 176: IF K < 1 OR K > 5 THEN 300
200 HTAB 1: CALL - 868: IF K = 5 THEN END
210 PRINT "PRESS 'LETTER' YOU WISH TO";: IF K = 1 THEN B$ = "LOAD"
220 IF K = 2 THEN B$ = "LOCK"
230 IF K = 3 THEN B$ = "UNLOCK"
240 IF K = 4 THEN B$ = "DELETE": FLASH
250 PRINT B$;: CALL - 198: NORMAL : GET K$: K = ASC (K$) - 48
300 IF K < 17 OR K > T + 16 THEN 130
310 CH = 1: CV = K - 16: GOSUB 1000: IF C = 194 AND (B$ = "RUN" OR B$ = "LOAD") THEN B$ = "B" + B$
320 FOR CH = 6 TO 39: GOSUB 1000: B$ = B$ + CHR$ (C): NEXT CH: HTAB 1: CALL - 868: PRINT D$;B$: GOTO 100
1000 C1 = INT (CV / 8): C2 = CV - C1 * 8: P = 1024 + 40 * C2 + 128 * C1 + CH: C = PEEK (P): RETURN
```

Maurice P. Tessier 12477
Saugerties, NY

Thanks, Maurice, we typed it in and found that it works just fine. Any other readers out there having short and useful routines are welcome to share them through this column too.

Commodore Bulletin Board Revealed

Dear Sir:

I have been a staunch supporter of your magazine for well over a year. Now that your magazine includes coverage of the Commodore 64 your readers may be interested in the following. I am running an electronic bulletin board that I helped write using the Commodore 64 and Commodore 1650 automodem. Since I am using my home phone, I am currently only operating from 9 p.m. until 12 p.m. on Friday, Saturday, and Sunday (Central time). Features of the board include such things as download capability and remotely written menu file capability. Commodore owners and others may wish to check in and look around or page me and I'll be happy to chat with them. Set your modem to 300 baud, full duplex, 8 bit word length with one stop bit, no parity and modem set to originate. The number is 601-327-5062.

As an owner of two TI-99/4A's, a VIC-20, and a Commodore 64, your magazine is a welcome addition to our household. Keep up your excellent work.

Dave Barron
Columbus, MS 39702

Thanks for the information on your home-brew computerized bulletin board, Dave. I am sure that other home computer users out there will be calling to chat with you.

Four home computers! That's quite a cache. Glad you enjoy HCM, and we hope that you also enjoy the conversations with the many users that will be contacting you on your bulletin board.

Dear Sir:

I purchased *Home Computer Magazine* after becoming a recent owner of a Commodore 64.

I really enjoyed the programs for the C-64 and even more the informative articles that go much deeper into different areas of computing than could be accumulated in reference books.

It disappoints me when I see a program written for another computer that I really want. I enjoy seeing articles comparing the different BASIC languages.

Steve Kitchens
Decatur, IL 62521

Steve, we understand your disappointment when you see a program written for one machine that will not run on yours. That is why we try to put every program (where feasible) on each of the machines that we cover—if not in the same issue, then in consecutive issues.

Dear Sir:

I was pleased to find your magazine on the newsstand recently, having noticed your former publications in the past. I was always intrigued by your 99'er Magazine. However, since none of it applied to my system I really did not take the time to read an issue. Let me say I am quite pleased with your expanded publication and as proof enclose my check for a subscription. These are the things that particularly appeal to me:

1. Variety of typestyles—interesting look.
2. Articles regarding my C64 scattered throughout the publication.
3. Mixture of applications—home and games.
4. Reviews.
5. Easily-read listings mixed with text instead of being grouped at the back of the magazine.
6. Side-by-side articles for other sytems aid in understanding other systems.

I feel your publication is as good as most of the Commodore-only magazines I've read. Keep up the good work!

Jim Colbutt
Richardson, TX 75081

We really appreciate this kind of feedback, Jim. By listening to our readers, we try to change and improve publication of HCM. It is particularly interesting to note your item 5 regarding the listings being mixed with text instead of grouped together in one spot. We have seriously listened to the readers that have responded and the overwhelming majority have asked to have all the listings in one spot so it is easier for them to key-in the programs. We hope that you won't be too unhappy to see, Jim, that in this issue we have therefore started grouping the listing in the center of the magazine.

Dear Sir:

In Volume 4, No. 1 you published a letter from Jeff Strong on C-64 keyboard buffer problems. You suggested he clear the keyboard buffer using two lines of code. There is an easier way—using a single line of code:

100 POKE 198,0

This clears the buffer completely.

Matthew Leeds
San Francisco, CA 94121

Thanks Matt for showing us the easier way. Sharing this kind of information with fellow readers is well appreciated. Many readers tell us the first things they read in each issue are Letters to the Editor and the Tech Notes so as to make sure they have not missed any tricks.

Dear Sir:

Please accept my congratulations of the best home computer magazine I have ever seen. The detailed look on the IBM PCjr stunned even battle-hardened salesmen in a local IBM store. No one has ever seen such a fine and accurate presentation.

I would like to warn TI-99/4A users that OKIDATA printers such as the MICROLINE 83A will not work with the parallel port unless one builds an interface. Okidata does not consider the TI-99/4A users as a large enough market to concern itself with, and thus no technical customer support is offered (even though it is stated in their manual). I thus would appreciate it if an article could show how to build this support so those of us that are "stuck" with the 83A can get some use out of it.

Thank you and please keep up the superior product.

Wolly Barabash
Edmonton, Alberta, Canada T5J 2L8

We are gratified, Wolly, that you enjoyed the IBM PCjr article. You are right. We have seen several "battle-hardened salesmen" that sell IBM products using that particular HCM issue to sell the IBM PCjr—and very successfully, too.

Your comments on the difficulty of Okidata printers being connected to the TI home computer are not uncommon. Anytime a printer of a different manufacturer is attached to a computer there will be several minutes, if not hours, of your time or someone else's to correctly set up and operate that combination. Once you have found the proper cable, interface, and software write it down in a log book somewhere and don't lose it. If there are any readers who have already solved the Oki-TI compatibility problem, please let us know so we can help Wolly (and others) out.

TI Pro Runs HCM Listings

Dear Sir:

We very much enjoy your magazine and originally ordered it when we had a TI-99/4A and the magazine was the 99'er. But in December I got a TI Professional computer.

Now that the magazine has changed to include IBM PC programs, we've converted a couple of HCM programs to work on the TI PC—actually only a couple of simple changes are usually required. We do have a different color code set though. It's not always clear to me where the colors are in the programs. Our BASIC manual is somewhat cryptic regarding graphics instructions. We also have pixel ranges of 0-719 for columns, 0-299 for rows so our graphics come out smaller than they would on the PC.

There may be other PC-type users who have problems similar to ours.

Anyway, keep up the good work and I hope to see more PC/PCjr-type programs and games in the magazine in the future.

Barbara Taylor
Huntington Beach, CA 92646

Barbara, you've discovered the same thing that we have about the TI Pro. Almost any program that can run on the PCjr can be very easily converted to run on the TI PRO with its color RGB monitor. The BASIC languages of the two machines are very close, although the Pro does have a few more capabilities than the jr.

Dear Sir:

I was a subscriber of *99'er Magazine* and very much enjoyed all its features. When "*Home Computer Magazine*" replaced the former publication, I was again happy to receive the first issue.

Do you know of any software (or the necessary programming tools) that would essentially be a "spell-check" program to interface with TI-Writer? That might be an interesting task for a programmer.

William Koseluk
Goleta, CA 93117

That sounds like a really great suggestion, William. Perhaps one of the two million TI *owners out there has already written a spell-check program for* TI-Writer. *If so, they should send it in to HCM and get it published. What a great piece of productivity software that would be.*

Dear Sir:

It seems as if every time someone writes an article about some language other than BASIC, a fundamental necessity is to malign BASIC. It's called "comparison." I think BASIC and LOGO have been compared unfairly more than once.

I have a spot in my programming heart for both LOGO and BASIC (and I plan to tackle FORTH and PASCAL). One thing I have learned is that each of the languages, BASIC and LOGO, have their strong points. They also both have weak points, a fact usually overlooked by authors of LOGO articles.

Recently, I was going through some back issues of HCM and I came upon the article by Henry Gorman, Jr., "The BASIC Issue and the Tortoise's Retort," page 49, May 1983.

Once again, we are reminded that BASIC doesn't allow recursion. But for all practical purposes, a FOR-NEXT loop is recursive, and whether that is a "scientific truth" is only of academic interest. To make my point, I have included a TI-99/4A Extended BASIC routine that does what Mr. Gorman suggested can't be done with BASIC: it counts the words in a list.

Of course, in BASIC many more lines are required than the four that LOGO uses. But there are BASIC routines that are shorter than the equivalent LOGO procedures. Producing a random number greater than nine is an example.

Both languages have their weak points and TI BASIC is notably weak in high resolution graphics, but in general what can be programmed in one language can be programmed in the other. (Of course, one or the other is usually better, depending on the project requirements.) I have written BASIC routines which accurately mimic the LOGO primitives BUTFIRST, BUTLAST, FPUT, and LPUT. By the way, there is no LOGO primitive which is the equivalent of TI BASIC's "XOR." An equivalent procedure can be written, however.

One final observation. In the same issue of HCM (May 1983, page 52), Professor Holl shows us, in BASIC, how to insert a word right into the middle of a list at any designated position. There's no LOGO primitive for that! A forthcoming procedure, anyone?

The BASIC routine for counting words in a list must correctly return the number of words, including zero if the list is empty. Hint: the routine must not be fooled if the list contains

Continued on next page

HCM Review Criteria

Each month, *HCM* reviews software packages for the IBM PC and PCjr, Apple II, II+ and IIe, TI-99/4A, and Commodore 64 and VIC-20 computers. These reviews take a detailed look at the quality of commercially available third-party software for these home computers.

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

- **Performance**—how well the activity responds to the player's commands; how well the sound effects, music, or speech are integrated with the software.
- **Documentation**—the quality of the printed matter that comes with the software: whether the instructions are clear and comprehensive; whether the machine configuration requirements are spelled out. Information such as how to load the program, use the keyboard, and restart the activity contributes to the documentation rating, as do tips on performance peculiarities.
- **Engrossment**—whether the game or activity has that intangible quality that holds the player on the edge of his seat while the hours tick by unnoticed.
- **Ease of Use**—the degree to which a user can interact with the software 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.

Education-Specific Criteria

Educational software may also be evaluated in the following areas:

- **Concept Presentation**—whether the concepts are presented clearly, in logical order, and in enough depth for the learner to be able to apply the learning to other situations.
- **Rewards**—whether the audio-visual rewards are motivating and whether they are appropriate to the activity.
- **Graphics**—rates the quality of the graphics and whether they enhance or detract from the educational purposes of the activity.

Letters to the Editor

. . . from p. 7

extra spaces between the words, or at the beginning or end of the list.

```
100 CALL CLEAR :: DIM
     CH$(255) :: X=0 :
    : COUNT=0 :: INPU
    T "LIST: ":LIST$
110 FOR C=1 TO LEN(LI
    ST$)
120 CH$(C)=SEG$(LIST$
    ,C,1) :: IF ASC(CH
    $(C))=32 AND CH$(
    C)=CH$(C-1)THEN 1
    40
130 IF ASC(CH$(C))=32
     THEN COUNT=COUNT
    +1
140 X=ASC(CH$(C))
150 NEXT C
160 IF X=0 THEN PRINT
     X :: GOTO 180
170 PRINT COUNT+1
180 END
```

P.S. In LOGO II there is a primitive, LENGTH, which returns the count of words in a list.

Gene Thomas
Little Rock, AR 72205

Touché—your points are well taken, Gene.

Dear Sir:

I just received my first issue of *Home Computer Magazine* and want to commend you on the publication of a very fine magazine. There is a touch of irony in that several months ago I subscribed to *99'er Magazine*. During that period I scrapped plans to upgrade my very basic TI-99, gathered my courage and my checkbook, and bought a PCjr. I wondered every now and then what I was going to do with a subscription to a TI-99 magazine. Imagine my delight when my first issue, Vol. 4, No. 1, showed up and I found it contained the best article I have read in any publication about the PCjr.

Of particular interest was page 43 entitled, "Options for PCjr System Configuration." I had waited to purchase a monitor because I wanted really good resolution, a 12-inch screen, and sound from somewhere other than the computer. The IBM RGBI met the first two specifications, but does not have a speaker. Other monitors I found with good sound had degraded resolution. Even a trip to the IBM Product Center failed to produce any good answer. Then your magazine arrived! Mr. Kaplan and Mr. Balthrop must be the only two people in the world who know about the external speaker port. I have shown this configuration to three IBM distributors today, including the Product Center, and they didn't even know there was an external speaker port.

Thanks very much for your help. I look forward to future issues.

G. Michael Fairley
Fairfield, CT 06432

We are pleased that Home Computer Magazine *still fills your needs. In future issues you will find more information on how to use the PCjr sound, including music. We hope that in the future you will continue to find a lot of sound advice within* HCM.

Dear Sir:

In the letter "Support Needed Down Under Too," Steven Shraibman asks how to activate the special functions of the EPSON FX-80 printer. I, too, have the FX-80, and it works nicely with TI-Writer.

If Steven wants his entire work in proportional, he can: 1) Press [CTRL U]. This sets the control character mode; a seldom-used mode because it isn't on the TI-Writer reference card. (All characters have 64 subtracted from their ASCII-value. Control characters are represented on the screen by tiny hexadecimal numbers). Then press [FCTN R] (an ESCape; represented by a tiny 1b), [CTRL U] (toggles the control character mode off), "p" (lower-case P) and then a "1" (numeral one). These keystrokes should appear as 'bp1 on the monitor. These codes must be sent to the printer before the text. Putting it at the beginning of the file will accomplish this. The formatter will send them to the printer like any other characters.

2) He could also use the transliterate command of the TI-Writer formatter to define characters that turn the proportional spacing on and off. For example, .TL 123:27,112,49 and .TL 125:27,112,48 will toggle the proportional mode by surrounding the text with braces {and}. That is, "*Home {Computer} Magazine*" would have the word "Computer" in proportional but neither of the other words.

Italics can work similarly by surrounding the text to be italicized with braces so it comes out in italics. To do this define 123 as 27,52 and 125 as 27,53. I chose 123 and 125 (left and right braces) arbitrarily. For me, they are unused characters. Any little-used character(s) could be used. Italics and proportional can be set at the same time also as evidenced by this letter which uses proportional, italics, and graphics; all out of TI-Writer.

When using the proportional mode, do not use the formatter commands & (underline) and @ (overstrike). TI-Writer counts columns and the columns do not line up in proportional mode. The underline will come later in the line than you wanted, so use the FX-80's own underlining and double-striking.

Please pass this along to Steven or publish it, so more users can utilize TI-Writer to its fullest.

Glenn Davis
St. Paul, MN 55117

Thank you very much, Glenn, for those tips on the use of TI-Writer with the Epson FX-80 printer. You are one of the few bold people to dig deeper into TI-Writer's capabilities including the formatter. Perhaps your explorations will encourage others to make discoveries that they too can share with our readers.

Pocket Canon Plays the Pops

Dear Sir:

I really enjoyed the Pocket Canon in your LONG awaited Vol. 4, No. 1 magazine. One thing I did find annoying was the "POP" of the canon. To do away with this unwanted noise I simply changed the duration in line 230 to a negative number. Changing this to a negative number causes the previous sound to stop and the new one to start immediately. This could be a problem in some applications, if there wasn't enough delay between two sound statements it would cut the duration of the previous sound short. But since the Pocket Canon program has a built delay of approximately 500 milliseconds it creates no problems in the program.

Your magazine is very much appreciated. Being from a small town it is one of the only ties I have with the outside "TI computer world." I do hope you will include articles on the newly released TI-FORTH.

Timothy Maes
Beeville, TX 78102

Gee whiz, Timothy, we thought that all cannons were supposed to pop. If not even "boom." Seriously though, we would love to receive articles on TI-FORTH *or any of the other* FORTH *implementations, particularly of a beginning nature.*

TI Bulletin Board!

Dear Sir:

I would like to find out more about the TI-99/4A computer's power in the communications line—especially the BBS. I would like to find out if there is anyone writing the programs for a BBS and if so, I want to find out if they are compatible with other BBS. I mean can you upload and download programs over the telephone lines? Is there someone I can get in touch with that can give me that information?

I have been calling all of the BBS I can find in my area and I have found no TI systems around. Do you people know of any? If not in my area, some other locale?

Paul Reinhard
Bellflower, CA 90706

Paul, a few months ago we heard that the Chicago Users Group was about to go on-the-air with a bulletin board system for Texas Instruments and we have heard of others that were starting up. As an example we would like to refer you to the following letter.

Dear Sir:

I don't know if you put these kind of things in your magazine or not, but way over here on the east coast there is a BBS called The *99'er Bull Board* and it operates 24 hours. The number is (301) 434-0117 and it has lots of interesting information for the TI users.

Phil Simerly
Silver Spring, MD

Thanks for the information, Phil. Perhaps one of the TI bulletin board system operators would like to share with us how to actually go "on-the-air" with the TI machine.

New Dimensions For His Apple

Dear Sir:

A few months ago I purchased an Apple IIe system for use in my home business and it has worked well for me. Recently while visiting a computer dealer, I saw *Home Computer Magazine* on his bookshelf. Having some time I looked through the magazine and noticed the article entitled "3-D IIe" about three dimensional graphics on the Apple IIe. I found that article very interesting since I have not considered using the Apple II for anything except business before reading your magazine. Now, after hours, my oldest son spends time keying-in game programs from *HCM*. Your magazine is definitely worth the cover price, but I'm going to subscribe to save money anyway.

Grant Withers
Royal Oak, MI

Well, Mr. Withers, I'm glad to see that reading HCM *has taught you to not take your Apple computer for granted. We hope that you enjoy the other exciting articles that you'll see in the next few months.*

HCM

CP/M®

PROCESSOR FOR THE TEXAS INSTRUMENTS 99/4A

NOW YOUR TI-99/4A REALLY MEANS BUSINESS!

Morning Star Software introduces the CP/M Processor for the TI-99/4A. Now the advantages of the most accepted business programs are yours. With the CP/M Processor, your TI-99/4A is ready to enter the business world.

The Morning Star Software CP/M Processor fits into your peripheral expansion system like any expansion card. It's as simple as that. Once the CP/M Processor is activated, you have all the advantages of two processors: The CP/M Processor handles all the program computations, while the TI Processor channels the information to the screen, disks and printers. This gives you the flexibility you've always wanted.

CP/M is the industry standard business format for computers today. CP/M offers you the advantage of working with programs that were designed to meet the needs of the professional business world. Powerful commands, time saving functions and proven methods are the result.

The CP/M Processor by Morning Star is built to take advantage of both the TI computer and CP/M technology. The CP/M Processor contains an 8085 CPU which, with the 16-bit TI-9900 CPU, translates to tremendous processing power. Your TI-99/4A combined with the CP/M Processor becomes a true business tool at a fraction of the expense of a new computer system.

- *Needs no modifications to the TI-99/4A system.*
- *64K RAM, 8K ROM.*
- *Makes additional use of the TI 32K memory expansion.*
- *Supplemental installation and usage manual included.*
- *Contains sufficient memory for virtually all programs.*
- *Compatible with TI-99/4A storage diskettes.*
- *Provides 60K transient program area.*
- *Fully burned-in and test cycled to ensure reliability.*
- *Fully shielded and protective metal enclosure.*
- *Ultra-reliable gold connector assembly.*
- *No chip sockets for better reliability.*
- *90-day warranty.*
- *CP/M-80 2.2 disk-based operating system included.*
- *Digital Research™ CP/M user's manual included.*
- *Access to thousands of commercially available software packages.*

THE MORNING STAR CP/M PROCESSOR

$595

TO ORDER: Check, Money Order, Master Card, VISA, and C.O.D. orders accepted. Add $5.00 shipping and handling. Dealer inquiries welcomed.

COMPATIBLE CP/M® SOFTWARE

The Morning Star CP/M processor is the only way the TI 99/4A will run CP/M based programs. The programs have been selected and re-formatted to run with the Morning Star CP/M processor. As Morning Star reviews CP/M packages, we will continue to expand our software selection. The programs we offer are the products which demonstrate outstanding values. They have met our criteria of being extremely user friendly, comprehensive in content, plus make the best use of the CP/M card's capabilities. They also represent the tasks most needed to be covered by a business computer.

BUSINESS MASTER® PLUS—Improved
Now enjoy this comprehensive business package without having to buy CBasic Interpreter! Designed to meet the needs of leading businesses. This package includes diskettes for General Ledger, Payroll, Fixed Assets Accounting, Accounts Receivable, Accounts Payable, Inventory, and Mailing List programs.
Regularly $495;
Introductory Price **$295**

SUPERWRITER™ by Sorcim
This is the word processing program you've been waiting for. Designed to be easily understood in ten minutes with straightforward language and screen prompts. Written specifically for the business person with word processing requirements. It provides all key features of a professional text editor. It incorporates form letters, report generation, list maintenance and even a spelling checker, the award winning SpellGuard™. Similar features cost as much as $900. SuperWriter™ only **$295**

PERSONAL PEARL™ Data-Base Manager
*Now easily create data systems for business, accounting, marketing, office management, education, and more. A complete user's manual, tutorial disk and screen menu guides even novices through the most sophisticated data base manager available. Unlimited files, forms and reports can be created. **Info World** reported "Personal Pearl offers you the strongest, easiest to use data-base manager I have seen at any price".* **$295**

THE RANDOM HOUSE PROOF READER
20,000 word Random House dictionary. Displays spelling errors in context and can make spelling suggestions. You edit the word. Corrections are automatically checked and substituted into the document. File new words into the dictionary with a keystroke. **$50**

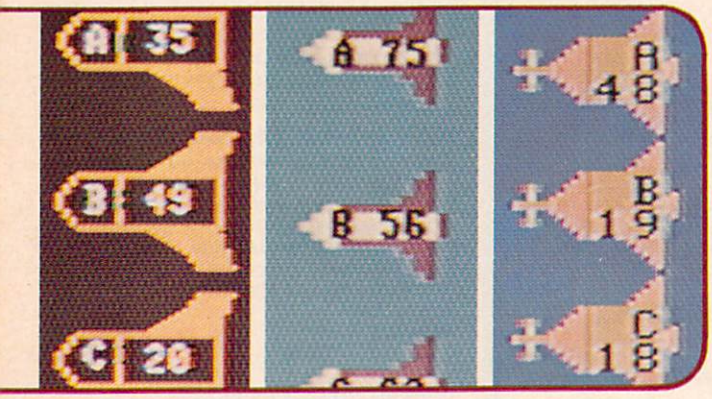
CBASIC® INTERPRETER
The language of choice for many CP/M compatible software packages. This brings you a variety of established and time tested applications and allows you to write your own CP/M compatible programs. **$150**

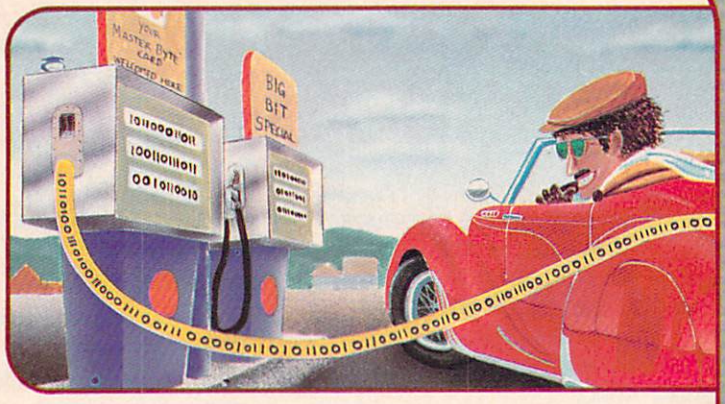
MORNING STAR SOFTWARE

4325 S.W. 109th Ave., Beaverton, Oregon 97005. Toll-Free 1-800-824-2412, in Oregon 503/646-4695, TEX NET NO. TI3416

Peripheral Requirements: Peripheral Expansion System, disk, disk controller 32K RAM card.

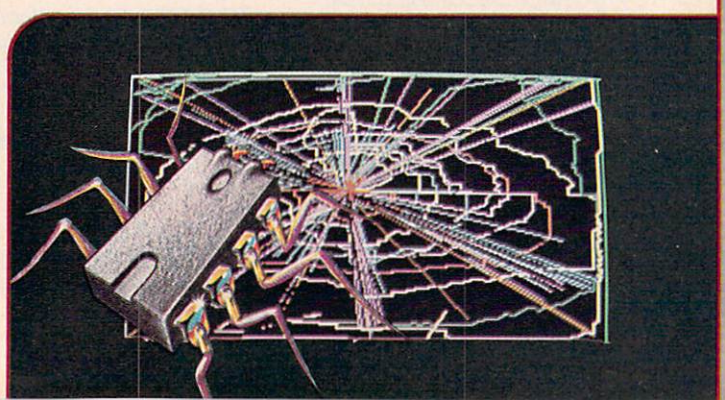
HOME COMPUTER™ magazine





FEATURES

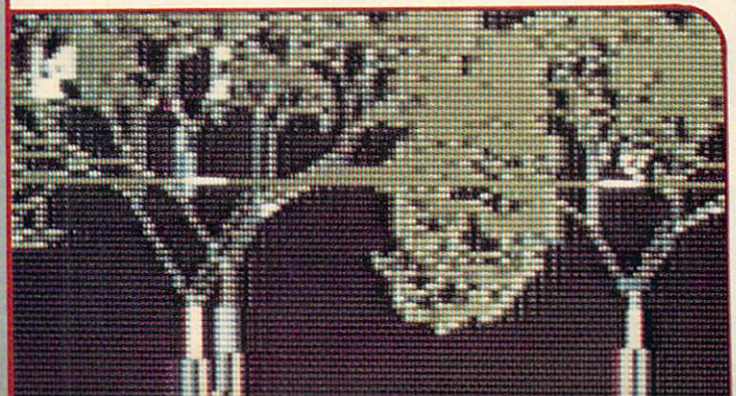
PRODUCT REVIEWS



CONTENTS

AUGUST, 1984

LOGO TIMES™



GAMEWARE BUFFET™

DEPARTMENTS

Snap-Calc™

The spreadsheet program is a handy tool for doing many financial calculations simultaneously. Its inherent versatility turns a computer into a super "What if . . ." machine. Using very simple commands, you can easily customize a spreadsheet for your own applications. These could include tracking your investments, evaluating alternative purchase options, calculating your company payroll, or tabulating accounts receivable. *Snap-Calc*™ is such a spreadsheet program written in BASIC.

The *Snap-Calc*™ spreadsheet itself is a matrix of numeric values. After you have entered values into the program, it will execute calculations that you have previously specified, and display the results through screen windows.

With *Snap-Calc*™ you will be able to create a formula for each row of the matrix. This same formula will then be used for all of the columns. You may also create names for each of the rows, and indicate which rows are to be included in the printed reports. (You may want to use certain rows to calculate other data.) In addition, you can specify the number of columns the spreadsheet will have, and set up a *totals* column which sums all columns used. The photo at the right is an example of the data entry screen.

DATA ENTRY IS A SNAP

To make an entry in any field, simply move the cursor to the field that you wish to change and type in the new value. A second set of cursor keys lets you move three columns left or right, or five rows up or down at one time. Not only is the screen updated with every key press, but the current value on the screen is stored in the spreadsheet. You can leave that entry field at any time, and the value displayed will be saved automatically.

LOGIC ENTRY MODE

When you select this mode, the screen will clear and a prompt > will appear at the bottom of the screen. At this point, you can enter logic commands. These commands will let you specify the *totals* column and *last* column in the spreadsheet. You can also assign a logic name to your model, as well as row labels. The

by Gary Strauss
and the HCM Staff

most powerful aspect to this option is that it lets you define equations for any of the rows in the spreadsheet. (See **r = eq** below.)

Two types of commands are used in this option: immediate action commands, and logic function commands. The immediate action commands are as follows:

NEW When this command is used, the memory will clear and you will be prompted to enter one of four suboptions. If you select Data, then all of the data entered on the spreadsheet will be cleared. If you select Logic, your current logic model will be erased from memory. If the third suboption—Both Data And Logic—is selected, both data and the logic model will be cleared from memory. The fourth suboption, Abort New Command, is an escape which allows you to return to the normal logic entry screen without clearing any memory.

ROW	NAME	1	2
1	COST 1	10.00	20.50
2	COST 2	25.00	35.75
3	COST 3	37.15	12.00
4	TOTAL COST	72.15	68.25
5	5	0.00	0.00
6	6	0.00	0.00
7	7	0.00	0.00
8	8	0.00	0.00
9	9	0.00	0.00
10	10	0.00	0.00

This command should not be confused with the BASIC command NEW, which erases the program in memory. Exercise caution when using this command. Always save your data and logic model to disk first.

LIST This command will list the logic model on the screen. The first item listed is the logic name. If no logic name has been assigned, then the message LOGIC NAME IS with no name following it will be printed.

TOTAL COLUMN IS col is always on the second line; here col is the totals column. If col is set to 0, then the totals column has been turned off. The default value for the totals column is 13.

LAST COLUMN IS col is always the next item listed and col is the last column in which data entry can occur. This value must always be less than the total column value. The default for this value is 12.

From this point on, the information listed is determined by the logic commands that you have entered into the model. If you created a row label for a row, the message r IS nnn will list, where r is the row number and nnn is the name of the row. Even if you entered the row name in the data entry section of the program, it will be displayed here. Following the row name will be all equations defined for that row. The message r = eq will list, where r is the row number and eq is the equation for that row.

PRINT This command performs the same functions as the LIST command except that the output is directed to the system printer.

MOVE xx TO yy This command allows you to move all of the data from column xx into column yy. You would then have duplicate entries—the information in column xx is not erased after the transfer.

END This command will cause you to exit the Logic Entry mode and return to the Data Entry mode. You may at any time return to Logic Entry mode and append changes to your logic model.

The following logic function commands affect the logic model itself:

LOGIC NAME IS nnn This command lets you create or change the name of the logic model, where nnn is the name of your logic. LOGIC NAME IS SAVINGS, for example. The name you assign to your logic model will be used as the file name when you save your model to disk or tape.

TOTAL COLUMN IS col
or
TOTAL COLUMN IS OFF This command lets you create or change the column that you want to use as your totals column. If the Off option is used, then the totals column will be set to zero and will not be calculated when the logic model is calculated.

LAST COLUMN IS col This command creates or changes the last column in which you will be able to enter data.

Continued on next page

Logically, this should be the column just before the totals column (if the totals column is in use), because you may not move your cursor beyond the LAST column when entering data. If the totals column is several columns beyond that point, you may not be able to display it on the screen.

The only restriction in designating the last column is that it must be a lower column number than the totals column, unless of course the totals column is turned off. In that case you can make the last column any value your machine will allow. The computer system you use will determine the maximum number of columns available.

r IS nnn This command is used for naming any of the rows in the spreadsheet. The row number is r and nnn is the name you assign to the row. An example might be: 3 IS ROW#3, where the name ROW#3 is given to row number 3.

r = eq This is by far the most powerful of the logic model's commands. It lets you assign an equation eq to a row r. The equation can be made up of other row numbers; real numbers called constants; an operator called LAG, which lets you access a previous column; or arithmetic operators. Rules for constructing the equation are as follows:

(1) A row can be specified by simply using the row number in the equation. For example 5 = 1 + 2 means that the value from row 1 is to be added to the value in row 2 and the total is to be placed in row 5. This same equation format is then used for all of the columns in the spreadsheet. Notice that there are spaces between each item. This is a requirement of the system so that the computer can decipher it easier, and execute it more quickly.

(2) A real number can be used as a constant in the equation when placed within parentheses. Taking the example from the previous paragraph, let's try: 5 = 1 + 2 * (12.53). The equation will now take the sum of row 1 plus row 2 and multiply it times 12.53. There is no priority of multiplication over addition—everything is executed from left to right. Rows 1 and 2 are added together before the sum is multiplied by 12.53.

You can use data from the previous column with the LAG modifier. When this modifier precedes a row number, the information will come from the same row number, but from the previous column. For example: **5 = 1 + LAG 2**. Here row 1 from the current column will be added to row 2 from the previous column. If the column currently being calculated is 8, then LAG 2 would cause the value from column 7 row 2 to be used. If you set up an equation such as: **5 = LAG 2**, then the value of the previous column, row 2 would be placed in row 5. In another example, **5 = 4 + LAG 5**, a running total for row 4 would be created in row 5.

There are four operators that you can use in your equations. They are:

+	**Add.**
−	**Subtract.**
*	**Multiply.**
/	**Divide.**

The length of an equation is limited only by the maximum string length for each system. There are minor differences in how the equation is to be entered on each machine, so consult the *Snap-Calc*™ section which covers your machine.

Once you have entered one of the above logic function commands, you can use the LIST command to check whether the command has been received and interpreted properly, or to simply check logic statements that have already been entered.

CALCULATE MODEL

When the key for the Calculate Model function is pressed, the computer will begin calculating your logic model using the data currently entered on the spreadsheet. All of the equations for each row will be executed in ascending order of row number. For example, if an equation in row 3 uses the result of an equation in row 5, you may have a problem. Because you can't enter data into a calculation field, field number 5 will probably be set to zero and will not contain the proper information when row three does its calculation. You should design your logic model with this in mind.

In addition, Calculate Model provides an automatic row total in the totals column. The totals column defaults to column 13 when the program is first run, but you may move it to a different column, or turn it off completely using logic model statements. If the totals column has been turned off with a logic statement, then the row total will not be calculated.

LOAD DATA FILE

When you select the Load Data File option, a menu of three suboptions will prompt you for the type of file to be loaded. The first suboption loads the data file only, which contains the data that has been entered into your spreadsheet. Select the second suboption if you wish to load the logic model. The third suboption lets you load both data and the logic model in one step. Once you have entered your file type, you will be prompted for the file name of the file you requested. The name you enter must reside on the data disk currently in the drive, or on a cassette tape.

SAVE DATA FILE

The primary difference between the Save Data File and the Load Data File options is in creating file names. If you choose to save the logic model in either the Load Logic Only, or Load Data and Logic modes, the file created for the logic model will use the LOGIC NAME as the name of the file. This is important to keep in mind so that when creating a logic name you only use characters that are legal in the file system being used. If the logic name has not been declared, then you will be prompted for the name of the file. That name will then become the logic name for that model when loaded at a later time. If you no longer want to keep a logic name, or wish to create a modified version, a logic name can be changed at any time.

CLEAR FIELD/ERASE

The Clear Field and Erase option can be used to back out of an entry and clear the field, or to simply reset a field back to zero. To use this function, place the cursor over the value that you wish to set to zero and press the key associated with this function. If the field is numeric, it will be redisplayed with a value of zero. If the field is a row label, it will be erased and left blank. Erasing a label from a row does not affect the rest of the row, or any equations set up for the row.

PRINT REPORT

The Print Report option allows you to generate a hard-copy report of your spreadsheet data on your system printer. Before the program starts printing, you will be prompted for the title of the report, the date, and the maximum number of rows to be printed in the report. The report generated here is formatted for an 80-column printer. This limitation restricts the report to printing only six columns of data across the page. Since there are more than six columns in most spreadsheets, the report will print all of the rows for the first six columns, then form feed to the top of the next sheet of paper and print the next six columns. This continues until all of the columns have been printed. The report will include every column up to the column designated as the totals column (or the column designated as the last column, if the totals column is turned off).

An asterisk (*) as the first or only character in a row name will prevent that row from being printed in the report. This allows you to "pretty up" your report by not printing rows that contain intermediate data for calculations. You may want to print only a summary report, and not a detailed report on every row in your spreadsheet.

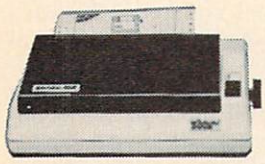
EXIT

If you want to exit any of the programs, you can do so by pressing the exit key for your machine, as described in the text for each computer. The Exit option allows you to back out and return to the data entry screen—but you will first be prompted with a message inquiring whether you want to halt the program and lose any data currently in memory.

Text continues for the IBM PC & PCjr on page 16
Text continues for the Apple II Family on page 17
Text continues for the Commodore 64 on page 24
Text continues for the TI-99/4A on page 34

Sensational Prices!

. . . On Our Most Popular Hardware!



GEMINI 10X PRINTER $289
Popular 120 char/sec bidirectional printer, full graphics, friction and tractor feed.

EPSON RX-80 PRINTER $299
Famous 100 char/sec bidirectional printer, full graphics, 128 typestyles, tractor feed.

AXIOM PARALLEL INTERFACE $84.95
Stand-alone device connects any standard parallel printer directly to TI 99/4A.

CORCOMP RS-232 INTERFACE $127
Stand-alone device provides one serial and one parallel port for printer, modem, etc. Connects to side of TI 99/4A console.

CORCOMP 9900 MICRO-EXPANSION SYSTEM . $329
RS-232 interface, 32K memory expansion, and DSDD disk controller. Complete!

CORCOMP 99000 EXPANSION SYSTEM ..... CALL
All the features of the 9900 system above, but in an expansion box with space for one disk drive or two half-height drives.

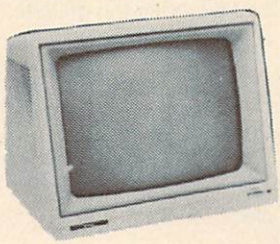
CORCOMP RS-232 CARD $89.95
Fits into your TI Peripheral Expansion Box.

CORCOMP DISK CONTROLLER CARD $169.95
Fits into your TI Peripheral Expansion Box.

CORCOMP 32K MEMORY CARD $109.95
Fits into your TI Peripheral Expansion Box.

FAMOUS WICO JOYSTICK $21.95
Best selling bat-handle version. Requires adapter for TI.
Wico TI Adapter **$8.95**

PROSTICK II $24.95
This game master has 4-way and 8-way action. Comes with TI adapter.



BMC COLOR MONITOR $239
Great resolution (350 lines) and a big 13" sceen. Full year parts and labor warranty!

SKC 5¼" FLOPPY DISKS
Certified 100% error free! Lifetime exchange guarantee!
Single-sided, single-density. Box of 10 ..... **$17.95**
Single-sided, double-density. Box of 10 ..... **$20.95**
Double-sided, double-density. Box of 10 ..... **$23.95**

TEAC HALF-HEIGHT DISK DRIVE $189
Great quality double-sided, double-density. Install in Expansion Box or in case below. Complete with cable and hardware. Functions as single-density with TI controller card and double-density with CORCOMP controller.
Case and power supply for external use of above drive. Holds up to two half-height drives **$59.95**

"EVERYTHING BOOK" FOR THE TI HOME COMPUTER — FREE —
No purchase necessary. Describes how to use hundreds of hardware and software items. Call or write for your free copy.

We're the big company that gives you personal service. Join the thousands of our satisfied customers who have saved money without sacrificing a bit of quality or service. All products are brand new, latest merchandise, in the original package, and with full manufacturer's warranties. Shipped from our warehouse direct to your home by UPS. Happy customers are our business.

From Your Friends At
TENEX Computer Express
T.M.

We gladly accept mail orders!
P.O. Box 6578
South Bend, IN 46660

Questions? Call
219/259-7051

SHIPPING CHARGES	
ORDER AMOUNT	CHARGE
less than $20.00	$2.75
$20.00-$39.99	3.75
$40.00-$74.99	4.75
$75.00-$149.99	5.75
$150.00-$299.99	6.75
$300 & up	7.75

NO EXTRA FEE FOR CHARGES

ORDER TOLL FREE
1-800-348-2778

Due to the rapidly changing electronics market there may be changes in prices and availability of merchandise by the time you receive this publication. Please call or write for latest information and lowest prices.

Ad H54

Circle 4 on Reader Service Card.

Snap-Calc . . . from p. 14

The IBM version of *Snap-Calc*™ will run on the IBM PC, PCXT, and PCjr, with either BASICA, or Cartridge BASIC. The IBM versions contain several enhanced features because of their larger memory capacity and screen size. The program uses TEXT mode only, and will work equally well with monochrome and color monitors.

The best enhancement is the size of the spreadsheet matrix. The program is currently set up for 30 columns and 60 rows, which should be large enough for almost any home application. The matrix size can be altered very easily—simply change the value assigned to variables A and B in line 210. A represents the maximum number of rows set up, and B represents the maximum number of columns. The arrays are then dimensioned with these variables, and all limit checks are done accordingly.

Before haphazardly changing these variables, however, consider how much space you really need. The size of your matrix determines the amount of disk used to store it all. It also affects calculation time if the computer has to scan more rows for equations, and execute equations for more columns.

Because of the complexity of the program, operational differences exist between each of the four program versions. These minor differences, which could not be covered in the main text of this article, are covered here.

The Keys To Success

The IBM keyboard is quite different from other systems. On this program, its function keys and separate arrow keys are used in the following manner:

FUNCTION KEY	FUNCTION
1	Load data from disk.
2	Save data to disk.
3	Clear entry/erase.
5	Calculate the logic model.
6	Print report.
7	Start logic entry mode.
9	Exit the program.

On the PCjr you need to press [Fn] in conjunction with the number keys 1 through 9. On the PC and PCXT you can simply press the function keys on the left side of the keyboard.

The four arrow keys can be used to move the cursor around the screen from one cell to another. When the cursor attempts to go off the edge of the screen, the screen scrolls in the opposite direction to bring the desired cell into view.

On the PCjr you can move by pages by pressing the [Fn] key with an arrow key. This allows you to move up or down five cells at a time, or left or right three cells at a time. On the PC and PCXT you can use the [HOME], [END], [PG UP], and [PG DN] keys to do the same thing:

PC PCXT	FUNCTION
HOME	Page up.
BEND	Page down.
PG UP	Page right.
PG DN	Page left.

Screen Size

The IBM versions display 40 columns of text. This is enough room to fit three columns of information on the screen at one time. The three columns can be either the row labels and two numeric columns, or three numeric columns. The row labels can be up to ten characters long, and the numeric values can be up to eight characters long, including a fixed decimal point. Only five digits may be entered to the left of the decimal point, and two digits to the right of it. Thus the maximum value that can be entered or displayed is 99999.99.

When entering numeric values in the spreadsheet, simply move the cursor to the selected cell and type in your numbers. The value will be updated on the screen automatically with every key press. After you have entered five digits to the left of the decimal point, the computer will not accept any more input except for a decimal point and digits to the right of it. If you enter only one digit to the right of the decimal, the computer will fill in a zero behind it.

Different Files

When you select either the Save or Load option, you are prompted to indicate whether you wish to work with the spreadsheet data or the logic model. The logic model includes all of the parameters that you set up in the Logic Entry mode. After selecting one of these you will be asked to enter a file name. If you selected the logic model under the Save option, the file name will be taken from the logic model name as specified in the Logic Entry mode. If no logic name has been specified, then you will be prompted for the file name, and the file name will then become the logic model name.

Logic Mode

Commands are entered in Logic mode with the INPUT statement of BASIC. This statement allows inputs of up to 255 characters—the maximum length for any command. The only command which can reach this length and still be a legal command is the one that lets you set up row equations (r = equation). It would take a very long and complicated equation to use up all 255 characters, and you will probably never encounter a case where more characters would be desirable.

The PRINT command will output to the system's default printer device. If the parallel printer interface is installed, it will be connected to the parallel port. If not, then output will be directed to the serial port.

BASIC Precision

If you use the IBM BASICA language, start up BASIC by answering the system prompt (A>) with BASICA/D. (This is to be done any time you RUN *Snap-Calc*™.) This enables the language to operate in double precision mode for increased accuracy. IBM Cartridge BASIC used with the PCjr requires no special start up to ensure high accuracy—just type BASIC.

[Note: The listing states that this program will run with Cassette BASIC. This is incorrect. This version will only run under BASICA or Cartridge BASIC. —Ed.]

SNAP-CALC (IBM PC & PCjr) Explanation of the Program

Line Nos.	
100-200	Program header.
210-260	Initialize program variables and key interrupt branching.
270	Control logic to display the data entry screen.
280-290	Control loop to accept entry on the data entry screen.
300-320	Routine to build numeric values and display them on the data entry screen.
330-340	Routine to build a row name and display it on the data entry screen.
350-420	Display the spreadsheet on the screen.
430-440	Two routines to move the cursor.
450-560	Load data or logic files.
570-650	Save data or logic files.
660	Routine to clear an entry or erase a row name.
670-830	Calculate row equations.
840	Calculate the totals column.
850-900	Print spreadsheet report to the printer.
910	Error routine for the report portion of the program.
920-1100	Main control loop for the logic entry mode.
1110	Routine to accept a new logic name.
1120-1130	Routine to accept a new totals column.
1140-1180	Routine for the NEW command.
1190-1220	Routine for the LIST command.
1230-1250	Routine to put the command string back together for the LIST and PRINT commands.
1260-1310	Routine for the PRINT command.
1320	Logic mode error message.
1330	Return to data entry screen when END command is used.
1340-1350	Move up one cell.
1360-1370	Move left one cell.
1380-1390	Move right one cell.
1400-1410	Move down one cell.
1420-1430	Move up five cells.
1440-1450	Move left three cells.
1460-1470	Move right three cells.
1480-1490	Move down five cells.
1500-1520	Key scan routine.
1530-1550	Routines to turn key interupts on and off.
1560-1590	Check for function arrow key inputs so that paging can be done.
1600	Program main error routine.
1610	Data containing logic mode commands.
1620	Data containing logic mode math operators.
1630	Routine to exit the program.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

The Apple version of *Snap-Calc*™ is different from the other versions because of the Apple's keyboard. The IBM PC and PCjr, Commmodore 64, and TI-99/4A computers all have function keys that can be used to select the options. But this is not the case on the Apple; its [CONTROL] key is used with a letter to activate the following functions:

Press [CONTROL]	for FUNCTIONS:
L	Load data.
S	Save data.
E	Clear entry/erase.
Z	Calculate logic model.
P	Print spreadsheet.
F	Start logic entry mode.
I	Cursor up one cell.
J	Cursor left one cell.
K	Cursor right one cell.
M	Cursor down one cell.
Y	Cursor up five cells.
G	Cursor left three cells.
H	Cursor right three cells.
B	Cursor down five cells.

You don't need to press the [CONTROL] key to initiate these two functions:

[RETURN] Cursor down one cell.

[ESC] Exit program.

Data Entry

Because the Apple screen is 40 characters wide, it can display three columns of information on the screen at one time. These columns can include the row labels and two columns of values, or three columns of values. The row labels field can be up to ten characters long, but the numeric fields of the spreadsheet will stretch to 11 characters long.

One character is reserved for the negative sign (to be used as needed). Another character is reserved for the decimal point. This leaves five digits to the left of the decimal point, and two digits to the right of the decimal point. Thus, the largest number that can be entered or displayed is 99999.99.

When entering numbers into a numeric field of the spreadsheet, you simply need to move the cursor (the cursor is the cell which is set to inverse video) to the cell you wish to work on. Then enter the value. You do not need to press [ENTER] to save the value in the field. If you fill up the five digits to the left of the decimal point, the numbers will automatically begin entering to the right of the decimal.

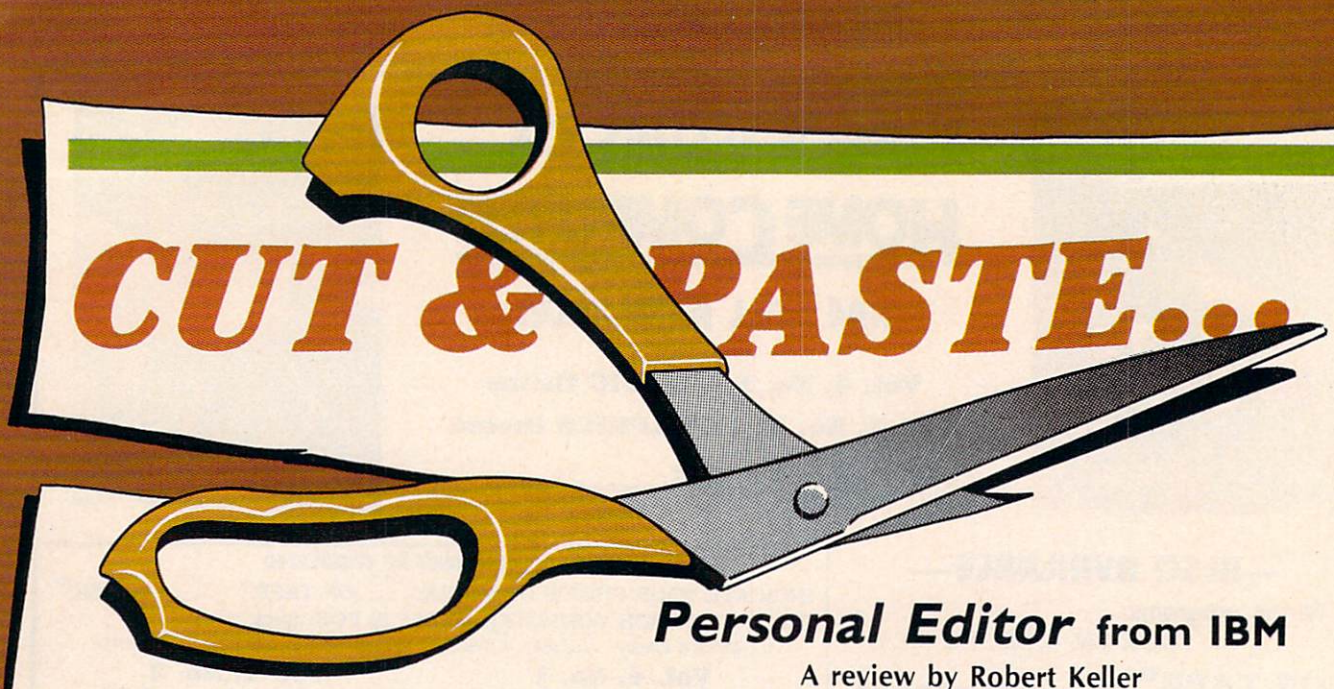
Snap-Calc™ on the Apple can handle up to 30 columns and 60 rows of information in the spreadsheet. However, these limits can be altered by changing the values assigned to variables A and B in line 580. A dictates the maximum number of rows to be used, while B dictates the maximum number of columns. Beware of increasing these figures, though, when you may not have sufficient memory—you could find yourself with a disastrous headache one day when your program runs out of memory, and you lose all of your data. For most home spreadsheet applications, 60 rows and 30 columns should be adequate.

Files

The Apple computer does not store information to cassette tapes, only to a disk drive. Thus the Apple version does not prompt you for a device type. Instead, the first prompt (when loading or saving files) asks whether you wish to work with the spreadsheet data, or the logic model. The logic model includes all of the specifications that you established in the Logic Entry mode.

The next prompt asks you for the file name. If you are saving a logic model to disk, you may not get the file name prompt. If you assigned a name to the logic model when in the Logic Entry mode, this name is used as the file name. Finally, if no name has been given to the logic model, then you will be asked for the file name. The file name you enter will then become the new logic name of the model the next time you load it into memory.

Continued on page 24



Personal Editor from IBM

A review by Robert Keller
HCM Staff

Thirteen or more word processors and text editors have clamored for my attention during the last few months. Each has had its day, yet I find myself going back again and again to IBM's *Personal Editor*.

What is it about this program that has commanded my attention so? Perhaps it's the program's versatility, for *Personal Editor* is both a fast, clean word processor *and* a programmer's tool. Moreover, if you don't like the way it performs, you can rebuild it to your own specifications.

Name:	IBM Personal Editor
Author:	Jim Wyllie
Program Type:	Full Screen Text Editor
Machine:	PCjr with 128K and disk drive; PC, XT & IBM Portable with 64K
Distributor:	IBM Corporation Boca Raton, Florida 33432
Price:	$100

	Poor	Fair	Good	Excellent
Performance				
Ease of Use				
Documentation				

As a Word Processor

The *Personal Editor* is simple to start up. Insert the disk and type PE, bypass a title screen by pressing [ENTER], and you only have one additional key to press—[ESC]—before you're ready to start writing. This helps make initial exploration painless, and I still appreciate it months later whenever I reach for *Personal Editor* in the midst of an engrossing problem or an inspiration for some text. Many other word processors require a whole lexicon of start-up procedures and commands.

Personal Editor also uses the standard DOS keyboard service routines, so you can go ahead with your work while the program is loading. Your keystrokes go into a buffer; the program catches up when it's loaded.

The program uses nearly the whole screen to display your text—other programs frequently display menus and icons and leave you with only a cramped half-screen to work in. *Personal Editor* commands are executed on a "command line" which always appears at the bottom of the screen. The cursor's current line number and its current column appear under the command line. Next to these numbers is either the word "insert" or "replace." What you type will either be inserted into the text at the cursor position, or written over existing text. You toggle back and forth between these two modes by pressing the [INS] key. And that's really about all there is to it.

Press [ESC] to move to the text area, between the start-of-file and end-of-file markers, and begin writing text. All of the IBM text editing keys, [HOME], [PGUP], [END] and etc., function as you would expect. The program does its work directly in video memory, which makes it remarkably fast.

If you want to save what you've written, return to the command line by pressing [ESC] again, type SAVE and a file name, and press [ENTER]. The file name appears under the command line and your work is saved to disk. The program disk will hold plenty of text if you want to store it there—the program takes up only 45K of the more than 300K of disk storage. The program also never needs to refer back to the program disk after it is loaded and running. If you have a single-drive system and prefer to store your text on a separate disk, you can put your storage disk in the drive after you begin, and work with no difficulty. Other programs either allow you only a page or two of storage on the program disk, or no storage at all; some even require repeated access to the program disk—a total headache if you have a one-drive system.

Commands

The *Personal Editor* is "driven" by 17 commands—eight of the most common are EDIT, SAVE, PRINT, ERASE, NAME, LOCATE, CHANGE and QUIT. The "menu-driven" versus "command-driven" debate goes on—menus make things easy at first, but once you become familiar with a program, they can be burdensome. *Command-driven* means that the program just waits for you to give it commands—a potentially faster and more flexible approach than using menus, but the commands had better be clear and easily remembered. In the case of *Personal Editor*, command-driven works well.

Active Files

To work on another file, type E (for EDIT) on the command line, type the file name, and press [ENTER]. The program will first look on the disk, and if it doesn't find the file, it will assume that you want to start building a new one. In the meantime, the first file that you were working on is still in memory. Up to 20 different files can be in immediate memory at once; with a single keystroke, you can quickly move between these "active" files. The *Personal Editor* uses all available memory for active files, and then creates a "spill file" on disk, using the disk as *virtual memory*. In other words, the disk becomes an extension of the computer's memory, invisible to the user except for an unavoidable slow-down caused by reading and writing to the disk. In months of use, however, I've only created one document large enough to require a spill file.

Word Processing Capabilities

Many standard word-processing capabilities are found in the program. Margins and tabs are handled in a flexible, capable

Continued on page 21

WITH IBM

Two Word Processors
. . . TO SUIT YOUR TASTE

EASYWRITING ON THE PC & PCjr:

A report on *EasyWriter II* as used on a single-disk drive system

by Wayne Koberstein
HCM Staff

Name:	EasyWriter II
Program Type:	Word Processor
Machine:	IBM PC and PCjr
Distributor:	Information Unlimited Software, Inc. 2401 Marinship Way Sausalito, CA 94965
Price:	EasyWriter II, EasySpeller II, and EasyMailer II Pack $395.00

On PC:	Poor	Fair	Good	Excellent
Performance			■■■■■	
Ease of Use			■■■■■	
Documentation			■■■■	

On PCjr:	Poor	Fair	Good	Excellent
Performance			■■■■	
Ease of Use			■■■	

This article investigates the word-processing system, *EasyWriter II*, designed for the IBM PC. It is partly a review of the program itself, and partly a tutorial in applying the program to the PCjr. The article is divided into six main sections: 1. **Editing Typewriter** discusses the general structure of the program. 2. **Becoming Friends** addresses the issue of user-friendliness in word processing. 3. **Power of the Program** describes *EasyWriter II*'s range of capabilities. 4. **One-Horse Cart** explains problems in using the program on the single-disk PCjr. 5. **A Preferred Junior Method** suggests a procedure and describes its limitations. 6. **The Keyboard Factor** sums up key-command changes for *EasyWriter II* on the PCjr keyboard. **Note:** This program was not intended by its developers to be used on the PCjr.

We know PCjr often has good cause to speak up to the elder PC—saying, "If you can do it, I can do it." Although this is not always true, many have wondered how much Junior can do in comparison to the larger machine. So to give Junior a real challenge, we let him munch on some software outside of his normal diet—*EasyWriter II*, a production word processing system by Information Unlimited Software (IUS). *EasyWriter II* is intended for the IBM PC with two disk drives. But will it work on the single-drive PCjr? We ran this program on both machines and compared their performance function by function—hoping that this word processor would work reasonably well on the smaller, cheaper model of the IBM machine.

Word processing is the first thing many users want to do with their home computers. And those with smaller models like PCjr may want a more powerful program than is specifically designed for the limitations of their machine. We must emphasize, however, that IUS does not recommend using *EasyWriter II* on the PCjr or any other machine for which it is not designed. Still, letting you be the judge, we can look at what the program can do normally, and how it performs when stretched beyond its usual boundaries.

An Editing Typewriter

EasyWriter II is configured around the typewritten page. Pages make up documents, and documents make up file folders. To create a document, you first open and prepare a file folder. Within that file you can then define any number of documents, limited only by the amount of space available on the data disk. Ordinarily, each disk will contain a total of about 160 pages of 54 lines. The disk space can be partitioned between *different* documents or as *one* very big document. Each time you prepare a new folder, the program will display the number of pages left to fill on your disk.

Typing and editing can be done together. They are not separate functions. While typing, you have a choice of 7 different editing modes and 21 editing functions. (See Chart.) Some of these functions operate differently in each mode. For instance, the delete function deletes everything from one character to a whole page, depending on which mode of operation you are in. Modes also define areas of font changes. All functions and modes are selected by either one or a combination of keys on the IBM keyboard.

The PC keyboard has a separate set of function keys which make most selections a one-button operation. (Selections on the PCjr are a little more complicated, as we'll see later.) A small chart listing all commands is included in the *EasyWriter II* package; it is to be placed at the top of the keyboard.

Continued on page 22

Chart of Editing Functions

EDITING MODES	FUNCTIONS WITHIN EACH MODE:
Character	Center
Word	Copy
Sentence	Cut
Line	Paste
Paragraph	Insert
Block	Delete
Page	Go To
	Next
	Previous
	Adjust

SEPARATE FUNCTIONS:
Acknowledge (Error Message)
Cancel (Unsaved Changes)
Clear (to paragraph end)
Headers/Footers (Define)
Help
Tag/Merge
Page Parameters (Adjust)
Print Ruler (Adjust Tabs and Margins)
Search/Replace (Word or Phrase)

FONTS:
Bold, Underline, Double-Underline, Normal, Overstrike, Shadow, Special, Subscript, Superscript

2 IMPORTANT PERIPHERALS

For Your Texas Instruments Personal Computer

$309

Direct-Connect Printer, GP-100TI

Because of our built-in interface, all you have to do to connect a GP-100TI printer to your Texas Instruments 99/4A computer is plug it in. The GP-100TI has adjustable tractors, cartridge ribbon, uses standard fan-fold paper and has full dot-addressable graphics. It provides clear, dot-matrix printout from all TI software including cartridge, which allows you to set number of line feeds per line, line length, margins and self-test. Another plus — expansion modules can be daisy-chained through the built-in edge connector.

$99

ParallAx TI Expansion Interface

You no longer have to settle for Texas Instruments' bulky expansion box with its messy cable when for just $99 you can connect any parallel-input printer to your TI computer using our ParallAx TI printer interface. It's smaller, neater, and plugs into the side of your TI computer. ParallAx TI has the same features as our GP-100TI printer (described above) and you can daisy-chain expansion modules through the built-in edge connector.

THESE QUALITY AXIOM PRODUCTS ARE AVAILABLE AT LEADING DEPARTMENT AND DISCOUNT STORES.

FOR THE OUTLET NEAREST YOU, CALL 818/365-9521.

AXIOM

AXIOM CORPORATION
1014 Griswold Avenue, San Fernando, California 91340
818/365-9521 • TWX 910-496-1746

Circle 5 on Reader Service Card.

Personal Editor ... from p. 18

fashion. You can mark a block of text and move it within a file or to another active file, deleting the original block or leaving it unchanged. You can delete a block, a line, a word, or a character. You can search and replace strings either forward or backward from the cursor position, making global changes, or have the program query you about the appropriateness of each change. Once you get accustomed to the commands, the program's "ease of use" is the match of any word processor I've tried.

For housekeeping, *Personal Editor* maintains three special "internal" files. One is .unnamed, which holds the last five changes made in any active file. If you blunder, you can call .unnamed up and move the lost text back to your workspace. The second internal file is dir, the current disk directory, and the third, .keydefs, I'll discuss later.

The *Personal Editor* allows you to build your own "text tool." For sending text over an RS232 channel to a computerized typesetter, for instance, you would need to include sets of complicated codes for typesetter formatting. With *Personal Editor*, you can build a word processor tailored to your specific transfer needs—one that inserts all the formatting codes with the touch of a key.

Word Processing Drawbacks

It is unfortunate that *Personal Editor* has no provision for inserting headers and footers, for pagination, or for halting printing on page boundaries. There could be an "add-on" remedy for these difficulties, but it probably wouldn't be as satisfactory as if the provisions were part of the original program.

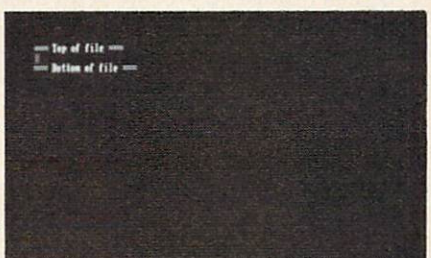
Additionally, *Personal Editor* won't display sophisticated font and format changes on its screen, although you can control such parameters in the eventual printed text. Another drawback is that you can't readily print just a section of a document. You have to first transfer the section to its own file, or erase all of the text before and after the section. Text reformatting during editing could also be handled more effectively: it appears to take a few unnecessary keystrokes now and then.

Finally, the program appears to have difficulty using the LOCATE command to find a string in the text when the string wraps from one line to the next—a deficiency that hamstrings the LOCATE command.

Beyond Words

Personal Editor is also a programmer's tool. It produces standard ASCII DOS files as its output (with an invisible compression of multiple spaces). PC DOS's TYPE, SORT, FIND, PRINT and piping facilities work on these files. Write a BASIC program with *Personal Editor* and it will look just as good to the PC and PCjr as if it were written directly from the BASIC language.

Personal Editor is equally forgiving when asked to access files produced elsewhere. Need to look at a file generated by another word processor? The only limitation here is that the program can input a line length of only 254 characters before it needs to find a carriage return. Some word processors eliminate all carriage returns and other such standard delimiters, substituting their own internal codes.



***Personal Editor*'s uncluttered entry screen.**

Need to send a file created with the Super-X word processor over a modem? The *Personal Editor* will quickly, with its CHANGE command, locate and eliminate all of the idiosyncratic codes that Super-X inserts in its text, giving you clean ASCII copy. Likewise, if you need to convert a BASIC program from the Super-Z Computer to the PC, just send the listing over to the PC by RS232, and you are all set to search and replace Super-Z commands with equivalent PC commands.

Want to write some Pascal? Need to search for the occurance of a variable in an ancient and convoluted BASIC program? Want to write some messages for electronic mail? Need to create a batch file for DOS? Want to write some BASIC and have the luxury of using labels to stand for subroutine line numbers you haven't written yet (so you can just go back and search and replace the labels when you're finished)? Want to change the variable A1 to COUNT throughout a program with just a few keystrokes? *Personal Editor* is ready for all these tasks.

Building Your Own

With the *Personal Editor*, it is possible to change the meaning of the keys. Thus, the statement

```
def a-t = 'this is a test'
```

makes the combination of the [ALT] and T keys produce the string "this is a test." Such strings can be up to 255 characters in length. There is also a *meta-language* of 63 text editing "functions." A few of these functions are:

[INDENT] . . . moves the cursor in from the margin the distance specified by the margin command.
[TAB WORD] . . . moves the cursor right one word.
[TOP] . . . moves the cursor to the start of the file.
[LEFT] . . . moves the cursor one space to the left.

These 63 functions can be combined with each other and with any characters to form a key definition, as long as the 255 character length limit is observed.

The current key definitions are always available for inspection in an internal file called .keydefs. I'm writing this article on the *Personal Editor*, so I'll pause here and call .keydefs into an active file. Now I'll go to that file and bring back some examples:

```
def up = [cursor data] [up]
```

This command moves the cursor into the text if on the command line, or moves it up one line from its current position. ("Up" stands for the PC up-arrow cursor-movement key.) Next, let's examine:

```
def a-f1 = [cursor command] [begin line] [erase end line] 'e' [execute]
```

This command moves you to the next active file. It moves the cursor to the command line, erases the command line, places an E there, and executes a carriage return.

Let's look at a more complicated assignment where [ALT]-W deletes the word under the cursor and reformats the paragraph:

```
def a-w = [unmark] [mark char] [tab word] [left] [mark char]
[begin mark] [delete mark] [mark line] [find blank line] [up]
[mark line] [reflow] [begin mark] [unmark]
```

Personal Editor comes with the keys already defined, so you don't have to start from scratch. The manual contains some suggestions for alternate designs, and clearly instructs you if you want to build something of your own. There is one outright bug: design suggestions in the back of the manual all indicate that you should use left and right single quotes, 'like this', in your key definitions, but only two right, two left, or double quotes will actually work. You can't mix quotes.

MACRO

If you precede a file name with MACRO, the *Personal Editor* will attempt to execute the file. You can put any combination of *Personal Editor* commands in a file, one to a line, and they will all be executed. You can quickly change the whole profile of the package by executing a file of key definitions, for example, or execute any list of repetitive or often-used commands.

Conclusion

The *Personal Editor* wouldn't be useful for someone who merely needed to turn out paginated papers by the pound. And although the manual is thorough and well-written, users will need at least a proclivity for programming in order to be able to restructure the program as I've described above. Nevertheless, *Personal Editor* is a fast and flexible tool with a lot of capabilities—foremost of which is its capability to allow a user to tailor it to individual needs. It's software like this that makes working with microcomputers productive and exciting.

HCM

Easywriting . . . from p. 19

Because there is a lot of emphasis these days on user-friendliness, I suppose some people will prefer making menu selections with a cursor or a "mouse." However, once you memorize a set of commands, it becomes just about as easy to push a few keys as it might be to move a cursor or clear a space on your desk for a small rodent to run around on. And a few milliseconds difference between selection procedures may not seem that significant when compared to the several seconds it takes to contemplate whether a change is needed in the first place.

Structures like pages, documents, and even menus can also be viewed as constraints. Some people consider a program with a more flexible command structure, such as IBM's *Personal Editor*, to be friendlier to their needs and use. These folks are willing to put up with more study and memorization in order to have greater freedom in defining files and creating their own commands. User-friendliness, it appears, is often in the eye of the user.

Becoming Friends

Friendliness *is* an issue, however, when it comes to learning the *EasyWriter II* program and going through it for the first time. Quite simply, it sometimes allows you to make mistakes and even lose text by not clearly directing you through the routine with specific screen prompts at every point of choice. This should not happen with a truly menu-driven system. For example, if you choose EDIT off the main menu before opening or preparing a file, you will be given a blank page, and you can start typing. But when you go to save that text, you'll find that you can't return to your file because it does not yet have a name and therefore does not "exist." In normal editing, you do have to respond to a query before deleting anything larger than a line or a sentence.

Word-wrapping is another feature that may give the beginner some surprise. Cursor movement from one line to the next is automatic, unless you begin a long word just before reaching the wrap-around point, or you space right continuously from the middle of the line. Either way, you may be startled to see your text disappear, leaving what looks like a few random words and letters along the screen's left side. This is the right edge of your text.

In the first instance, pressing the space bar after the word usually triggers the wrap-around, and your page will shift back to the right into view. In the second case, your cursor will continue to march merrily along on a blank screen until you space back to the left. Believe me, this has fooled some into thinking that their text had crashed—causing them to reboot and lose work before escaping to save. This comedy of errors could be prevented with a specific warning in the instructions.

With these exceptions, documentation for *EasyWriter II* consists of a very thorough and mostly readable text and diskette tutorial—although the sheer size of the material can be somewhat imposing. A HELP function will display useful, basic information when called. Careful study of these aids will allow you to avoid most costly errors. With this package, there are two important lessons to learn: (1) study well *before* you start typing that prize-winning novel, and (2) *don't panic*. Once you learn the program and all its variables, you can take advantage of its considerable power.

Power of the Program

Some of the more powerful functions of this word-processing system have to do with text manipulation. It's very nice to be able to type a line and then center it on the page with a simple command. It's even nicer to have the freedom to assemble blocks of text from several different documents (or even disks) into one coherent whole. I particularly liked the TAG and MERGE function, which I used extensively to put together this article. Similar to CUT and PASTE, which is done piece by piece, TAG and MERGE is more flexible because it allows you to go through and tag many separate sections of a rough draft and then insert them at specific locations in the main text—all in *one* operation. This is handy for writing long documents and customizing shorter texts, such as form letters. *EasyWriter II* also includes other useful functions of this type. PAGINATE, for example, allows you to adjust the length of the page and copy a section from one page to another while changing its format, and leaving the unchanged version where it was. In addition, you can go back and add page headers and footers with automatic page numbering any time you wish.

System functions beyond editing or manipulating text greatly extend the power of *EasyWriter II*. These are accessible on the "Housekeeping" disk which is part of the program package. They include copying documents to the same or other diskettes, importing and exporting programs or documents to and from MS-DOS-based files, checking data integrity, and changing between printers or altering printer configurations. It is also possible at this level to completely change default settings for most of the functions, including page formatting. This feature is rare among word-processing programs intended for the personal computer.

Of the Housekeeping functions, IMPORT/EXPORT may be one of the most useful. With it you can write a program or text in *EasyWriter II*, taking advantage of all its handy features like SEARCH/REPLACE. Then, for example, you can export your document to another MS-DOS file in another language such as Pascal. This increases your editing power immensely over the screen editing programs usually provided for these other language systems.

EasyWriter II tops it off by offering several appealing features: You have your choice of up to nine different fonts. You can mark any place in the text and the program will find that mark when you ask. You can also have *EasyWriter* search out a certain word you choose—either just to find the word and call it to your attention, or to replace it with another word of your choice. Probably the most impressive of these features is the optional *EasySpeller II* program, which you can copy onto the system diskette and call up at any time. This is a built-in dictionary. It will search through your text and call any misspelled words to your attention. You can also have it check spelling on any specific word simply by placing the cursor on the first letter and punching VERIFY.

One-Horse Cart

We tested *EasyWriter II* on both a two-drive PC and the little Junior itself. On the former, the program works very quickly and smoothly. On the latter, as you would expect, it doesn't go quite as quickly or smoothly; but with some patience, it does *go*. You can, in fact, get about as much mileage out of the one-horse cart as the two-horse carriage, although it's a bit like having to change carts at every major intersection. In addition, there are some hills you cannot climb.

If you're feeling adventurous and you want to try this yourself, be prepared to do a lot of disk-swapping—that is taking one disk out of the machine and replacing it with another. Although you can operate most of the *EasyWriter II* editing functions successfully on both the one-disk Junior and the PC, many require interplay between the data disk and the System program disk. With two disk drives, the computer switches from one disk to the other. With one drive, *you* do. You can get around a lot of this by storing data on the System disk's remaining free space; but for that, you sacrifice software power and quite a bit of storage space for text.

EasyWriter II works from a set of master disks including System and Housekeeping, with *EasySpeller II* (spelling checker) and *EasyMailer II* (a routine for bulk mailings). To use these programs, you must first format a blank disk with a DOS 2.1 file disk; then you copy the *EasyWriter* program onto the formatted disk. Either Housekeeping, *EasyMailer*, or *EasySpeller* can be incorporated with the System program on the same disk through an additional procedure. With any of these additions, however, about 95% of the System disk is occupied. This leaves room for only 8-10 pages of text. Without these large routines, only about 54% of the System disk is used up. This leaves room for about 80 pages—not bad, really.

Of course, if you create a separate data disk you get separate storage for a full 160 or more pages. However, on the PCjr it takes over a dozen swaps just to set up this way, and another six to get into the program. Other moves requiring exchanges between data and System disks include (1) changing pages, (2) saving files, (3) manipulating large blocks of text (as with TAG

and MERGE), (4) copying, (5) access to System Functions, (6) printing, and (7) others we've yet to encounter. When a disk trade is needed, a prompt will appear on the screen, saying to put in the disk for either Drive A or Drive B and press any key when ready. When it says Drive A, you put in the System disk. For Drive B, you put in the Data disk. The prompt appears a few lines above the blinking cursor, which sometimes will place it right over existing text; you therefore have to watch closely for each prompt. One note: You ignore this prompt when it lingers after you press the key.

Having tried it, I would recommend that you *not* separate the System program and your text on two different disks. It is simply too confusing and too "hazardous to the health" of your disks. Instead, here is a method (albeit with a few limitations) for using the program with a shared Data and System disk.

A Preferred Junior Method

Drive B is the default drive for the data disk. Drive A is the default drive for the program disk. If you want your text to be on the same disk as the system program, you have to address Drive A when you open or define a file folder. To do this, type a: before the the name of the file as you open it, as in a:easywriter 2—the name I've given the file containing this article. The a: routes your text to the System disk on the single drive of the PCjr. This enables the computer to go back and forth between the *EasyWriter II* program and the stored data without having to switch disks. Without the a:, your text will be stored on the disk for default drive B, requiring a manual disk trade every time the computer searches for or stores data.

Most editing functions are both immediately accessible and fast with this shared-disk method. However, both the one-disk and two-disk methods of storing data have important limitations on a machine with only one drive. Functions like importing and exporting files from one disk to another, and copying documents between disks are impractical because of the number of disk trades involved. What is a very fast process on a two-drive PC becomes virtually impossible on the single-disk PCjr. This rules out the practical use of *EasySpeller II* or *EasyMailer*. You would have to type your finished document into the limited space available on a System disk containing *EasySpeller II* or *EasyMailer II* before employing either option. [One possible remedy would be installing the supplementary programs on a "RAM disk." Watch for an interesting product review in a forthcoming issue.—Ed]

You can, however, export or import files from your shared System/data disk onto the *same* disk. Turning an *EasyWriter II* file into a DOS file in this way removes a lot of the program's command structure, thus reducing the file to about 30 percent of its original storage size. Therefore, if you store your finished documents as "exported" DOS files, and save enough room to "import" a file back into *EasyWriter* for editing or printing, you will nearly triple available text storage on your System/data disk. During the Export/Import process, type a: before the name of each exported or imported file when prompted, and all files will remain on the same disk.

The Keyboard Factor

When running *EasyWriter II* on a one-disk PC, comparison in performance between it and the PCjr boils down to their respective keyboards. Some program functions require pretty fancy fingering on the PCjr. Several commands use the [SHIFT] key as part of their configuration. With one such command, specifically SEARCH/REPLACE, nothing happens until you press the *right-hand* [SHIFT] key! Hitting the left-hand [SHIFT] will simply have no effect. To key other functions, *first* press [FN], *release* it, then hit the corresponding number. If either the [SHIFT] or the [ALT] key is involved in the command, hold it down *while* pressing the others. For example, you reach TAG by holding down [ALT], hitting [FN], and then 7. Many commands will work if the [FN] and number keys are pressed simultaneously—but not others, such as TAG.

PCjr's keyboard itself is mechanically awkward, with nearly flat, block-like keys and stiff action. An experienced typist can comfortably type about 50-55 words per minute on the Junior compared to 80-85 wpm on the PC. Incidentally, I was able to use *EasyWriter II's* 80-column format with an ordinary TV hooked up via the IBM Modulator—although the letters were somewhat fuzzy.

Let Us Hear

In general, *EasyWriter II* works well enough on the PCjr to be highly useful. Of course, you will have to balance its price tag against your own particular needs and tastes. However, now that you PCjr owners have heard the news, you may want to choose this powerful option for your own machine. Coming up in future issues will be more information about *EasyWriter II*, as well as new ideas on how to get more out of Junior. One such feature will discuss a method of patching a second disk drive into the unit—even though it is designed for only one. This definitely makes "EasyWriting" on Junior a whole lot—well, *easier*. We will also be including more key-in-and-RUN software for the IBM PC itself. In the meantime, we would like to hear your response to this experiment. So keep those cards and letters coming.

HCM

Snap-Calc . . . from p. 17

Printer Reports

Because the Apple II family (with the exception of the IIc) is capable of hosting a printer interface card in many different slots, you may need to modify one line of the program in order to access the proper slot address. The program as listed here sets the printer port to slot 1. However, if your printer resides in a different slot, you will need to modify line 5320—which contains a PR# command—to access the slot that contains your printer card.

Logic Mode

In Logic mode you are prompted to first enter one of the logic function commands. The maximum length for any command on the Apple is 255 characters, but the only command capable of reaching this length is the row equation assignment (r = equation). After typing in each equation, press [RETURN] to enter it.

It is possible to enter control characters into a logic command which would normally be fatal to the program. Control characters are not printed on the screen and are difficult to detect. For this reason, a special error message ILLEGAL CHARACTERS ENTERED appears if you enter control characters. If the problem is simply an incorrectly-entered command and the computer can't make heads or tails of it, the message CAN'T UNDERSTAND YOUR ENTRY will be displayed.

Getting Out

When you're all done working with your *Snap-Calc*™ program, you can exit it by pressing the [ESC] key in the upper left corner of the keyboard. This will not automatically stop the program—you will first be prompted with the message ARE YOU SURE YOU WISH TO END THE PROGRAM AND ERASE ALL DATA?. At this point you can press Y and exit the program, or press any other key and return to the data entry screen. The [ESC] key option will only work while you are in the Data Entry mode.

[Note: ProDos users will need to make the following changes in the program in order to access the printer properly when running under ProDos:

Line nos.

```
4800 PRINT CHR$(4);"PR# 0"
5210 PRINT CHR$(4);"PR# 0": HOME
5320 PRINT CHR$(4);"PR# 1": RETURN
5530 PRINT CHR$(4);"PR# 0": HOME : VTAB 10 : HTAB 1
```

—Ed.]

SNAP-CALC (APPLE II Family) Explanation of the Program

Line Nos.	
100-210	Program header.
220	Set up error handling.
230-250	Branch to set up title screen.
260-370	Key input for spreadsheet data entry.
380-690	Initialize program variables and display the title screen.
700-980	Routine to display the spreadsheet.
990-1080	Format numeric variables for display on the spreadsheet.
1090-1240	Display the cell contents in inverse characters (the cursor).
1250-1340	Move cursor up.
1350-1440	Move cursor left.
1450-1560	Move cursor right.
1570-1690	Move cursor down.
1700-1780	Move up five rows.
1790-1860	Move left three columns.
1870-1980	Move right three columns.
1990-2060	Move down five rows.
2070-2540	Load files from disk.
2550-2980	Save files to disk.
2990-3070	Clear entry/erase.
3080-3240	Routine for NEW command.
3250-3420	Routine for LIST command.
3430-3530	Number entry.
3540-3610	Row name entry.
3620-4030	Calculate the logic model.
4040-4470	Main control loop for logic mode entry.
4480-4510	Routine for logic name.
4520-4570	Routine for totals column.
4580-4590	Return to data entry mode.
4600-4660	Routine for the last column.
4670-4690	Bad entry message.
4700-4730	Routine for the MOVE command.
4740-4760	Bad characters entered message.
4770-4810	Print logic routine.
4820-5250	Routine to print out entire spreadsheet.
5260-5300	Routine to scan the keyboard.
5310-5360	Set the printer channel.
5370-5440	Routine to exit the program.
5450-5490	Program data.
5500-5670	Error routines.

HCM

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

C64

The Commodore 64 version of *Snap-Calc*™ differs only slightly from the other versions, primarily because of the C-64's keyboard, screen size, and file-handling procedures.

The Keyboard

The Commodore keyboard is equipped with four function keys on the right side of the console. When pressed alone, they are function keys 1, 3, 5, and 7. But, if they are pressed along with the [SHIFT] key, they become function keys 2, 4, 6, and 8. These keys perform the following functions:

KEY	FUNCTIONS
F1	Load data file.
F2	Save data file.
F3	Clear entry/erase.
F4	Calculate the logic model.
F5	Print report.
F6	Start logic entry mode.
F8	Exit the program.

The cursor movement keys are as follows:

CURSOR KEY	FUNCTION
Left	Move left one cell.
Right	Move right one cell.
Up	Move up one cell.
Down or [RETURN]	Move down one cell.

In addition, you can move the cursor over more than one cell at a time with the following keys:

Press [SHIFT]	for FUNCTIONS
J	Move left three cells.
L	Move right three cells.
I	Move up five cells.
M	Move down five cells.

Data Entry

The Commodore screen is 40 characters wide, which is sufficient for up to three columns of data—this could include the row-names column and two numeric columns, or three numeric columns. The maximum size for a row name is ten characters. The spreadsheet is limited to numeric values ranging from −99999.99 to 99999.99. This is not true, however, for internal calculations within the logic model, as long as the end result to be placed in the cell does not excede those values.

To make an entry, simply move the cursor to the desired cell with the cursor control keys. The cursor is a sprite which places a line above and below the cell you are working on. Once you have moved the cursor to the appropriate cell, simply press the keys to make your entry—pressing [RETURN] is not necessary.

If you wish to make a correction or change an entry, you can move off the cell and then move back to it. If you do not press any keys (other than the cursor keys) while in Data Entry mode, no change will occur in the cell.

Files

If you select either function keys 1 or 2 to load and save data, you will be prompted to input the type of data you wish to work with. You have three options: data (from the spreadsheet), logic (the logic model), or both at the same time. If you choose to save the logic model and the logic name has already been established, then the logic name is used as the file name. If no logic name has been established, then you will be prompted for a file name, and the name you enter will also become the logic name. If you are saving data or loading either the logic model or data, you will be prompted for the file name. If an error occurs, the program will simply return to the data entry screen.

Logic Entry

The same logic entry conventions are used for the C-64 as for the other systems. The only difference here is the length of the entry—the Commodore allows only 78 characters to be entered with the INPUT statement. If you exceed this length, the entire entry may be lost. Thus, when in this mode, you should never let your entries exceed two lines on the screen.

The only time you can legally have a longer entry is when you enter equations for a row. If you need a more complex equation than what will fit on one row, you can break it up over two or more rows. Or, you can have one equation use the result of a previous equation. However, equations are calculated in row number order. For example, an equation in row 10 could not use the result of an equation in row 12, because row 12 would not be calculated yet.

SNAP-CALC (C-64)
Explanation of the Program

Line Nos.	
100-170	Program header.
180-210	Initialize variables.
220-490	Main control loop.
500-530	Exit program routine.
540-840	Subroutine to calculate logic model.
850-870	Subroutine to return a value in an equation.
880-1390	Control loop for logic entry mode.
1400	Subroutine to return a numeric value from an entry string.
1410-1420	Delete the cursor (sprite).
1430-1530	Subroutine to place entry string into F$( ) array.
1540-1700	Subroutine to rebuild entry string for listing to the screen or printer.
1710-2600	Subroutines for file handling.
2610-2760	Routine to build row names on the data entry screen.
2770-3130	Routine to build the cell value in data entry mode.
3140-3150	Routine to display the mode on the screen.
3160-3250	Subroutine to input the logic entry string.
3260-3280	Key scan routine.
3290-3510	Subroutine to display the data entry screen.
3520-3750	Subroutine to move the cursor one position.
3760-4210	Subroutine to print the report.
4220-4310	Routine to clear data or logic from memory.
4320-4350	Subroutine to simulate the PRINT USING statement.
4360-4390	Subroutine to check for logic entry errors.
4400-4460	Routine to store the sprite cursor's shape data in memory.
4470-4600	Sprite data for the cursor.
4610	Subroutine to simulate the PRINT AT statement.
4620-4650	Routines for opening and closing the printer port. HCM

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

HOME COMPUTER magazine proudly announces

COMPU-CRUISE *is unlike any other computer conference.*
COMPU-CRUISE *is a unique event designed to bring together people interested in personal computers, software, and the industry at large —with a complete conference agenda*—*

- *Hardware & Software Choices*
- *Word Processing & Data Base Management*
- *Spreadsheets & Decision Making*
- *Communications & Networking*
- *Accounting Packages*
- *Personal Computers for Tax & Financial Planning—*

All in the pleasurable surroundings of one of the world's most luxurious cruise liners.

**The seminars are divided into 2 Tracks:*
Track A for individual or institutional users or purchasers of personal computer products.
Track B for those involved in Marketing and Sales of personal computer products.

COMPU-CRUISE '84 REGISTRATION COSTS:

Registration costs for Compu-Cruise '84 are based on double occupancy and receipt of deposits by August 30, 1984.
A late registration fee of $150 per person must be added to the below prices if received after August 30, 1984.
The costs per person are as follows:

☐A STATEROOMS DELUXE	$1,990	☐F OUTSIDE DOUBLE ROOMS	$1,395
☐B DELUXE OUTSIDE DOUBLE ROOMS	1,735	☐G LARGE INSIDE DOUBLE ROOMS	1,375
☐C DELUXE OUTSIDE DOUBLE ROOMS	1,675	☐H INSIDE DOUBLE ROOMS	1,295
☐D LARGE OUTSIDE DOUBLE ROOMS	1,575	☐I INSIDE DOUBLE ROOMS	1,195
☐E OUTSIDE DOUBLE ROOMS	1,475		

A number of staterooms equipped for 3rd and 4th persons are offered on a space-availability basis as indicated below.
Single occupancy from 150% of base stateroom cost is also available.
Each additional guest sharing stateroom with two full-fare guests . . . . . Cruise alone $550
Cruise and Conference $950
No cancellation penalties if you cancel 60 days prior to departure date. Graduated penalties apply to all cancellations after that point.
Contact COMPU-CRUISE, Inc. for complete details and cancellation fee waivers.

ITINERARY: December 15-22

ms Nieuw Amsterdam
From Tampa,FL

Day	Port
SAT	TAMPA
SUN	AT SEA (conference seminars)
MON	PLAYA DEL CARMEN, MEXICO
	COZUMEL, MEXICO
TUES	AT SEA (conference seminars)
WED	MONTEGO BAY, JAMAICA
THU	GEORGE TOWN,
	GRAND CAYMAN
FRI	AT SEA (conference seminars)
SAT	TAMPA

HURRY! Registration Space is Limited

COMPU-CRUISE™ '84

The COMPU-CRUISE '84 *package includes:*

- **LUXURY 7-DAY CARIBBEAN CRUISE**
  Featuring specified ports of call, all meals, entertainment, and all other services as provided aboard ship.
- **COMPLETE 3-DAY-AT-SEA- SEMINAR PROGRAM**
  With notable guest speakers from the industry, all seminar hand-out materials, hands-on demonstrations and use of a fully equipped computer room.
- **ROUND-TRIP AIR FARE FROM NEAREST MAJOR GATEWAY CITY TO POINT OF EMBARKATION**
  Over 80 U.S. cities included; local transportation to gateway city at extra cost.
- **TRANSFERS AND LUGGAGE HANDLING**
  Included in package.

Gratuities, optional shore excursions, excess luggage charges, and items of a personal nature such as telephone and telegraph expenses, laundry and valet service, bar bills or personal services, and port taxes are not included.

Travel arrangements for **Compu-Cruise '84** will be handled exclusively by INTERTRAVEL, INC., Eugene, Oregon. Requests for additional registration information and/or registration deposits* should be forwarded to:

Compu-Cruise
1500 Valley River Dr., Suite 315
Eugene, Oregon 97401
Tel. (503) 341-1032

***All registration deposits and payments should be made payable to: Compu-Cruise**

Reserve Now—Limited Space Available

Circle 40 on Reader Service Card.

Please clip or photocopy

☐ Yes, I want to register for **Compu-Cruise '84.** Enclosed please find my reservation request and deposit(s) for:

Number in party: _____

Company: _____

Name(s) _____

Address: _____

City: _____ St: _____ Zip: _____

Day Phone: _____

Please check your primary area of interest:

☐ User/Institutional Buyer ☐ Product Sales/Marketing

☐ Check or money order is enclosed for the deposit.
Deposit is $250 per person.

☐ Please bill my VISA/MasterCard for the deposit(s).

Acct. No. | | | | | | | | | | | | | | | | |

Exp. Date _____

_____
Signature

Please send additional information on:

☐ **Compu-Cruise** travel arrangements ☐ Seminar agenda

BARS and PLOTS

by John Gunter *and the HCM Staff*

How many times have you gazed bleary-eyed at bunches of numbers before being able to figure out what the numbers really meant? Think how much easier and quicker it would have been to view a graphic representation instead. The program presented here will do just that—allow you to enter numerical values at the keyboard and instantly see them transformed into a bar graph on screen.

Two versions of this program are provided—one for the TI-99/4A, and one for the VIC-20. There are some basic differences between the two, but both versions perform the same fundamental operations.

The program allows you to scale your input to enter either large or small values for clear visual comparison. You can also change the color of the bars at any time, displaying up to four different colors on the 99/4A, and three colors on the VIC-20. Bar value corrections are made by simply moving a *bar cursor* at the bottom of the graph.

FUNCTIONS

C (Change color)

When you press C in response to ENTER DATA, you will be prompted to enter a number that represents your color choice. Any bars you create following this input will appear with the new color. The color choices on the TI-99/4A are blue, red, yellow and grey. The VIC-20 gives you a choice of red, blue or green.

ERASE (Erase a bar)

This command erases any bar positioned above the cursor.

P (Print report)

When this command is chosen, the computer executes a screen dump of the current graph displayed to a printer. If you are using the TI-99/4A, you are first prompted for your printer's parameters. Any printer should work for this function, because only standard ASCII characters are used to plot the graph. With the VIC-20, the printer output goes to the system's default printer on port #4.

END (Exit program)

Here the two systems differ slightly. Because the 99/4A has more memory, it was possible to add several extras to it which are not available on the VIC-20.

If you enter END on the VIC-20, the program will stop, and you will be returned to BASIC. On the 99/4A, however, you are asked to enter two lines of text as a LEGEND at the bottom of the chart. Now you can select one of several options by pressing a key. If you press S (for Stop) the program will halt and you will return to BASIC. If you press N (for New) the screen will clear, and you will be prompted for new parameters to begin a new chart. If you press P (for Print) you will get a hard-copy screen dump of the chart, including the two legend lines you entered.

The 99/4A not only contains more memory than the VIC-20, but it also can display more characters on the screen—up to 25 bars in the graph for the TI machine, compared to only 15 on the Commodore. Each bar is displayed on the screen with high resolution—i.e., the top character in the graph is redefined so that you get a more accurate representation of its true value.

In this version of the program, you can use any one of four colors to generate your bar graphs—all four colors can be used in the same graph. The computer

Continued on page 30

DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES

PRICE BREAKTHROUGH

Super Sale on New Disk Drives

Starting at **$~~169.00~~!** NEW LOW PRICE

Tandon — Siemens — Remex — MPI — Teac — Shugart — Tabor

40 or 80 Tracks — Single or Dual Head — New 3½'' Drivette™

Our Disk Drives are Capable of Single **and** Dual Density Operation

The **NEWEST** Technology Capable of Operating on Most Popular Computers

Drive a Hard Bargain!!™ For your TRS/80, Color Computer, IBM, Apple, Franklin

5 M.B.-20 M.B. Complete Systems . . . . . from $999.95

Diskette Breakthrough — 10 Pack in Library Case — $~~18.95~~ NEW LOW PRICE

SAVE!! PLEASE CALL FOR OUR MOST CURRENT PRICE REDUCTIONS.

TOLL FREE ORDERING
1-800-343-8841

GENERAL AND TECHNICAL
1-617-872-9090

Disk Drives (0123) TRS/80-IBM-Apple - TI Franklin-Max/80-LNW . . . . . ◄
Model I/III/IV Upgrade (Disk Drives - Memory) . . . . . ◄
Printers — Daisywheel/Dot Matrix . . . . . ◄
Percom Double Density Controller (Model I) . . . . . ◄
Color Computer Printer Interfaces . . . . . ◄
Disk Drive Operating Systems . . . . . ◄
Repair Services Now Offered — FAST Turn-a-Round . . . . . ◄
Apple/Franklin Compatible Add-On Drives with Case & Cable . . . . . ◄
Diskettes in Library Cases . . . . . ◄

CALL TOLL FREE FOR NEW PRICES

DISK DRIVE CASES AND POWER SUPPLIES . . . . . starting at $59.95
Printer **Buffers** 8K to 512K . . . . . starting at $143.95
Holmes Model I/III Speed-up Mod . . . . . starting at $90.00
Cables — Printer/Disk Drive . . . . . starting at $23.00

Warranty on Disk Drives — 6 Months to 1 Year

SOFTWARE SUPPORT, INC.

One Edgell Road, Framingham, MA 01701 (617) 872-9090

Hours: Mon. thru Fri. 9:30 am to 5:30 (E.S.T.) Sat. 10 am to 4:30 pm

DEALER INQUIRIES INVITED.

TERMS:
M.C./Visa/Amex and personal checks accepted at no extra charge.
C.O.D., please add $3.00.
Shipping: Please call for amount.
Not responsible for typographical errors.

CANADA
MICRO R.G.S. INC.
751, CARRE VICTORIA, SUITE 403
MONTREAL, QUEBEC, CANADA, H2Y 2J3
Regular **Tel. (514) 845-1534**
Canadian Toll Free 800-361-5155

Service! Service!

All in stock products are shipped within 24 hours of order. Repair/Warranty service is performed within 24 hours of receipt unless otherwise noted. We accept C.O.D., foreign and APO orders. School and D&B corporate P.O.s accepted.

TRS/80 Registered Trademark Tandy Corp. IBM-PC Registered IBM Corp. Apple Registered Trademark Apple Computer Corp.
Franklin Registered Trademark Franklin Corp. Max/80 Registered Trademark Lobo Int.

DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES

Circle 10 on Reader Service Card.

Circle 11 on Reader Service Card.

Circle 12 on Reader Service Card.

Bars and Plots

itself is capable of creating up to 16 colors. For each color, however, a number of new characters must be redefined for the top of the bars. Thus, you are limited to using only four of them.

After starting the program, you will be asked whether or not a printer will be used. If you reply Y for yes, the program will also ask you to enter your printer's parameters—e.g., RS232. The message PLEASE WAIT — GOTTA REST will appear while the computer prepares itself and defines the graphics shapes.

Next, the screen will clear and display the bar graph's grid. You will be asked to enter the maximum value for the graph. The program can handle a maximum value of 100,000. The minimum value is 20. The grid scale is rounded off to the next highest multiple of 20—i.e., if you enter a scale of 67, the scale used will be 80. This makes the calculations for character definition easier, and prevents messy values from appearing at the quarter, half, and three-quarters marks.

The next prompt will ask you to enter a side label. This label can be up to 24 characters long, and is displayed on the left side of the screen. You can then enter a label for the bottom of the grid. This label will be displayed right under the grid, and can be up to 25 characters long, which allows for one character under each bar.

Now you're ready to start entering data onto the graph. The asterisk cursor is under the active bar. You can enter any value from 0 to the maximum value of the chart, or one of the commands mentioned earlier in the text. If you enter a number, a bar will be displayed at that point. If there is already a bar at that point on the graph, then the new bar will replace the old bar.

If you make a mistake while entering a value, you can press [FCTN] 3 (erase), clear the entry, and start over. If you enter the wrong value, or you simply wish to move to a different bar position, use the [FCTN] key with S and D to move the cursor back and forth. Move the cursor to the bar on which you want to make an entry and enter data as you normally would.

Bars and Graphs (TI-99/4A) Explanation of the Program

Line Nos.	
100-160	Program header.
170-240	Display the title screen and input printer option.
250-280	Key input routine. Wait for the [ENTER] key, then clear the screen.
290-620	Initialize graphics and program variables.
630-730	Draw the grid on the screen for the bar graph.
740-760	Enter the maximum grid value.
770-920	Key input routine to simulate the input statement.
930-1020	Subroutines to move the bar cursor left and right.
1030-1070	Subroutine to build the input string.
1080-1130	Subroutine to erase an entry from the input routine.
1140-1180	Test the contents of the input string. If nothing was input, then use the last value entered.
1190-1390	Check the value entered for maximum value to ensure that it is legal.
1400-1460	Enter the side label for the graph.
1470-1560	Enter the bottom label for the graph.
1570-1780	Enter a bar color.
1790-1890	Routine to enter data onto the graph and accept commands.
1900-2550	Subroutines to handle the different commands.
2560-2690	Routine to print the graph.

HCM

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

VIC

The VIC-20 version of this program is much shorter because of its memory constraints. We have, however, squeezed as many of the same features into this shortened version as possible.

When you run the program, there will be a slight delay while the program variables are initialized, the graphics characters are placed into RAM, and their shapes changed. The bar grid is then displayed on the screen, and you are asked to enter the maximum value for the grid. The maximum value must be a multiple of 16 (e.g., 32, 48, 272) and can't excede 100,000.

If the value you enter is not a multiple of 16, then the program will round your entry off to the next highest multiple. For example, if you enter a maximum value of 263, the computer will round it off to 272 which is the next highest multiple of 16.

Multiples of 16 are used for two reasons: (1) it makes the calculations for displaying graphics easier, and (2) it prevents you from entering a maximum value that leaves fractions when divided by 2 or 4. You don't want these fractions to show up when the values along the left side of the grid are displayed, representing 1/4, 1/2, and 3/4 of full scale.

After entering the maximum value, you will be asked to enter the side label and bottom label, which will then be displayed with the grid. Next, you will be asked to choose one of three colors—red, blue, or green.

Now you're ready to start entering data onto your grid. When prompted with ENTER DATA, you can either enter a value—which will draw a bar on the graph—or one of several commands. A numeric value will produce a bar on the grid at the current bar cursor position. This is indicated by an asterisk below the bar. If the value you enter excedes the maximum value for the grid, the bar will be drawn to the top of the grid.

If you want to enter bars in an order other than from left to right, you can move the bar cursor to any position before making your entry. You may also wish to do this to make corrections to existing bars. To move the bar cursor, simply press R (for move right) or L (for move left) and then press [RETURN]. The cursor will move one position in the indicated direction.

A number of other commands are available from the ENTER DATA prompt. If you enter P and press [RETURN] you will get a printed copy of the screen from the system's printer. This is a default output to device #4. If you use a different device port for your printer, you may want to slightly modify your program in line 840. Here, we have assigned device #4 to channel #4 in the OPEN statement. You may change the device port to one of your choice.

A problem may arise if you use a printer that is not compatible with Commodore graphics. This program does a screen dump of the screen graphics, and uses the special Commodore graphics characters to create the graph on the printer. If your printer cannot print Commodore graphics, the results will be unpredictable.

You may change the color of the bars at any time. If you enter C on the ENTER DATA line, you will see the three color choices and can select one of them. After making your selection, you return to the data entry mode. Any bars created after changing the color will be drawn with the new color. Bars made before the color change will stay the same.

The NEW command lets you clear the screen and start with a completely new bar graph. The old bar graph erases from memory, as does the screen.

If you simply want to erase a single entry on the bar graph, use the ERASE command. Move the bar cursor to the bar you want to erase and enter ERASE at the ENTER DATA line. The bar will be cleared, and the bar cursor will remain where it is.

When you're ready to end the program, enter END on the ENTER DATA line. The program will halt, taking you back to the BASIC system.

Bars and Graphs (VIC-20) Explanation of the Program

Line Nos.	
100-170	Program header.
180-220	Initialize program graphics and variables.
230-310	Graphics data.
320-460	Enter maximum data value, side label, and bottom label.
470-540	Enter data and check for commands entered.
550-630	Display the bar on the screen.
640-660	Draw the bar grid on the screen.
670-690	Check for valid numeric entries.
700-750	Convert a string input to its poke values. Display either the side or bottom label.
760	Subroutine to erase a bar from the grid.
770-790	Subroutine to change the bar color.
800-830	Move the bar cursor to the right or left.
840-980	Routine to print a screen dump to the printer.
990-1000	Key scan routine.
1010	End of the program.

HCM

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

TRUCKLO

TEXAS INS

TEXAS INSTRUMENTS HAS EMPTIED THEIR WAREHOUSE OF ALL EXIST

WE NOW HAVE OVER 75,000 TI MODULES READY FOR SHIPMENT INCLUDING MORE! ORDER NOW AND RECEIVE A FREE MODULE OF YOUR CHOICE.

The BACH Company
760 SAN ANTONIO ROAD
PALO ALTO, CA 94303

Call us TOLL FREE at 800-227-8292 (incl Alaska & Hawaii), in California call 415-424-0800.

YOUR CHOICE-$14.95

PHM 3023 Hunt the Wumpus
PHM 3030 A-Maze-Ing
PHM 3052 Tombstone City:21st Century
PHM 3050 TI Invaders
PHM 3054 Car Wars
PHM 3057 Munch Man
PHM 3056 Alpiner
PHM 3110 Chisholm Trail
PHM 3112 Parsec
PHM 3031 The Attack
PHM 3032 Blasto
PHM 3034 Hustle
PHM 3022 Personal Real Estate

YOUR CHOICE-$19.95

PHM 3007 Household Budget Management
PHM 3002 Early Learning Fun
PHM 3003 Beginning Grammar
PHM 3009 Football
PHM 3018 Video Games I
PHM 3036 Zero Zap
PHM 3038 Connect Four

PHM 3004 Number Magic
PHM 3005 Video Graphs
PHM 3010 Physical Fit
PHM 3114 Alligator
PHM 3115 Alien Addi
PHM 3118 Minus Mis
PHM Meteor Multipli

YOUR CHOICE-$24.95

PHM 3016 Tax/Invest.
PHM 3020 Music Maker
PHM 3015 Early Readin
PHM 3043 Reading Fun
PHM 3046 Reading On
PHM 3047 Reading Roun
PHM 3048 Reading Rall
PHM 3082 Reading Flig
PHM 3027 Addition &
PHM 3028 Addition &
PHM 3929 Multiplica
PHM 3049 Division I
PHM 3094 Integers
PHM 3095 Fractions

SUMMER SPECIAL

DURING AUGUST & SEPTEMBER **THE BACH COMPANY** IS OFFERING FREE TI MODULES. JUST PURCHASE THE FOLLOWING AMOUNTS OF MERCHANDISE AND CHOOSE YOUR FREE MODULE FROM THE CORRESPONDING CATEGORY:

1. ORDER $100 WORTH OF MERCHANDISE AND RECEIVE ONE $14.95 CARTRIDGE OF YOUR CHOICE FREE

AD SALE!

TRUMENTS

ING TI-99/4A SOFTWARE AND ACCESSORIES - RIGHT INTO OURS!

MANY NEW TITLES: JAWBREAKER, FACEMAKER, HOPPER, BURGERTIME, AND

SEE DETAILS BELOW.

ness
Mix
tion
sion
cation

ecord Keeping

up

Subtraction I
Subtraction II
tion I

PHM 3042D Tunnels of Doom (Disk)
PHM 3042T Tunnels of Doom (Tape)
PHM 3131 Moonmime
PHM 3041D Adventure (Disk)
PHM 3041T Adventure (Tape)
PHM 3014 Statistics
PHM 3035 Terminal Emulator II
PHM 3194 Jawbreaker
PHM 3050 Numeration I
PHM 3051 Numeration II
PHM 3092 Multiplication
PHM 3097 Percents
PHM 3098 Number Readiness
PHM 3099 Laws of Arithmetic
PHM 3100 Equations
PHM 3101 Measurement Formulas
PHM 3145 Sneggit
PHM 3083 Computer Math Games II

YOUR CHOICE $34.95
PHM 3146 Munchmobile
PHM 3177 Facemaker
PHM 3233 Burgertime

NEW!

PHM 3189 Return to Pirate Island
PHM 3229 Hopper

NEW!

YOUR CHOICE-$44.95
PHM 3059 Scholastic Spelling 3
PHM 3060 Scholastic Spelling 4
PHM 3061 Scholastic Spelling 5
PHM 3062 Scholastic Spelling 6
PHM 3055 Editor/Assembler
PHM 3040 Logo I

All Others
PHM 3111 TI Writer.....$109.95
PHM 3113 Microsoft Multiplan....$109.95
PHM TI Logo II.....$109.95
PHM Mini Memory.....$ 79.95
PHM Black Jack & Poker.....$ 15.95
PHM 3006 Financial Decisions....$ 15.95
PHM 3039 Yahtzee.....$ 21.95
PHM 3033 Black Jack & Poker.....$ 15.95

2. ORDER $200 WORTH OF MERCHANDISE AND RECEIVE ONE $24.95 CARTRIDGE OF YOUR CHOICE FREE

3. ORDER $300 WORTH OF MERCHANDISE AND RECEIVE ONE $44.95 CARTRIDGE OF YOUR CHOICE FREE

CALL US **TOLL FREE** AT **800-227-8292** (INCLUDING ALASKA & HAWAII),
IN CALIFORNIA CALL **415-424-0800**

Circle 15 on Reader Service Card.

Snap-Calc . . . from p. 14

REQUIRED EQUIPMENT: TI Extended BASIC

The TI-99/4A is the machine on which *Snap-Calc*™ was originally designed. The TI version differs from the other versions because of the 99/4A's keyboard layout and screen size incompatibilities.

The Keyboard

The TI keyboard does not have separate function keys. On the 99/4A, you select functions by pressing the [FCTN] key in the lower right corner of the keyboard, in conjunction with the numeric keys. The functions perform as follows:

Press [FCTN]	for FUNCTION
1	Load data from disk or cassette.
2	Save data to disk or cassette.
3	Cancel entry/erase.
5	Calculate the logic model.
6	Print report.
7	Logic entry mode.
9	Exit *Snap-Calc*™.

In addition, several other keys work with the [FCTN] key to move the cursor:

KEYS	FUNCTIONS
[FCTN] E	Move the cursor up 1 cell.
[FCTN] S	Move the cursor left 1 cell.
[FCTN] D	Move the cursor right 1 cell.
[FCTN] X or [ENTER]	Move the cursor down 1 cell.

The [CTRL] key is useful for moving the entire window of the spreadsheet.

Press [CTRL]	for FUNCTION
E	Move up 5 cells.
S	Move left 3 cells.
D	Move right 3 cells.
X	Move down 5 cells.

Screen Size

The 99/4A is capable of printing only 28 columns, using the PRINT and DISPLAY AT commands. This limits screen displays of row labels to nine characters wide, and numeric entries to seven characters wide—including a decimal point.

Data Storage

The 99/4A is capable of saving and recalling information from both a cassette recorder and a disk drive. Either device will work with this program. When you select either SAVE or LOAD, you must decide whether you want to work with the spreadsheet data or the logic model, which are kept in two different files.

After making your selection, you are prompted to input the device name. If you are using a cassette, simply type CS1 and press [ENTER]. For a disk drive, enter DSK1 or DSK2. It is not necessary to enter the period separator or the file name at this time, because the computer will just ignore them.

If you selected a device other than a cassette, you will also be prompted for the file name. The only exception to this is when you are saving the logic model. In this case, if you have already assigned a logic name, the logic model will automatically use that name for the file. If no name has been assigned, or you are loading the logic model, you will be prompted for the file name. The file name will then be used as the logic model name. When saving or loading data you are always asked for the file name.

Reports

The 99/4A can work with a maximum of only 13 columns because of its memory limitations, so this program is designed to print all 13 columns on one page. To do this, it is necessary to set the printer to condensed mode, which allows an 80-column printer to print 132 columns. This is done in line 1440. CHR$(15) sets condensed mode on most printers. If your printer requires a different code to enter condensed mode, however, you may need to change line 1440 to your printer's specifications. If you are using a letter-quality printer, or one of the less expensive dot matrix printers, you may not be able to print in condensed mode.

Logic Mode

The maximum string length on the 99/4A is 127 characters. You are limited to this size when you enter commands. If the calculation you enter gets too big, it should be broken down into smaller parts.

It's in this mode that you set up your logic model. It can include row labels, a totals column, a last column, a name for your model, and row equations. If you assign a row name longer than nine characters, it will be truncated. All spaces are removed. If you want spaces in your row name, you can enter them directly in Data Entry mode.

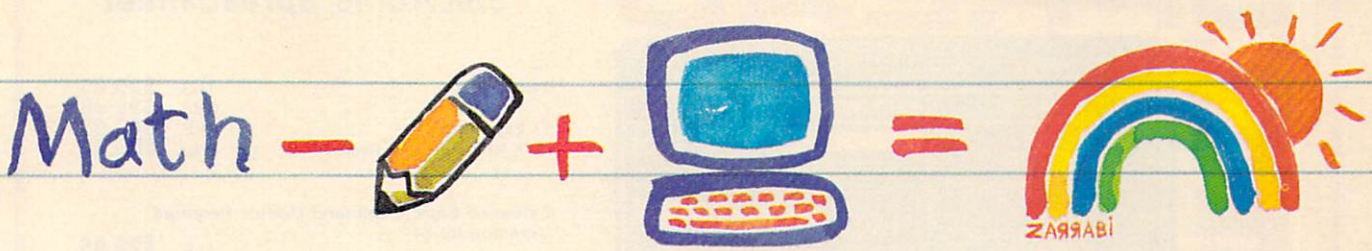
The maximum number of rows you can use for data or equations is 20, and the maximum number of columns is 13. But, if you run the program with the 32K memory expansion connected, you could easily modify the program to accept more columns and rows. The modifications are simple using the following procedure:

In line 190 change the value assigned to A to the number of rows you want. Then change the value assigned to B to reflect the number of columns you want. You also need to change the values used to dimension the arrays in this same line. D$( ) and E$( ) need to be expanded to the number of rows you wish to use. The J( ) array should have its first value set to the number of rows you wish to use, and the second value set to the proper number of columns.

SNAP-CALC (TI-99/4A) Explanation of the Program

Line Nos.	
100-160	Program header.
170-200	Initialize program variables and functions.
210-280	Main control loop for data entry mode.
290-300	Exit program routine.
310-320	Subroutine to clear an entry and display the new value.
330-470	Subroutine to calculate the logic model.
480-490	Subroutine to return the value to be used in the equation.
500-690	Control loop for logic entry mode.
700	Subroutine to return a numeric value from the entry string.
710	Subroutine to delete the cursor.
720-750	Subroutine to take entry string apart by placing each word into a cell of the FS() array.
760-910	Subroutine to rebuild the entry string for listing to the screen or printer.
920-1090	Subroutine for file handling.
1100-1120	Builds the row names from the entry string.
1130-1140	Creates a cell value for the spreadsheet during data entry mode.
1150-1160	Control loop repaints the data entry screen.
1170	Routine to display the current mode.
1180-1190	Input the logic entry string.
1200	Direct output to either the screen or the printer.
1210	Key scan routine.
1220	Relocate the cursor.
1230-1250	Key scan routine waits for a different key to be pressed before continuing.
1260-1300	Display the data entry screen.
1310-1390	Move the cursor up, down, left, and right one position.
1400-1490	Print the report.
1500-1520	Image formats for printing single items.
1530-1540	Clears the logic model from memory.
1550-1570	Error routine.
1580-1590	Image formats for a full screen.
1600-1620	Open and close the printer port. HCM

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.



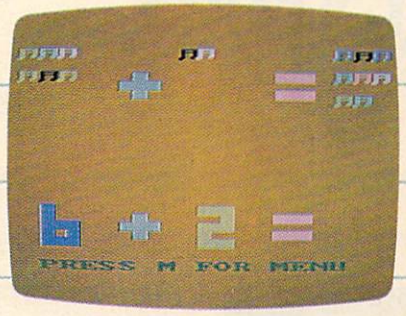
Elementary Addition and Subtraction

by Mark Dewese
and the HCM Staff

In the previous issue of *Home Computer Magazine* we introduced a VIC-20 version of *Basic Addition and Subtraction*, an educational program for preschoolers. Now for parents of preschoolers with either a Commodore 64 or TI-99/4A, we present two additional versions.

Basic Addition and Subtraction makes learning the basics of addition and subtraction an entertaining experience for a small child. The program offers simple problems (with answers ranging from zero to nine) on three levels of difficulty.

You help your child get started by selecting either addition or subtraction from the first menu. The next menu lets you choose one of the three difficulty levels. The easiest level is aimed at children just learning to count; each problem is accompanied by a graphic representation of the problem—including a graphic answer. The next level includes graphics of the problem, but the answer's graphics do not appear until the child indicates an answer. The hardest level displays only numeric problems.



To enter an answer, the child presses any one of the number keys. The computer evaluates the answer and provides immediate feedback. As with most good educational software, the child is rewarded for selecting the right answer—in this case, with a little tune and colorful graphics.

If the answer if not correct, the computer erases the child's incorrect answer and offers another chance. Following a correct answer, the screen is erased and a new problem appears. At any point you may change modes by pressing M and returning to the first menu.

The TI version of *Basic Addition and Subtraction* will run with either BASIC or Extended BASIC. If you use Extended BASIC and have the Speech Synthesizer, you can use the speech option part of the program, which is displayed on the first screen. If you have both Extended BASIC and the Synthesizer, press S. If you don't have one or the other, then press any other key to turn the speech option off. If you don't have those items and press S anyway, you will get an error message, and the program will halt the first time it tries to speak.

If you enter the program under BASIC and later attempt to run it under Extended BASIC, you will get a syntax error in the speech routines. This is caused by the double colons in the lines of code that utilize the speech option. In Extended BASIC, the colons are interpreted as a break between statements for multiple statement lines. The only time the colon becomes a legal operator in BASIC is within the PRINT statement to designate a carriage return. If you later load the program under Extended BASIC you will find spaces between each of the colons, making them illegal operators for Extended BASIC.

Even if you are using BASIC and don't have a Speech Synthesizer, enter all of the lines in the program—don't skip those that only work with speech. There is important program logic in these lines and the program won't work properly without them.

Basic Addition and Subtraction
(TI-99/4A)
Explanation of the Program

Line Nos.	
100-180	Program header.
190-310	Title screen.
320-540	Initialization and opening.
550-670	First menu.
680-830	Second menu.
840-920	Begin displaying problem.
930-1150	Define graphics characters.
1160-1250	Choose graphic.
1260-1700	Select and display left number.
1710-1750	Print sign.
1760-2330	Select and display right number.
2340-2430	Get answer.
2440-2560	Draw graphic of answer.
2570-2630	Get player's input.
2640-2680	Make computing noise.
2690-2860	Evaluate answer and branch appropriately.
2870-2950	Print final score.
2960-2980	Right answer noise.
2990-3720	Print answer.
3730-4270	Speech routines.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

On the Commodore 64, when you want to put characters on the screen memory by POKEing them directly into video memory, you have to give the character a color by POKEing a corresponding color memory location. This means that you have to constantly update and keep track of two addresses. It is much simpler, however, to derive a color memory address directly from the character memory address. You can do this by initializing a variable to a value that will be added to the character memory address. Consider the following BASIC lines for example:

```
10 S=54272:AD=1024
20 FOR I=0 TO 999
30 POKE AD,1:POKE S+AD,2
40 AD=AD+1:NEXT
50 END
```

This short program is a simplified version of the POKE graphic character subroutines in the main program. It POKEs red A's on the screen beginning in the upper left corner. Line 10 initializes a constant (S) and a variable (AD). S is used as an *offset* to color memory. AD is the character address. AD (1024) is also the starting address of the screen in memory. The value for S is found by subtracting 1024 from the address of the first

Continued on next page

Elementary

color position (55296-1024=54272). Line 20 begins the program loop and line 30 POKEs the character and color to the character and color memory respectively. In this way the correct color memory location is derived by adding the character memory address to the constant S.

Line 40 updates the character memory address in preparation for POKEing the next position, and then returns control to the beginning of the FOR__NEXT statement in line 20. This programming concept is very useful when you are POKEing different locations on the video screen and need to find the corresponding color memory addresses.

Basic Addition and Subtraction (C-64)
Explanation of the Program

Line Nos.	
100-170	Program header.
180	Read sprite definitions.
190-210	Initialization.
220	Read in sound data.
230-300	Sprite data.
310-320	Sound data.
330-450	Display program prompts and messages.
460-660	Derive and display left and right numbers.
670-910	Input answer, evaluate, and reward.
920-940	Get keyboard input.
950-1020	Sound subroutine.
1030-1210	Print subroutines to position numbers on screen.
1220	Print plus sign.
1230	Print equal sign.
1240	Print minus sign.
1250	ON GOTO statement for numbers.
1260-1300	Position cursor subroutines.
1310-1390	Poke graphic character subroutines.
1400-1410	Clear portions of screen subroutines.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

.....Texas Instruments TI-99/4A — COMPUTERS, COMPONENTS AND SOFTWARE.....

TEX-COMP * P.O. BOX 33084 * GRANADA HILLS, CA 91344* (818) 366-6631

TEX-COMP™

TI USERS SUPPLY COMPANY

TEXAS INSTRUMENTS COMPUTING PRODUCTS AUTHORIZED SALES

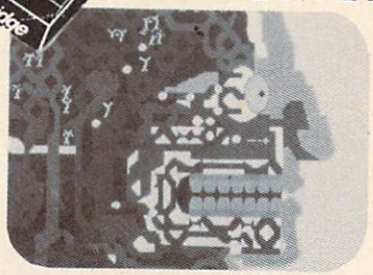
America's Number One TI computer retailer

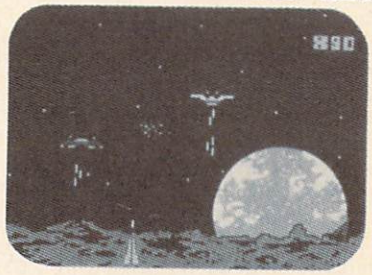
TI-99/4 PRODUCTS AT PROFESSIONAL PRICES

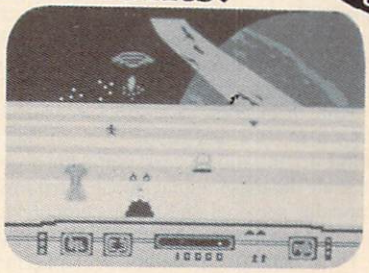
ATTENTION TI-99/4 USERS TEX COMP HAS BEEN SUPPORTING THE ENTIRE 99/4 PRODUCT LINE TO PROFESSIONAL AND INDUSTRIAL ACCOUNTS SINCE THE INTRODUCTION OF THE TI-99/4. NOW YOU CAN PURCHASE ALL YOUR TI-99/4 REQUIREMENTS FROM ONE SOURCE AT LOW "PROFESSIONALLY DISCOUNTED" PRICES ON THE **ENTIRE** 99/4 PRODUCT LINE. WE ARE A DIRECT TI NATIONAL ACCOUNT AND DEAL DIRECTLY WITH TI. ASSURING YOU THE BEST SERVICE, EARLIEST SHIPMENT AND LOWEST POSSIBLE PRICES

AUTHORIZED DEALER

Action • Adventure • Education • Entertainment
Excitement • Thrills • Chills • Danger
Shoot 'em ups • Knock 'em downs.
All new. All from Texas Instruments.







Microsurgeon
speech

The patient is critically ill! Check the patient's status chart to see which vital organs need immediate attention. Then navigate the tiny robot probe through the blood stream, outmaneuvering white blood cells that attempt to destroy you as foreign matter. Hurry—every second counts.

Demon Attack®
"Game of Year"

A staggering array of winged warriors wreaks havoc from above! Only your laser cannon stands between you and oblivion! Blast those bizarre demons to smithereens. Your arsenal includes missiles—and sheer intestinal fortitude. Are you equal to this colossal challenge? You'd better be!

Moonsweeper™

Miners are trapped on Jupiter's moons. Meteors and asteroids hurtle past you. Choose one of the four types of moons and land if you can — or dare!

Navigate your ship across the lunar surface. Radar shows where the miners lie. Enemy at 2 o'clock! Destroy them! Then lift off and look for other moons and trickier trials.

It's the year's best software at the year's best prices

TEX-COMP HAS PURCHASED ALL OF TI'S REMAINING INVENTORY OF THESE THREE TOP GAMES BY IMAGIC AND IS PASSING THE SAVINGS ON TO YOU. THESE GAMES REPRESENT THE "STATE OF THE ART" IN GRAPHICS, COLOR, AND ACTION AND WERE INTRODUCED AT A $39.95 RETAIL. LOOK FOR THESE AND MANY OTHER SUPER VALUES COMING YOUR WAY FROM TEX-COMP. SEND $2.00 FOR OUR CURRENT CATALOG AND ORDER KIT AND WE ENCLOSE A $5 CERTIFICATE. NEW 99/4A PRODUCTS INCLUDE INFOCOM SOFTWARE INCLUDING ZORK I, DATABAR BAR CODE READER AND SOFTWARE, RS232 STAND ALONE FROM CORCOMP, THE NEW WIDGET FROM NAVARONNE, AND THE SAKATA COLOR MONITOR. ALL AT THE LOWEST DISCOUNT PRICES. WE ARE DEDICATED TO SUPPORTING THE 99-4A USER.

$19.95 EACH

VISA and MASTERCARD HOLDERS CALL DIRECT:

[818] 366-6631

SEND ORDER AND MAKE CHECKS PAYABLE TO:

TEX-COMP™ P.O.BOX 33084 — GRANADA HILLS, CA. 91344

TERMS: All prices F.O.B. Los Angeles. For fastest service use cashiers check or money order. Personal checks take at least ten days to clear. Add 3% shipping and handling ($3.00 minimum). East of Mississippi 4½%. (Free shipping on all software orders over $100 and on postpaid specials). Add 6½% S.T. for delivery in Calif. Prices and availability subject to change without notice. We reserve the right to limit quantities. *this ad supercedes all previous ads.*

Credit Cards add 3%

MasterCard VISA

TEX-COMP IS THE LARGEST EXCLUSIVELY TI HOME COMPUTER DEALER IN THE WORLD. WHAT WE ADVERTISE, WE STOCK. TEX-COMP MAINTAINS A $100,000.00 TI INVENTORY AT ALL TIMES AND IS A DIRECT TI DEALER.
WHEN IN SOUTHERN CALIFORNIA VISIT OUR MODERN WAREHOUSE OUTLET STORE WHERE YOU CAN PURCHASE THE ENTIRE TI LINE AT OUR REGULAR DISCOUNT PRICES.

Circle 52 on Reader Service Card.

PFS:Report

PFS:File

Name:	PFS:FILE and PFS:REPORT
Program Type:	Database manager
Machines:	Apple II+, IIe, IIc, IBM PC
Distributor:	Software Publishing Corp. 1901 Landings Dr. Mountain View, CA 94043
Price:	PFS:FILE $125 for Apple version; $140 for IBM version PFS:REPORT $120 each

System Requirements:
For Apple: 48K; second disk drive; printer for reports
For IBM: 128K RAM, minimum 1 disk drive, 2 drives preferable

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

DATABASE BASICS

If you're new to computing, the term "database management" can be intimidating—conjuring up the esoteric, complex data processing of big corporations and agencies. But all that the term "database management" really describes is the storage and retrieval of information—tasks at which the computer is especially good. Programs thus designed to store and retrieve are known generically as "database managers." True, they are used most often for the management of huge amounts of information. The Internal Revenue Service, for instance, would be hard-put to collect taxes if it were not for powerful computers and the database software running on them. But the use of database management programs can be appropriate for personal applications too. You can use them to create home or business inventories, magazine article indexes, mailing lists, client lists, and recipe files—in short, for any kind of information organizational task you want to make easy. Dozens of off-the-shelf database management programs are available for the more popular personal and home computers, such as the Apple II family, the Commodore 64, the TI-99/4A, and the IBM PC and PCjr. These commercial database management programs may be divided into four general categories, or levels:

RAM-Based: The entire file, containing all the records, is loaded into memory at one time. The size of the file is limited to the computer's memory size. Its advantage is that this immediacy generally makes program operation extremely fast; after the information is in memory, no further disk access is required. This type of database software is usually inexpensive, from $25 to $75. Examples: *Modifiable Database* (Synergistic Software), *Notebook* (D.C.Heath).

Single Data-Disk: The file is stored on one data disk, and records are accessed by the program as they are needed. This permits a larger file size than RAM-based programs do, but the file cannot be larger than will fit on a single data disk. This type is usually priced in the $100 to $250 range. Examples: *Visifile* (Personal Software), *PFS* (Software Publishing Corp).

Multiple Data-Disk: This level of database management program permits one file to be stored on several data disks, which affords still greater file size—thousands of records. Prices are in the $200 to $400 range. Example: *DB Master* (Stoneware).

Complex Database Systems: These "interactive relational-database" systems allow relationships between files, and are so powerful and complex that they are very nearly programming languages in themselves. They are the most expensive of all, from $600 to $1000. Example: *dBase II* (Ashton-Tate).

Generally, database management programs perform the same types of operations to manipulate information: data entry, storing records, retrieving records for viewing or correction, deleting records, sorting records, searching for specific information using search "keys" or criteria, performing limited math operations on the data, and creating reports using the data.

We'll use the example of a teacher's class list to explain three terms you'll encounter in discussions of database management:

Field: The smallest element of information, a basic item such as the student's "Name:" or "Address:" (or anything specified when you set up the organization of the data).

Record: Contains all of the fields of information, such as the information on one student.

File: Contains all records, such as all information on all students.

—Michael D. Brownsworth

A Review by Michael D. Brownsworth

The database management program, *PFS:FILE*, is the leading member of what Software Publishing Corporation calls their integrated "PFS Family of Software." The "family" also includes *PFS:REPORT*, which extends *PFS:FILE*'s report printing capability; *PFS:GRAPH*, which creates bar, line, and pie charts; and *PFS:WRITE*, a word processing program. The term "integrated" as used here means that some of the programs can directly share data between them. For example, *PFS:WRITE* can use data from *FILE* for insertion in form letters, or merge bar charts created with *GRAPH* into the body of a document. And, of course, *REPORT* creates custom-formatted printed reports with data from *FILE*.

PFS:FILE and *PFS:REPORT* together form a moderately powerful and flexible database management system. Each program would be fairly limited without the other, and for that reason most database management systems include both such modules in one package. The PFS people, however, offer them separately.

The PFS software runs on both the Apple II+, and IIe, and on Apple-equivalent systems such as the Franklin. It is also available for the IBM PC, PCjr (forthcoming), and IBM PC-equivalent systems (Columbia, Compaq, Corona, TI Professional, and Hyperion). A version of *PFS:FILE* has just been announced for the new Apple Macintosh. The Apple II+ version is available only in 40-column screen format; the IIe version does support 80 columns if the machine is equipped with an 80-column card. [The program will also run on the newly announced Apple IIc. —Ed.]

Purchasers of each package receive a manual and two copies of the program disk. *REPORT* includes one extra disk called *SORTWORK*, used for record-sorting workspace. The disks in the Apple versions are protected against multiple copying, but Software Publishing will replace a damaged disk for a nominal charge. The manuals are easy to understand and provide helpful examples and screen illustrations.

PFS is a single data-disk system; the file sizes are limited to the storage space of one data disk. With the Apple versions, a file can theoretically hold up to 1000 records. The IBM versions specify up to 1100 for the 160K disks and 2200 for the double-density 320K disks. However, these would be very simple records indeed—names only, perhaps. The actual record storage capacity depends upon the number of fields in each record and how much data is entered for each field. The manual indicates that the average mailing label (name, street address, city, state, zip) would allow only about 500 records per file. Of course, the single data-disk limitation can sometimes be minimized by dividing a large file into smaller logical groups (A - L and M - Z alphabetically, for example)—each of which should be small enough so as to be contained on one data disk. *PFS* does permit more than one file to be stored on a single disk, but the files will compete for storage space until the disk is full.

Using PFS:FILE

The main functions of *PFS* programs are selected from menus, so descriptions of menu options will offer the best understanding of how *PFS:FILE* works.

1.DESIGN FILE offers two options: CREATE FILE sets up a new file by designing a record form on screen (with designation and placement of field labels). The form may have multiple pages, but the total number of fields is limited to 100. CHANGE DESIGN allows a form design to be changed; the redesigned form can receive data transferred from the old design to the fields that remain the same.

2. ADD presents you with a form that is blank except for field labels. You add new records by moving to the appropriate field and filling in data, just as if the screen were a paper form. Status messages at the bottom of the screen display the file name, record number, and percentage of the file that has been used.

3. COPY offers three options—all requiring a second disk drive. COPY DESIGN ONLY copies the form design of an existing file to a new file that uses the same form design as the first (if, for instance, the first file is full). COPY SELECTED FORMS splits or merges files by copying selected records from one to the other. This useful feature provides flexibility. In the Apple version, you can copy from a 40-column to an 80-column file, but not the reverse. COPY WHOLE DISKETTE copies the data disk.

4. SEARCH/UPDATE allows you to search through your file to find records for review, correction, printing, or removal from the file. Search criteria are entered in the field in which a match is desired. There are five types of search criteria (or "retrieve specifications," as *PFS* calls them): (1) the full item match; (2) partial item match; (3) substring match, employing "wildcard" symbols and @); (4) numeric range matches; and (5) the "not" match. Search time is fast for full item matches in the first field position, but slows down when the other search specifications operate.

5. PRINT includes PRINT FORMS, which enables you to (a) choose forms to print, (b) choose which items on a form to print, and (c) choose how these items are to be printed. Forms are selected by the same specifications for retrieval used in SEARCH/UPDATE. Print formatting is limited within this option—suitable only for something as simple as mailing labels. You may save a print format by using DEFINE PRINT SPEC.

6. REMOVE allows you to remove selected forms from the file. Once forms are removed, their spaces are reclaimed, but the form number is not used again. It is possible to copy the form design and selected files to regain sequentially-numbered records.

Using PFS:REPORT

PFS:REPORT is an indispensable companion for *PFS:FILE*, providing the database manager far more power and flexibility in formatting reports from *FILE*'s data. *REPORT*'s menu options follow:

1. PRINT A REPORT prints reports with the records in a column format of up to nine columns, one field per column. Sort capability is provided on the first two columns—either alphabetically in ascending order, or numerically in descending order. Calculations you can perform on the columns include subtotals, totals, counts, and averages. In addition, you can derive up to three columns of numeric data from other columns. All numeric data is right-justified; alphanumeric data is left-justified. Decimals are always aligned. You may specify a report title of up to 30 characters. Reports may either be displayed on the screen or sent to a printer, but no provision exists to save them directly to disk— a useful option which, unfortunately, is not present.

2. PRE-DEFINE A REPORT allows up to eight report formats to be defined and stored to disk.

3. SET NEW HEADINGS permits new headings to be used instead of the field labels that normally head columns.

Registered PFS purchasers become members of the PFS: User Group and receive copies of *PFS: NEWS*. This newsletter consists primarily of new PFS product announcements, but also offers order forms for *PFS:SOLUTIONS*—pre-defined recordkeeping applications for *PFS:FILE*. Some of the ready-made applications include Home Budget, Home Inventory, Mail List, Disk Library, Employee, Payroll, Ledger, Invoices, Checks, and Tickler. *PFS:SOLUTIONS* sells for $20 each.

Flies in the Files

So far, *PFS:FILE* and *REPORT* may sound like the perfect database system. Unfortunately, the software falls short in an all-too-common area—the user interface. Ideally, the user interface should provide a natural, logical operation of the program for the user. If a program forces the user to adapt to unusual key sequences for commands and operations, or if it fails to check for proper input, then the user interface is poorly designed. On both counts, these are precisely the problems with the PFS database management system.

The programs never use the [RETURN] key to enter a typed line or to begin or continue a function: the IBM version uses a function key; the Apple version uses [CTRL][C]. In the Apple version, [CTRL][C] was an especially bad choice because in BASIC [CTRL][C] is the command to halt program execution—the "break" command. This halt command quickly becomes second nature to anyone who has used BASIC in even a limited way. True, *PFS* was written in Pascal, not BASIC, but it is a questionable practice to disregard such an entrenched convention and have it mean "go ahead" instead of "break." It would have been better to use [RETURN], the standard key for entering data.

Furthermore, on the Apple IIe and IBM versions, the [TAB] key (used in advancing to the next field) is located directly under the [ESC] key. If you should unintentionally press [ESC], you are returned to the menu, and all data in the record on the screen is thrown away. Data will inevitably be lost in this fashion.

On the Apple II+ version, the keys that move the cursor (T, F, V, G) seem arbitrarily chosen; it would have been much better to use the standard Apple II+ cursor-movement diamond—I,J,K,M.

The worst feature of the user interface design is that it does *not* check for correct input. While on a menu screen, if a user happens to press the wrong key (for example, the unused [RETURN] key), the cursor will jump "out in left field" to a corner of the screen. At that point, the program is perfectly willing to let the naive user type anything on the screen—even fill it up with characters that quickly overwrite all other printing on the screen! To prevent this unnecessary catastrophe, input should always be checked, and on-screen printing should be controlled.

The *PFS* database system is, otherwise, a flexible, moderately powerful package that performs rather well. It is, therefore, a shame that this product should to be marred by these user-interface eccentricities. *[Editor's Note: Updated versions of these PFS programs have recently been created by Software Publishing Corp. to run on both the Apple IIe and the new Apple IIc. For a limited time, the exchanges will be transacted for free; following that time period, exchanges will cost the usual $35. For details on obtaining updated PFS products, contact the Customer Service Department of Software Publishing Corp., or your local dealer.]* HCM

UNISOURCE
ELECTRONICS, INC.
for your Texas Instruments TI 99/4A Home Computer

WHAT IS UNISOURCE?

- The leading mail order supplier of TI-99/4A equipment and software
- A specialist in the TI-99/4A Home Computer
- Has availability of over a thousand different items for the TI-99/4A
- Provides a complete Encyclopedia/Catalog at a nominal charge (or free with an order) that describes all of the items
- Offers discount prices to give you more buying power
- Provides a toll free order line for your convenience (please, all calls other than orders should be directed to the office)
- Accepts MasterCard or Visa charge cards at no additional charge for your shopping convenience
- Normally ships in-stock merchandise to you within 48 hours

TALK TO YOUR COMPUTER
Voice Recognition: MBX EXPANSION SYSTEM

200 IN STOCK!

This system is the only one providing voice recognition available for the home computer. Works with 10 software packages developed by Milton Bradley. Includes Control Unit, Joy Stick and Microphone Headset **$99.95**

SOFTWARE - $29.95 ea.

Terry's Turtle Adventure, Honey Hunt, Championship Baseball, Sewermania, Meteor Belt, I'm Hiding, Soundtrack Trolley, Space Bandits, Bigfoot & Superfly.

HIGHLY AFFORDABLE DOT MATRIX

$179.95  $179.95

SEIKOSHA GP-100 ..... only **$179.95**
your package choice...the Axiom or RS232 card and cable ..... only **$279.95**

Simplicity of operation and proven reliability, together with the Direct-Connect package makes the GP-100 the perfect starter printer. Multiple copies, adjustable tractors, and surprisingly good graphics add to the bargain. Another plus - you can daisy chain other peripherals, i.e. disk drives and memory units. (requires Axiom parallel connector or RS232 card for P.E. Box and parallel cable...your choice...$100.00 with purchase of printer.)

FEATURES GALORE

GEMINI 10X

In addition to the features on the most popular printer, it has:

- 120 cps Print Speed, compared with 80 cps
- Friction and Tractor Feed
- Three additional built-in character fonts, plus design your own character fonts...total of nine
- $4.00 Replacement Ribbon instead of $12.00 Replacement Ribbon

Package #1: (for TI Peripheral Expansion Box) Gemini 10X printer with RS232 card and cable.
..... only **$399.95**

Package #2: Gemini 10X printer with Axiom direct connector ..... only **$389.95**

LOWEST PRICE EVER

SMITH CORONA TP-I
only **$299.95**
While They Last!!

Letter quality Daisy Wheel, 12 cps, Serial hook up package with RS232 card and serial cable ..... only **$399.95**
(LIMITED QUANTITY)

Now your letters, documents, forms, reports, price lists, data sheets and computations can all be printed with the quality you deserve...at a readily affordable price.

DISK/MODULE STORAGE

Storage for disks or modules and cassettes. From Innovative Concepts.

Flip'N File	Holds 10 Disks	**$ 5.95**
Flip'N File	Holds 15 Disks	**$ 9.95**
Flip'N File	Holds 50 Disks	**$23.95**
Cassette'N Game File	Holds 18 Cass. or Modules	**$19.95**

DISKETTES

Lowest Price Ever

5¼" floppy diskettes from Elephant.
Lifetime warranty; hub-ring; soft-sectored.
SS/SD **$19.95**/box of 10
DS/DD **$29.95**/box of 10

PRINTER CONNECTION BREAKTHROUGH

ParallAx TI by AXIOM

Hook up your parallel printer **without a P.E. Box!!** You can even connect to additional peripherals, nothing else required.

only **$99.95**

- Connects any standard Centronics parallel printer to the TI 99/4A
- Self-powered, works with or without Peripheral Expansion System
- Compact, easy to install and use. Runs all standard software

CENTRONICS COMPATIBLE PRINTER CABLE

only **$29.95**

Custom made 8' parallel cable that connects the parallel port of the TI RS232 card **to any Centronics compatible printer,** i.e. Gemini, Epson, etc.

NEW MUSIC SYNTHESIZER PERIPHERAL CARD

NEW!! NEVER BEFORE AVAILABLE!

THE BAND by Texas Peripherals, Inc. ..... only **$299.95**

THE BAND will provide 12 voice, 2 channel stereo output with software control of attack, decay, sustain and release of each channel. A composing system, on Disk, is included to allow you to program the system as a synthesizer. THE BAND is not usable from BASIC or other programming languages but is meant as a tool to turn the TI 99/4A into a first class music synthesizer for the musician or hobbyist. Required hardware includes disk drive, 32K memory and Editor/Assembler.

PERIPHERALS NOT REQUIRING P.E. BOX

DISK DRIVE SYSTEM by PerCom Data (Limited Quantity) ..... only **$349.95**
(available now - includes built in disk controller)
Nothing else required! One sleek, plug- compatible disk drive system...COMPLETE. Single density, single sided, 92K byte (formatted), 5¼" floppy disk drive system designed specifically for the TI 99/4A.

PARALLEL RS232 by AXIOM ..... only **$99.95**
(available now - parallel port only)

32K MEMORY EXPANSION BOX by Boxcar ..... only **$129.95**
(expected March availability)
Plugs directly into console and can be used with other peripherals

- Daisy chain compatibility with the computer's built-in connector without extra cables
- Durable and attractive enclosures
- State of the art design and reliability
- Easy to use installation/operation manual
- 90 day limited warranty

BOXCAR

SIGNALMAN MARK III MODEM

A low cost, direct connect, high quality and super reliable TI 99/4 and 99/4A compatible modem that comes complete and ready to use - just plug it into a TI RS232 **$89.95**

$100 SPECIAL - FREE Source/TEXNET sign on and first hour, with purchase of Signalman Mark III

ACCESSORIES: 9V-DC Power Supply **$9.95**

SIGNALMAN MARK VII MODEM

300 Baud Auto Dial/Auto Answer **$159.00**

SIGNALMAN MARK XII MODEM

1200 Baud Auto Dial/Auto Answer **$349.00**

GROMBUSTER $39.95

If you have one of the new TI computers that won't run third party cartridges or modules, we have the solution. NEW!! The GROMBUSTER from Navarone Industries.

Unfortunately, some of the new TI 99/4A consoles will not play some of the new third party cartridges from Atarisoft, Romox and others. (You can tell if your computer is one of these because the title screen will say "1983 Texas Instruments V2.2.")

The GROMBUSTER plugs into the I/O port and overrides the internal operating system which allows you to use any cartridge manufactured for the TI.

CARTRIDGE PROGRAMMER

Put your own programs on cartridges.

Cartridge Programmer System	**$299.95**
Manual	**$25.00**
Blank Cartridges	**$19.95**

(allow 4-6 weeks for delivery)

CARTRIDGE EXPANDER

Extends the game port by providing 3 additional cartridge slots which will hold 3 of your favorite command modules. Reduces wear on Cartridge Connector, prevents overheating of cartridges & has built-in Reset button.
$39.95

JOYSTICK ADAPTER

Now you can use almost any joystick with your TI. Makes any Atari-style joystick compatible with the TI. Includes ports to plug in two joysticks. **ONLY $12.95**

PROSTICK II

A crisp, precise, professional video game-computer controller. Convenient "left/right" single function firing buttons. Unique switchable 4 way/8 way gateplate. Solid steel, short throw control shaft with arcade style ball. Durable leaf-switch mechanism. Easy-grip, hi-impact molded plastic housing. Extra long 6' cord with "strain relief" ends. Full five-year limited warranty. Complete with TI adapter. **ONLY $28.95**

3RD PARTY HARDWARE!!!

COMPLETE P.E. SYSTEM by CorComp

ITEM	PRICE
RS232 Card	**$ 89.95**
32K Memory	**$119.95**
Disk Controller	**$ CALL**
Mini P.E. Box	**$ CALL**

WE HAVE TI HARDWARE!!!

As of publication date Unisource has in stock: P.E. Boxes; Memory Expansion; Disk Controller Cards; Disk Drives; Color Monitors; Speech Synthesizers; Phone Modems; Program Recorders; Joysticks and much more. Prices and availability fluctuate. Please call for current status.

COMING SOON!! **The TI 99/4A UNISOURCE Encyclopedia/Catalog**

You will find this catalog represents the most complete collection of software, peripherals and accessories available today for your TI 99/4A Home Computer. It includes all previously manufactured (by TI) software and peripherals, and hundreds of new & exciting software packages from both TI and third-party developers. There's also a collection of peripherals, accessories and supplies for your 99/4A.

Our first three editions of the Encyclopedia/Catalog were extremely popular and we are now completing our 4th revision. It will be available to you for ONLY $3.00, refundable with your first order. And, you'll automatically get future updates at no extra cost. For our current customers who already own our catalog, a new edition will be sent to you free.

The best part of all is that your money goes farther when you buy from Unisource. Our new discount program will allow you to receive additional savings on 3rd party software and books ... 10% OFF on orders of $50 or more ... 20% OFF on orders of $100 or more. (not including shipping)

The Unisource TOLL-FREE ORDER HOTLINE makes your shopping simple and the Encyclopedia/Catalog gives you the most comprehensive shopping list for your 99/4A available today. Order your copy now!!

TOLL FREE ORDER HOTLINE
1-800-858-4580

IN TEXAS CALL 806-745-8834

Call today. Give us your name, address, Visa or MasterCard number, and we'll charge the $3.00 plus $1.50 shipping to your account. Or, send check or money order for $4.50 to us at the address below. Remember, the $3.00 is refundable with your first order!

P.S. When you're in Lubbock, come visit us at our retail store ... The UNISOURCE Home Computer Center in the University Square Shopping Center, University Avenue at South Loop 289.

UNISOURCE ELECTRONICS, INC.
P.O. Box 64240 • Lubbock, Texas 79464

Prices and availability may change at any time...CALL

Ordering Information

ORDER HOTLINE - TOLL FREE
1-800-858-4580

IN TEXAS CALL 1-806-745-8834

Just give us your name, shipping address and Visa or MasterCard number and we'll charge the order to your account. Please include $2.00 shipping and handling per order for software ($5 for peripherals).** Texas residents also add 5% sales tax. For mail-in orders, send to P.O. Box 64240, Lubbock, Texas 79464. Allow two weeks clearance for personal checks.

Normal orders of in-stock items will be shipped within a few days of order placement. Allow two weeks for delivery on any phone orders.
(prices and terms may change without prior notice)

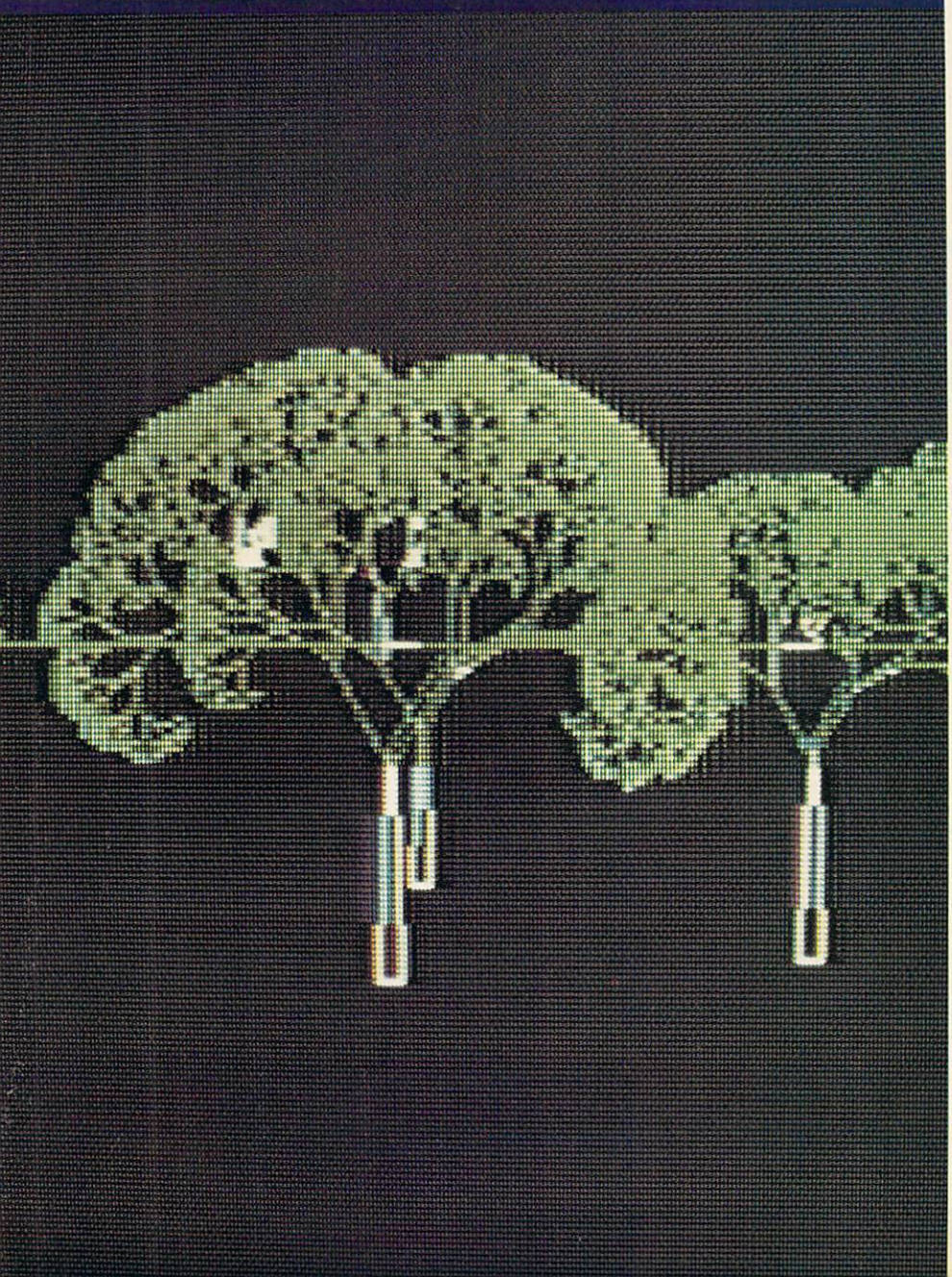
**Continental U.S. only. All others include actual airmail and insurance. A separate shipping charge of $10 for the PE Box will apply.

Toll Free Order Hotline
1-800-858-4580

in Texas call 806-745-8834

Circle 53 on Reader Service Card.

THE RESOURCE FOR LOGO TIMES THE LOGO LANGUAGE ™



The Binary Forest

By Robert S. Keller
HCM Staff

"This is Smokey the Bear talking. I guess the computer age has finally caught up with forestry—we use this LOGO program to work out our new tree designs. Quite a change from the old days, when every experiment took 40 years and a ton of wood.

"Of course, a computer tree will never replace a real tree—but for modeling purposes, the computer's a real time-saver. Later, we'll discuss how to try a few designs of your own.

"To grow five trees, simply type FOREST 5, *and press* [ENTER]. *The number that you grow is up to you—just pass the* parameter *this way to the procedure* FOREST, *and be sure to leave a space after* FOREST.

"The first trees will be bare except for a green bud or two. Edit the procedure BRANCHES *and change the 7 to a 3 in* IF :LENGTH < 7 [STOP] *to bring on spring. Change the 3 to a 1 when you're ready for summer.*

"Each tree uses random numbers to grow a unique shape. Each also has its roots in fractal mathematics, binary numbers, and recursion, as we shall see. I've become pretty familiar with these, even though I still feel more comfortable with a shovel."

The key-in LOGO listings for the Apple, IBM, and Commodore versions are found on page 49.
The tutorial for *The Binary Forest* begins on the following page.

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 Contents

August, 1984

The computer trees in the *Binary Forest* are an example of a fractal, a relatively new object in the field of mathematics. The landmark paper on fractals in the mid-70's posed this question: How long is the coastline of England? The paper demonstrated that if you measure the distance by the path a person walks, it is quite a bit greater than the figure commonly accepted by geographers. This is because there is quite a bit of detail—little bays and inlets and rocks and spits—ignored by the map-makers. As the beetle crawls, up and down every hillock, stone, and blade of grass, the distance is greater still. If you measure as the microbe slithers—or however they move—up and down every grain of sand, the distance is enormously greater. (In fact, the paper said the distance could be shown to be infinitely great.)

England is a *fractal*—when you look at it from an airplane, you see something with a jagged outline. When you get down on your knees and look at its shoreline, you see another version of the same jagged outline. Use a microscope, and you see the same jaggedness repeated on another level.

Understanding fractals allows mathematicians to make better models of them than have ever before been possible. Computer graphics programmers, taking the cue, have been building successful and realistic images—even whole mountain ranges—composed of smaller and smaller versions of the same basic shape.

The basic shape of the binary trees—the trunk and two branches—is repeated on a smaller and smaller scale until the smallest resolution of the LOGO graphics screen, 1 unit, is reached. (When the length of a branch becomes short enough, it turns green and becomes a leaf.)

Our natural tendency as human artists is to find some easy and abstract way to represent the small leaves and branches on a tree. But the computer makes the fractal approach possible, for it does the tedious work without complaint.

The basic tree structure works like this:

```
TO TRUNK :LENGTH
PD FD :LENGTH
MAKE "LENGTH :LENGTH*.75
BRANCHES :LENGTH
END
```

This procedure accepts the value you supply for :LENGTH and uses it to draw a trunk. Then it reduces :LENGTH and calls the procedure BRANCHES.

```
TO BRANCHES :LENGTH
RT 30
FD :LENGTH
BK :LENGTH
LT 60
FD :LENGTH
BK :LENGTH
END
```

Here, the turtle turns to the right 30 degrees and travels forward :LENGTH, then it backs up :LENGTH. It has just drawn a right-hand branch. Next, it turns left 60 degrees, travels forward :LENGTH, and backs up again. It has drawn a left-hand branch.

Fractals: The Root of the Binary Forest

We can control the further growth of the tree (as in Figure 1) through something called *recursion*. When a procedure is *recursive*, it means that it is mentioned within itself. Suppose we make this insertion (denoted by ★ ★) into the procedure BRANCHES:

```
TO BRANCHES :LENGTH
RT 30
FD :LENGTH
★ BRANCHES :LENGTH*.75 ★
BK :LENGTH
LT 60
FD :LENGTH
BK :LENGTH
END
```

Now, right in the middle of the BRANCHES procedure, BRANCHES is mentioned again. The second BRANCHES procedure, with a reduced value for :LENGTH, starts running, and the first BRANCHES procedure *still exists*. Even though :LENGTH has been reduced by .75 in the second BRANCHES procedure, :LENGTH still exists at its original size in the first BRANCHES procedure because each procedure has what is called a *local environment*. LOGO's ability to create local environments is necessary for this recursion to work properly.

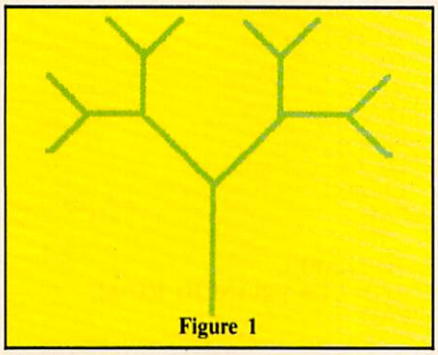


Figure 1

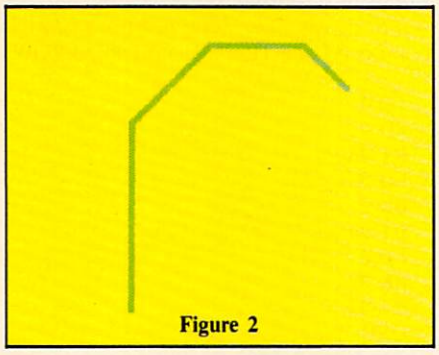


Figure 2

The first BRANCHES executed FD :LENGTH before calling the second BRANCHES, so the second BRANCHES begins drawing smaller branches where the first BRANCHES left off—at the end of a right-hand branch. We need a way to stop this recursion, for the second BRANCHES will now call a third, and the third a fourth, until like the sorcerer's apprentice, we face some unimaginable doom involving infinitely small bits of wood.To avert this fate, we insert this line:

```
TO BRANCHES :LENGTH
★ IF :LENGTH < 1 [STOP] ★
RT 30
FD :LENGTH
BRANCHES :LENGTH*.75
BK :LENGTH
LT 60
FD :LENGTH
BK :LENGTH
END
```

Now, BRANCHES calls BRANCHES until a BRANCHES has been called whose :LENGTH is less than 1. This invocation of BRANCHES stops itself right away. At the moment that the last invocation of branches STOPs, the drawing will look something like Figure 2. Here's the crucial point about recursion: all the previous invocations of BRANCHES still exist.When BRANCHES with a :LENGTH less than 1 ends, the BRANCHES that called it resumes execution. When that BRANCHES reaches its end, it ceases to exist and the BRANCHES that called it continues execution. Each BRANCHES finishes its work by drawing a left-hand branch.

To complete the tree, we need to get the left-hand as well as the right-hand branches to sprout. To accomplish this, we make this addition:

```
TO BRANCHES :LENGTH
IF :LENGTH < 1 [STOP]
RT 30
FD :LENGTH
BRANCHES :LENGTH*.75
BK :LENGTH
LT 60
FD :LENGTH
★ BRANCHES :LENGTH*.75 ★
BK :LENGTH END
```

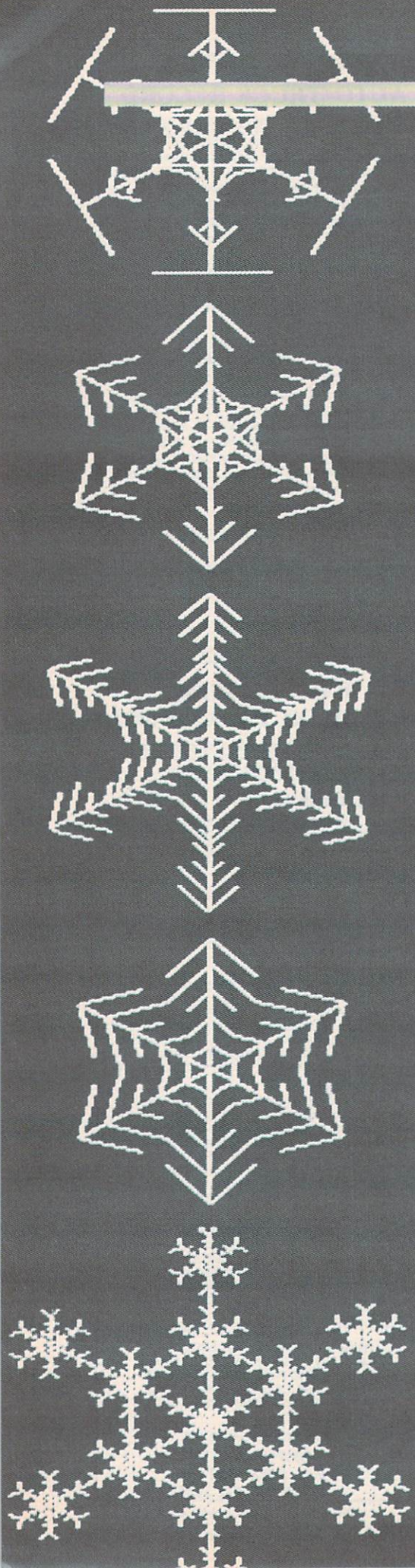
There's more to this program: Trunks and branches look better with some thickness to them. The procedures RIGHTSIDE and LEFTSIDE choose a thickness for each branch based on its length. TRUNK contains statements to build a thick trunk and to taper it from bottom to top.

FOREST draws the horizon, sets the graphics mode and color, and partly solves the problem inherent in growing trees on a flat screen—i.e., trees in the background need to be drawn first, so that they can be overlapped by the nearer trees. FOREST creates a variable called RANGE, and then calls DRAWTHEM. DRAWTHEM randomly selects tree locations one by one, using RANGE and LOW to make sure that planting starts in the background. After DRAWTHEM selects a place to plant, it calls TRUNK and the tree starts to grow.

Nature Random in Tooth and Claw

If you draw a computer tree using just the procedures we've described so far, it will be detailed, but too regular to be convincing. Many factors create a subtle variation among a real tree's branches. This program doesn't attempt to analyze this complexity—instead, random numbers are employed to *model* it. R becomes a

Continued on page 48



LOGO Flakes

by James Schwartz

No matter what they say about LOGO's advanced list processing capabilities and artificial intelligence roots, I love its simple, elegant graphics and its potential as a thinking stimulator. After diving head first into LOGO in July, 1981—and trying with varying degrees of success to learn and then teach its inner mysteries—I now believe that its greatest strength is its seemingly simple surface. LOGO truly is for learning.

As an educator, I have spent the past two years collecting and developing successful, educational uses for LOGO, looking for its best function. I've found it in the type of LOGO activity that develops a structure, lends itself to solving by pieces, and opens the door for creative exploration. The snowflake activity included here is an example of just such an ideal LOGO learning activity.

The Program

Remember your grade school science lessons? All snowflakes are variations on a six-sided geometric design. According to one of LOGO's learning foundations, we should begin by creating the simplest possible snowflake, and worry about how to make it more complex later. I imagine such a snowflake looks like Diagram 1.

The *Rule of 360* says that if the turtle is to make a complete spin, from the beginning of its drawing to the end, it must turn a total of 360 degrees. Therefore, each of six equal turns must be 60 degrees. In an effort toward elegance (and to simplify later developments) let's use REPEAT and the Rule of 360 to create this first design:

```
TO FLAKE0
REPEAT 6 [FD 21 BK 21 RT 60]
END
```

Setting Up The Pieces

The two most significant steps used to spice up this design are FD 21 and BK 21. These two steps form a single branch of the snowflake. Another LOGO learning foundation is to solve by pieces, so let's make a BRANCH procedure and an appropriate modification to FLAKE:

```
TO FLAKE0
REPEAT 6 [BRANCH RT 60]
END

TO BRANCH
FD 21 BK 21
END
```

We now have a very significant improvement! Any change of the BRANCH procedure will result in a new snowflake. For a simple modification, why not put some crosspieces (Diagram 2) on the branch?

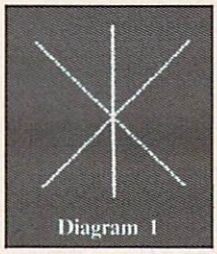


Diagram 1

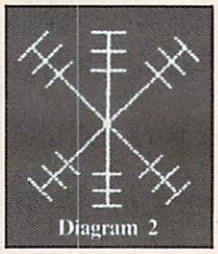


Diagram 2

Are you developing a habit of solving by pieces? If so, you have probably noticed that each branch is composed of three T-shaped pieces:

```
TO BRANCH1
REPEAT 3 [TEE]
END

TO TEE
FD 7
RT 90
FD 3
BK 6
FD 3
LT 90
END

TO FLAKE1
REPEAT 6 [BRANCH1 RT 60]
END
```

Become Creative

If that was easy, why not try a branch composed of a TEE, a Y, and an ARROW?

```
TO Y
FD 7
RT 45
FD 3
BK 3
LT 90
FD 3
BK 3
RT 45
END

TO ARROW
FD 7
RT 135
FD 3
BK 3
RT 90
FD 3
BK 3
RT 135
END
```

(Did you notice the Rule of 360 in one of these pieces?)

Continued on page 46

HERE'S AN OFFER YOU CAN'T REFUSE

Join The 99/4A Program Exchange Users Group for a one-time fee of only $10. Then pay only $2 each for programs. That's right. Only $2 each.

Membership provides. . .

. . . **LOW PRICES.** Programs for 99/4A users on cassette or diskette. $2 each.

. . . **QUALITY & DIVERSITY.** Over 1000 programs for 99/4A users. Games. Demonstrations. Education. Business. Home. Application. Utility.

. . . **NEWSLETTER** filled with hints, program reviews, problem solvers, & much more.

. . . **5 for 1 OPTION.** Submit a program to our software library & receive 5 FREE programs in exchange.

That doesn't mean you need programming experience to become a member.

. . . **1-TIME MEMBERSHIP FEE.** One guy we know about refused to join. He disappeared. Get the picture? We expect to hear from you. And soon. Besides, it's the best deal in town.

O.K. I GET THE PICTURE. HERE'S MY $10, THIS ENTITLES ME TO **PROGRAMS FOR $2 EACH** AND YOUR MONTHLY NEWSLETTER.

☐ Check or Money Order ☐ Visa/Mastercard

Card No. _____ Exp. _____

Name _____

Address _____

City _____

State _____ Zip _____

The 99/4A Program Exchange, THE 99/4A PROGRAM EXCHANGE INTERNATIONAL USERS GROUP P.O. Box 3242, Torrance, CA 90510

Circle 56 on Reader Service Card.

LOGO Flakes ... from p. 44

Take time to create a flake you like by arranging these three segments in different ways. You can, of course, use more than three segments to make a branch—just be sure to back up the turtle enough so it finishes the branch in the same place that it started it.

Here is an easy way to create a new flake without destroying a previous one: first, type TO FLAKE. Then move the cursor back and add a number to the word FLAKE (such as FLAKE1). Then move the cursor down one line, to the end of the word BRANCH. Add a number to the end of BRANCH (such as BRANCH1). Now press [FCTN] 9. Type TO BRANCH1, and define your new branch. This new flake will be called FLAKE1, and your original FLAKE will remain unchanged. This process defines all subsequent flakes.

Up to now, we have provided a flexible structure to work within, we have practiced solving by pieces, and we have opened the doors for creativity. (How many different branch segments and branches can you invent?) Now let's move on to the use of variables as a way of enhancing our creative explorations.

By giving our branch segments variable-length crosspieces, we will open up whole new worlds of snowflake design:

```
TO TEE :L
FD 7
RT 90
FD :L
BK :L * 2
FD :L
LT 90
END

TO Y :L
FD 7
RT 45
FD :L
BK :L
LT 90
FD :L
BK :L
RT 45
END

TO ARROW :L
FD 7
RT 135
FD :L
BK :L
RT 90
FD :L
BK :L
RT 135
END
```

Now, when we use these procedures in BRANCH, we will also need to specify numbers as inputs so that the variable :L will have a value.

```
TO BRANCH
ARROW 10
ARROW 20
ARROW 25
ARROW 20
ARROW 10
BK 35
END
```

At this point—with an infinite variety of shapes, and the ability to make those shapes in any sizes and combinations of sizes—we truly have some raw materials for creative exploration. Each new design—whether it is created through careful analysis leading to synthesis, or whether it results from combining elements randomly—will bring joy to the artist's heart.

Increasing Complexity

At some point in your creative exploration you may be troubled by the fact that larger flakes wrap around the screen, marring their beauty. The flakes could, of course, be made smaller by changing the length of each of the segments that compose the branches. This can lead to problems, though, because great changes in the length of these segments will result in disproportional flakes.

But, we can include a variable in the flake procedures to control the size of the flake, and then use this variable in each segment as a scale variable. Thus, the procedures given above for TEE, ARROW, and Y are rewritten as follows:

```
TO ARROW :X
FD :H
LT 135
FD :X
BK :X
LT 90
FD :X
BK :X
LT 135
END

TO TEE :X
FD :H
LT 90
FD :X
BK :X * 2
FD :X
RT 90
END

TO Y :X
FD :H
RT 45
FD :X
BK :X
LT 90
FD :X
BK :X
RT 45
END
```

The BRANCH procedures provided here offer some ideas for the scale variable :H. Each FLAKE procedure must be changed to allow input of a value for :H (for example, TO FLAKE :H). The SEE procedure is a top-level procedure which allows the user to choose a scale. It then displays a few of the snowflakes that are designed using these techniques.

Notice that this activity has brought you from a relatively simple procedure to some complex programming concepts and techniques. Each step from beginning to end involved a small, easily-learned concept. At every stage there was an element of beauty and an opportunity for creativity.

Although this activity was presented as a complete package, it is, in reality, open-ended. Refine it and experiment with it. The only logical END to this activity is the question, "I wonder if . . ." HCM

Complete key-in listing for *LOGO Flakes* is on the following page.

LOGO Flakes

```
TO FLAKE0 :H↵
REPEAT 6 [BRANCH RT 60
]↵
END↵
↵
TO SEE↵
CS PRINT [CHOOSE A SCAL
E: 2,5,OR 10 ]↵
CALL FIRST READLINE "S↵
TELL TURTLE↵
HT SC 15 CB 0↵
MAKE "FL "0123456 0 REPE
AT 7 [CS SY 20 MAKE "F
( SE ( WORD "FLAKE FIRS
T :FL ) [:S WAIT 100 ]
) RUN :F MAKE "FL BUTFI
RST :FL ]↵
END↵
↵
TO BRANCH↵
ARROW :H * 2↵
ARROW :H↵
ARROW :H / 2↵
ARROW :H↵
ARROW :H * 2↵
BK :H * 5↵
END↵
↵
TO FLAKE :H↵
REPEAT 6 [BRANCH RT 60
]↵
END↵
↵
TO FLAKE1 :H↵
REPEAT 6 [BRANCH1 RT 60
 ]↵
END↵
↵
TO FLAKE2 :H↵
REPEAT 6 [BRANCH2 RT 60
 ]↵
END↵
↵
TO BRANCH2↵
TEE :H * 2↵
ARROW :H↵
Y :H / 2↵
ARROW :H↵
TEE :H * 2↵
BK :H * 5↵
END↵
↵
TO FLAKE3 :H↵
REPEAT 6 [BRANCH3 RT 60
 ]↵
END↵
↵
TO FLAKE10 :H↵
REPEAT 6 [BRANCH10 RT 6
0 ]↵
END↵
↵
TO OUT↵
FD 30↵
RT 60↵
REPEAT 6 [RT 60 FD 30 ]
↵
LT 60↵
BK 30↵
END↵
↵
TO ARROW :X↵
FD :H↵
LT 135↵
FD :X↵
BK :X↵
LT 90↵
FD :X↵
BK :X↵
LT 135↵
END↵
↵
TO TEE :X↵
FD :H↵
LT 90↵
FD :X↵
BK :X * 2↵
FD :X↵
RT 90↵
END↵
↵
TO BRANCH3↵
Y :H * 2↵
TEE :H↵
ARROW :H / 2↵
TEE :H↵
Y :H * 2↵
BK :H * 5↵
END↵
↵
TO FLAKE6 :S↵
TYPE [PLEASE STAND BY..
. ]↵
CB 15↵
SY 24↵
REPEAT 6 [BRANCH6 RT 60
 ]↵
OUT↵
REPEAT 4 [PRINT [ ] ]↵
CB 0↵
WAIT 1000↵
CB 15↵
END↵
↵
TO BRANCH1↵
ARROW :H / 2↵
ARROW :H↵
ARROW :H↵
ARROW :H * 2↵
ARROW :H * 2↵
BK :H * 5↵
END↵
↵
TO FLAKE4 :H↵
REPEAT 6 [BRANCH1 FD 25
 BRANCH1 BK 25 RT 60 ]↵
END↵
↵
TO FLAKE5 :H↵
REPEAT 6 [BRANCH5 RT 60
 ]↵
END↵
↵
TO BRANCH5↵
ARROW :H↵
Y :H↵
BK :H * 2↵
END↵
↵
TO BRANCH6↵
FLAKE3 2↵
FD 30↵
FLAKE3 2↵
FD 30↵
FLAKE3 2↵
BK 60↵
END↵
↵
TO Y :X↵
FD :H↵
RT 45↵
FD :X↵
BK :X↵
LT 90↵
FD :X↵
BK :X↵
RT 45↵
END↵
↵
TO BRANCH10↵
Y :H↵
Y :H + 5↵
Y :H + 10↵
Y :H + 15↵
Y :H + 20↵
BK :H * 5↵
END↵
↵
```

HCM

Binary Forest . . . from p. 43

random number between 0 and 14 in BRANCHES. Then if R = 1, (as will happen 1 in 15 times) the length of the next branch is reduced by 1/4. If R = 2, (as will also happen 1 in 15 times) then the branch length will be reduced by 1/2. These numbers were selected by trial and error.

Other Trees

Changing the angle of the branches in the procedure BRANCHES can make a tree look radically different. If you change RT 30, you should probably change LT 60 so that it is still twice as great, or you may get some lop-sided trees; on the other hand, that may work if your trees are growing at the ocean. Another fruitful possibility is to make the angle of the branch vary randomly. MAKE "ANGLE 28 + RANDOM 5, and then RT :ANGLE, for instance, will make the angle vary randomly between 28 and 32. If you then put in LT :ANGLE * 2 instead of LT 60, the left branch of each particular BRANCH procedure will duplicate the angle of the right branch. If instead you type MAKE "ANGLE 28 + RANDOM 5, and LT :ANGLE, then the left branch will find its own random angle within the same bounds.

You may have noticed that some trees have one branch angle for large branches, and another (usually larger) angle for the smaller branches. Maybe this helps a tree grow above its surroundings and then spread its leaves. Try simulating this with these statements in BRANCH:

```
IF :LENGTH > 9 [MAKE :ANGLE 28]
IF :LENGTH < 9 [MAKE :ANGLE 60]
```

then:

```
RT :ANGLE
```

and

```
LT :ANGLE * 2
```

Try different angles until you find a combination that looks right to you.

The branches on young trees generally seem to diminish in size more quickly than the branches on a mature tree. Perhaps each time DRAWTHEM invokes TRUNK to draw a new tree, you could randomly select this rate so that your forest is a mix of young and old trees. Instead of having 0.75 *hard-coded* into the program, you will need to turn it into a variable.

By combining these two factors, angle and branch length, you can simulate some unusual trees—such as the Poplar, whose long branches grow almost directly upward.

LOGO's procedural ability means that you can easily import the *Binary Forest* into any of your other LOGO programs. Call the program up and it will join whatever else is in your workspace at the time. If you import the forest into a game or another program, you may find the time required for tree drawing too great to be useful. (It can often take 20 or 30 minutes to draw a five-tree, fully-leafed forest, depending on tree size and machine.) If you want to cut this time down, you need to explore simpler ways to draw the leaves. Various solutions to this problem include increasing the size of the smallest :LENGTH drawn, or going to another mode of drawing when :LENGTH is reduced into the realm of the leaves.

jr PC

Before entering LOGO, be sure you are in color mode. (Use "MODE CO40" from DOS.) You also need a color graphics card in your PC for growing binary trees.

IBM LOGO uses the command SETPC instead of the more common PC to set the pen color. It uses SETPAL to set a color "palette," and it requires that the conditional parts of an IF statement be placed in brackets (as in IF :LENGTH > 3 [SETPC 3] in the procedure BRANCHES). SETPRECISION 5 tells LOGO to carry out the results of its arithmetic operations to only five digits instead of the normal ten.

C64

In order to have the proper green trees of summer, we had to change the graphics mode of Commodore LOGO from the normal default mode (SINGLECOLOR) to DOUBLECOLOR. SINGLECOLOR mode allows for one color other than the background color in an eight-by-eight pixel area. When drawing green leaves with white branches on a black background in this mode, the colors of the leaves and branches often vacilate between the two colors, with the trees ending up with unsightly white blights where there should be green, leafy areas.

The DOUBLECOLOR mode allows two colors apart from the background, which allows for trees that look quite satisfactory. In DOUBLECOLOR mode the branches tend to be a little thicker—but the coloration is as it should be. The change is made when the graphics are initialized in the TO FOREST procedure by simply entering DOUBLECOLOR.

The only other difference in the C-64 version stems from the change of graphics mode. Because the resolution is not quite as fine, we've made the trees more likely to be created smaller and in the background, by decreasing the number used in the MAKE :RANGE statement in the TO FOREST procedure from 35 to 30. Also, the pen color designations are slightly different because green is set with a PC 5.

Apple

The Apple version of *Binary Forest* is identical to the IBM PC version, except for two small changes: The resolution on the Apple screen is slightly lower (280 horizontal pixels by 192 vertical pixels as opposed to 320 by 250), so in the MAKE "RANGE statement we use 15 instead of 35 to establish the trees' sizes and placement on the screen. This difference in resolution also necessitated our altering the MAKE WIDTH statement and the distance the turtle goes forward (FD) in the TO DRAWTHEM procedure. Aside from these minor changes, and the substitution of PC 1 for 3 and PC 2 for 1, the two versions are identical.

Binary Forest

TERRAPIN LOGO — APPLE II Series

```
TO TRUNK :LENGTH
  PC 1 PU BK :LENGTH
  RT 90 FD ( :LENGTH * .15 ) LT 92
  PD BK ( :LENGTH * 2 ) RT 92
  BK ( :LENGTH * .3 )
  LT 88 FD ( :LENGTH * 2 ) PU
  RT 88 FD ( :LENGTH * .1 ) LT 92 PD
  BRANCHES :LENGTH
END

TO BRANCHES :LENGTH
  IF :LENGTH < 7 THEN STOP
  MAKE "R RANDOM 15
  IF :R = 1 THEN MAKE "LENGTH :LENGTH * .75
  IF :R = 2 THEN MAKE "LENGTH :LENGTH * .5
  RT 30
  IF :LENGTH > 4 THEN PC 1 ELSE PC 2
  RIGHTSIDE :LENGTH
  FD :LENGTH
  LEFTSIDE :LENGTH
  BRANCHES :LENGTH * .75
  IF :LENGTH > 4 THEN PC 1 ELSE PC 2
  LEFTSIDE :LENGTH
  BK :LENGTH * .8 PU BK :LENGTH * .2 PD
  RIGHTSIDE :LENGTH
  LT 60
  RIGHTSIDE :LENGTH
  PU FD :LENGTH * .2 PD FD :LENGTH * .8
  LEFTSIDE :LENGTH
  BRANCHES :LENGTH * .75
  IF :LENGTH > 4 THEN PC 1 ELSE PC 2
  LEFTSIDE :LENGTH
  BK :LENGTH
  RIGHTSIDE :LENGTH
  RT 30
END

TO RIGHTSIDE :LENGTH
  PU RT 90 FD :LENGTH * .1
  LT 90 PD
END

TO LEFTSIDE :LENGTH
  PU LT 90 FD :LENGTH * .1
  RT 90 PD
END

TO DRAWTHEM :TNUMBER
  MAKE "WIDTH 5 + RANDOM 271
  MAKE "DEPTH :LOW + RANDOM ( :LOW + :RANGE )
  MAKE "LOW ( :LOW + :RANGE )
  PU HOME
  RT 90 FD 140
  BK :WIDTH LT 90
  TRUNK :DEPTH
  IF :TNUMBER = 1 THEN STOP
  MAKE "TNUMBER :TNUMBER - 1
  DRAWTHEM :TNUMBER
END

TO FOREST :TNUMBER
  RANDOMIZE
  MAKE "RANGE 15 / :TNUMBER
  MAKE "LOW 5
  CS FULLSCREEN HT
  PU HOME PC 1 PD RT 90 FD 280 PU
  DRAWTHEM :TNUMBER
END
```

COMMODORE LOGO — C-64

```
TO BRANCHES :LENGTH
  IF :LENGTH < 7 STOP
  MAKE "R RANDOM 15
  IF :R = 1 THEN MAKE "LENGTH :LENGTH * 0.75
  IF :R = 2 THEN MAKE "LENGTH :LENGTH * 0.5
  RT 30
  IF :LENGTH > 4 THEN PC 1 ELSE PC 5
  RIGHTSIDE :LENGTH
  FD :LENGTH
  LEFTSIDE :LENGTH
  BRANCHES :LENGTH * 0.75
  IF :LENGTH > 4 THEN PC 1 ELSE PC 5
  LEFTSIDE :LENGTH
  BK :LENGTH * 0.8 PU BK :LENGTH * 0.2 PD
  RIGHTSIDE :LENGTH
  LT 60
  RIGHTSIDE :LENGTH
  PU FD :LENGTH * 0.2 PD FD :LENGTH * 0.8
  LEFTSIDE :LENGTH
  BRANCHES :LENGTH * 0.75
  IF :LENGTH > 4 THEN PC 1 ELSE PC 5
  LEFTSIDE :LENGTH
  BK :LENGTH
  RIGHTSIDE :LENGTH
  RT 30
END

TO RIGHTSIDE :LENGTH
  IF :LENGTH < 12 THEN STOP
  PU RT 90 FD :LENGTH * 0.075
  LT 90 PD
END

TO LEFTSIDE :LENGTH
  IF :LENGTH < 12 THEN STOP
  PU LT 90 FD :LENGTH * 0.075
  RT 90 PD
END

TO TRUNK :LENGTH
  PU BK :LENGTH
  RT 90 FD ( :LENGTH * 0.15 ) LT 92
  PD BK ( :LENGTH * 2 ) RT 92
  BK ( :LENGTH * 0.3 )
  LT 88 FD ( :LENGTH * 2 ) PU
  RT 88 FD ( :LENGTH * 0.1 ) LT 92 PD
  BRANCHES :LENGTH
END

TO DRAWTHEM :TNUMBER
  PC 1
  MAKE "DEPTH :LOW + RANDOM :RANGE
  MAKE "WIDTH 5 + RANDOM 311
  MAKE "LOW ( :LOW + :RANGE )
  PU HOME
  RT 90 FD 160
  BK :WIDTH LT 90
  TRUNK :DEPTH
  IF :TNUMBER = 1 THEN STOP
  MAKE "TNUMBER :TNUMBER - 1
  DRAWTHEM :TNUMBER
END

TO FOREST :TNUMBER
  MAKE "LOW 4
  MAKE "RANGE 30 / :TNUMBER
  DOUBLECOLOR FULLSCREEN HT CS PU
  HOME PC 1 PD RT 90 FD 160 HOME PU
  DRAWTHEM :TNUMBER
END
```

LOGO SYSTEMS LOGO — IBM PC & PCjr

```
TO TRUNK :LENGTH
PU BK :LENGTH
RT 90 FD (:LENGTH * 0.15) LT 92
PD BK (:LENGTH * 2) RT 92
BK (:LENGTH * 0.3)
LT 88 FD (:LENGTH * 2)
PU
RT 88 FD (:LENGTH * 0.1) LT 92 PD
BRANCHES :LENGTH
END
TO BRANCHES :LENGTH
IF :LENGTH < 7 [STOP]
MAKE "R RANDOM 15
IF :R = 1 [MAKE "LENGTH :LENGTH * 0.75]
IF :R = 2 [MAKE "LENGTH :LENGTH * 0.5]
RT 30
IF :LENGTH > 4 [SETPC 3] [SETPC 1]
RIGHTSIDE :LENGTH
FD :LENGTH
LEFTSIDE :LENGTH
BRANCHES :LENGTH * 0.75
IF :LENGTH > 4 [SETPC 3] [SETPC 1]
LEFTSIDE :LENGTH
BK :LENGTH * 0.8 PU BK :LENGTH * 0.2 PD
RIGHTSIDE :LENGTH
LT 60
RIGHTSIDE :LENGTH
PU FD :LENGTH * 0.2 PD
FD :LENGTH * 0.8
LEFTSIDE :LENGTH
BRANCHES :LENGTH * 0.75
IF :LENGTH > 4 [SETPC 3] [SETPC 1]
LEFTSIDE :LENGTH
BK :LENGTH
RIGHTSIDE :LENGTH
RT 30
END
TO RIGHTSIDE :LENGTH
IF :LENGTH < 12 [STOP]
PU RT 90 FD :LENGTH * 0.075
LT 90 PD
END
TO LEFTSIDE :LENGTH
IF :LENGTH < 12 [STOP]
PU LT 90 FD :LENGTH * 0.075
RT 90 PD
END
TO DRAWTHEM :TNUMBER
SETPC 3
MAKE "DEPTH :LOW + RANDOM :RANGE
MAKE "WIDTH 5 + RANDOM 311
MAKE "LOW (:LOW + :RANGE)
PU HOME
RT 90 FD 160
BK :WIDTH LT 90
TRUNK :DEPTH
IF :TNUMBER = 1 [STOP]
MAKE "TNUMBER :TNUMBER - 1
DRAWTHEM :TNUMBER
END
TO FOREST :TNUMBER
SETPRECISION 5
MAKE "LOW 4
MAKE "RANGE (35 / :TNUMBER)
FULLSCREEN HT CS PU
HOME SETPAL 0 SETPC 3 SETBG 8
PD RT 90 FD 400 HOME PU
DRAWTHEM :TNUMBER
END
```

HCM

THE BEST OF 99'er™

Pick up a copy at your favorite bookstore or dealer. To order direct, use form on page 157.

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

VOLUME 1

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

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

THE BEST OF 99'er ON TAPE

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

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

To order see page 157

People Who Know the Home Computer Best™

HOME COMPUTER™ magazine

Emerald Valley Publishing Co.
P.O. Box 5537, Eugene, OR 97405

AN EMERALD VALLEY BOOK
EVP

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 Dr., Suite 250
Eugene, OR 97401

RF INTERFERENCE REMOVED

The engineers at Bytes & Pieces have developed a kit that eliminates RF interference problems common in older models of the Commodore 64. The Interference Filter Kit requires opening the computer and making three solder connections. Instructions are included. The kit retails for $15.95.

Bytes & Pieces
550 North 68th St.
Wauwatosa, WI 53213

HAND-HELD-SIZE PRINTER

Computer Peripherals, Inc. has announced its Printelex printer model TH-40-160, a hand-held-size printer weighing only 1.5 pounds. It has Centronic parallel and RS232 serial interfaces; auto wrap-around for texts over 40 columns wide; and software-controlled form feed, vertical tabs, right justification, page size and print speed. Custom cable configuration is available for most computers. The Printelex retails in the $130-$150 range.

Computer Peripherals, Inc.
6400 Canoga Ave. Suite 305
Woodland Hills, CA 91367
(818) 710-1717

ANIMATION AID FOR BEGINNERS

Milton Bradley Company's Education Division has announced a computer literacy aid product titled Programming Animation and Graphics for the TI 99/4A, and the Apple II/IIe computers. Its set of 60 task cards is designed to introduce animation and graphics to beginning programmers. Developed for teacher use, the cards include definitions and programs for user interaction with the computer. It is for ages 10 and up, and costs $9.95.

Milton Bradley Company
Springfield, MA 01102-2888
(413) 525-6411

WHAT'S NEW FROM BIG BLUE?

The IBM PC, PCjr, and PC/XT will soon be able to be used as videotex terminals to send or receive data. Videotex images may be stored on diskettes for later viewing. PC/Videotex will generally be available in October 1984, with a one-time charge for the program diskette—for PC or XT, $250; for PCjr, $220. The new IBM PCjr Color Display is a direct-drive display for personal productivity and graphics applications on the IBM PCjr. It includes sound capabilities and carries a suggested retail price of $429. Three new text-processing programs have also been introduced by IBM. *DisplayWrite 1*, a $95, menu-driven, word processing program designed for the PCjr, can run on the other PCs as well. It uses ASCII data representation, allowing data from printable DOS ASCII files to be incorporated into a *DisplayWrite 1* text document. *DisplayWrite 2*, a $299 program similar to the Displaywriter Textpack 4, is designed for light to medium typing in an office environment. It runs on the PC, XT, Portable, and the 3270 PC, and features a spelling verification aid of about 100,000 words. *DisplayComm* enables PCs, XTs, and Portables to send documents written in *DisplayWrite 2* and information from other PC files to similarly equipped PCs, *DisplayWriter*s, and other IBM office system products. Using the $375 program, the documents can be transmitted at high speeds in binary synchronous communication form.

IBM Corp., Information Systems Group
900 King St.
Rye Brook, NY 10573
(914) 934-4488

PRACTICALC II FOR APPLES

Computer Software Associates, Inc. has introduced *PractiCalc II*(tm) for the 48K Apple II series computers. CSA's spreadsheet program was originally released last year for Commodore computers, and is priced at $69.95. It can be used to balance a checkbook, keep lists and files, prepare financial statements, and produce business forecasts. CSA has also announced *PractiFile*(tm), a data base program for the C-64. The program is interactive with CSA's *PractiCalc 64*(tm) and *PS*(tm) spreadsheets, and supports several word processing programs for file merging. It is designed to help families, businesses, and educators handle tasks involving numbers or lists. *PractiFile*(tm) retails for $55.

Computer Software Associates
44 Oak St.
Newton Upper Falls, MA 02164
(617) 527-7510

HOME COMPUTER product news

STICKYBEAR LEARNINGWARE

Four new software learning and game programs have been released by the Computer Software Division of Xerox Education Publications. The programs are initially for use on Apple II computers, and are each priced at $39.95. *Stickybear BOP* is a multilevel computer game for the family, and includes a game disk, a full-size poster, a stand-up game, and Stickybear stickers. *Stickybear ABC* is an alphabet program for three- to six-year-olds featuring animated pictures with sound. The pictures display objects that move over color backgrounds, without the need for special hardware. *Stickybear Numbers* allows children ages three to six to develop number recognition and build counting skills through play. *Old Ironsides* is a two-player game of naval strategy with three-masted ships which recreate the age of sailing.

Xerox Education Publications
245 Long Hill Road
Middletown, CT 06457
(203) 347-7251

K-TEL RELEASES 160 TITLES

K-TEL Software, Inc., a wholly-owned subsidiary of K-TEL International, Inc., has released for shipment 160 titles of budget priced software for Apple, Commodore 64, VIC-20, and Atari computers. The software includes game, educational, and business titles, and will be marketed under the "K-TEL" brand.

K-TEL Software, Inc.
11311 K-TEL Drive
Minnetonka, MN 55343
(612) 932-4000

ANALOG & VIDEODISK I/O

Micromagic has introduced two new hardware products for the TI-99/4A. The ADM-100 Analog to Digital Interface plugs directly into the 99/4A, does not require additional peripherals, cartridges, or an expansion system, and can be used with TI BASIC. A built-in three-step attenuator provides for three input ranges of 0-5, 0-10, and 0-50 volts dc. Software is supplied to make the computer function as a digital voltmeter. Its total cost is $79.95. The LVM-99 Videodisc Interface/Controller also plugs directly into the 99/4A and requires the TI Mini Memory to operate. The unit interfaces the computer to the Magnavox 8010, Sylvania 7200, or Pioneer VP-1000 videodisc players. Its cost of $249 includes demonstration software and all cables and connectors.

Micromagic Software
4129 Abercorn Road
Knoxville, TN 37921

MAKE IT *Ez* ON YOURSELF

Softron, Inc. has announced two aids for disk drive users, and a keyboard overlay *Ez Disk*(tm) for the Commodore 64 and VIC-20 allows users to call up a menu to Load, Save, and Run BASIC and machine language programs. It will also Show Directory, Read Errors, Rename, Erase, and Compress files. The program supports up to four disk drives, and is available for $24.95. *Ez/Disk Plus*(tm) for the Commodore 64 performs all of the above functions, plus it prints files directly from disk to screen or printer, appends programs, and more. *Ez/Disk Plus* sells for $34.95. Keyboard Soft/Lay(tm), is a keyboard overlay for the Commodore 64 and VIC-20. BASIC commands, memory locations, maps, sprites, and sound information are provided for easy reference. It is available for $8.95.

Softron, Inc.
150 Nassau St., Suite 2024
New York, NY 10038
(212) 608-2922

EXTENDING EXTENDED BASIC

J & KH Software has released its *Super Extended Basic* (SXB) for the TI-99/4A. The software adds a third layer of subroutines to TI BASIC and Extended BASIC. Most of its more than 100 subroutines fall into five categories: data base, string array, string, integer, and video display processor. *SXB* is priced at $99.95, and requires the TI Extended BASIC command module and 32K memory expansion.

J & KH Software
2820 S. Abingdon St.
Arlington, VA 22206
(703) 820-4131

PLUG COMPATIBLE PRINTER

Axiom Corporation has announced a printer that is plug-compatible with the Texas Instruments 99/4A computer. The GP 100-TI does not require the TI interface box, because the printer is shipped with a cable that plugs directly into the side of the computer. It is compatible with TI's sidecar modules and expansion interface. Users can set the number of line feeds per line, line length, margins, and get dot-addressable graphics. The GP100-TI retails for $299.

Axiom Corporation
1014 Griswold Ave.
San Fernando, CA 91340
(818) 365-9521

SPEECH SYSTEMS GUIDEBOOK

WGBooks has announced the publication of *Speech Systems for Your Microcomputer*, a guidebook for microcomputer users considering voice technology. Author Gary A. Shade discusses applications of voice input and output in education, in the home, and for the handicapped. The book also examines existing speech systems on the market, and provides guidelines to help readers determine which system is best for their own home use. *Speech Systems for Your Microcomputer*, priced at $14.95, includes program listings, photographs, and a glossary of speech technology terms.

The Wayne Green Publications Group
Peterborough, NH 03458
(603) 924-9471

A NEW KEYBOARD FOR JR

Key Tronic Corporation has introduced a plug-compatible keyboard for the IBM PCjr. The KB5151jr is engineered similarly to Key Tronic keyboards for the IBM PC. Priced at $255, the new keyboard features LED indicators on lock keys, solid-state capacitive switches, and separate numeric, cursor, and function key areas

Key Tronic Corporation, Department E6
P.O. Box 14687
Spokane, WA 99214
1-800-262-6006

PLUG IN TO YOUR COMPUTER

Thought Technology has created a monitor and software program that together allow users to plug themselves into their home computers and mentally and physically interact with the system. The Calmpute 1(tm) monitors a user's tension level through galvanic skin resistance, which measures the opening and closing of the skin's pores. The monitor produces an individual stress profile, and automatically adjusts for individual differences in stress responses. The GSR monitor has inputs to monitor heart rate, temperature, and muscle activity. The *Calmpute*(tm) program demonstrates how both physical and mental stress factors affect people, and helps teach stress control. It includes several biofeedback games that stimulate users, who must remain calm under the games' pressures to win. The monitor and the software together retail for $79.95.

Thought Technology Ltd.
2180 Belgrave Ave.
Montreal, P.Q. Canada H4A 2L8

PHYSICAL SOFTWARE

Two new software packages–one for children, and one for the athletic population, are among the programs and software applications available from a new catalog by Digital Cybernetics International. Diet and exercise play an important role in *Physicalc*, which has capabilities to monitor and graph body weight and fat percentages over extended time periods. *Capitals* teaches users the names and correct spellings of the 50 U.S. state capitals. Digital Cybernetics also offers a number of educational- and engineering-oriented software. The catalog can be purchased by sending $1 to DCI.

Digital Cybernetics International
P.O. Box 264
East Amherst, N.Y. 14051-0264
(716) 688-1250 ext.1

COMPACT EXPANSION FOR 99/4A

Tex Micro Inc. has released TIMPAC, a "TI eMulating Peripheral Access Controller." At 1-1/2" high, TIMPAC includes everything necessary to add 32K RAM, printers, and floppy disk drives to the TI-99/4A. It will also accommodate a speech synthesizer card, a hard disk controller interface, and an access cable to the console expansion port. For $499, the purchaser receives TIMPAC, a Disk Access Cartridge, a Texdisk diskette, and a manual.

Tex Micro Inc.
Titusville, FL 32783-5366
(305) 267-4513

TI QUICK REFERENCE GUIDE

The *Q*Card*(tm) Reference Card is now available for users of TI BASIC and Extended BASIC. The brochure-size, fan-folding reference guide was compiled using excerpts from *The Texas Instruments User's Encyclopedia* by Gary Phillips and David Reese. Arranged in alphabetical order, it contains definitions and symbols for the commands and functions used in TI BASIC and Extended BASIC. The *Q*Card* costs $2.95.

Texas Instruments User's Encyclopedia
1001 Bridgeway, Suite 205
Sausalito, CA 94965
(415) 331-2395

COLOR DISPLAY MONITOR

Sakata U.S.A. Corporation has introduced its Sakata Model SC-100 CRT Composite Color Display Monitor. It is compatible with Apple II and IIe, Commodore 64, VIC-20 and other personal computers. The Model SC-100 13" CRT has 90 degree inline with 0.65mm dot pitch. It retails for $329.

Sakata U.S.A. Corporation
651 Bonnie Lane
Elk Grove Village, IL 60007
(312) 593-3211

BASIC TIPS FOR PROGRAMMERS

AMLIST, Inc. has announced the publication of *Basic TIPS: Comprehensive Program Instructions for the TI-99/4A* by Terrance K. Castle. The book stresses the effective use of commands and functions, instead of their definitions. Chapters cover programming structure, debugging, data files, graphics, sorting arrays, algorithms, and memory conservation. The manual includes 16 programs used as part of the book's teaching technique. *Basic TIPS* can be purchased for $15.95, which includes a cassette of all programs.

Distributed by TexComp
P.O. Box 33084
Grenada Hills, CA 91344
(818) 366-6631

A *NEWWORD* PROCESSOR

Newstar Software, Inc. has announced that *Newword*(tm), a word processing/merge print package, is now available for the IBM PC, PCjr, and compatibles. It is upward compatible with *Wordstar*(tm)/*Mailmerge*(tm) programs. The package features merge print, support of seven screen attributes, dot-matrix printer support, function key support with all 40 keys programmed by menu selection, and an "undelete" key. *Newword* retails for $249.

Newstar Software Inc.
1601 Oak Park Blvd.
Pleasant Hill, CA 94523
(415) 932-2278

DISK ENVELOPE DESIGN

Triot Software has released its *Catalog-Comments Program Disk Envelope Designer*. This program allows users with an expanded TI-99/4A system to design individual disk envelopes with a catalog listing printed on the outside. Comments on each listing in the disk's catalog may also be stored on a separate comment file disk if the enveloped disk contains less than 54 files. Disk catalogs with more than 54 files can be printed on normal or legal size paper. The comment file disk enables users to update the catalog and alter comments at any time. The *Disk Envelope Designer* requires a 99/4A computer with 32K Memory Expansion, TI Extended BASIC, one or two disk drives, and a printer. It is available by ordering Version 1 of *Disk Envelope Designer* from Triot Software for $20, prepaid.

Triot Software
P.O. Box 115
Liscomb, IA 50148
(515) 496-5455

COPY COVER (PAT PEND)

KEEPING COMPUTERS FUZZFREE

C-Thru Products has introduced the Copy Cover, a combination dustcover and copyholder made of lucite. It is custom fit for most computers, including IBM PC, PCjr, XT, Apple, TI-99/4A, Commodore 64, VIC-20, Radio Shack, and Kaypro computers. The Copy Cover protects keyboards from dust and dirt, is static free, and becomes a copyholder when flipped up above the keyboard. It is available for $39.95.

C-Thru Products
6351 Lake Worth Road, Suite 111
Lake Worth, FL 33463

RELIEF FOR COMPUTER FATIGUE

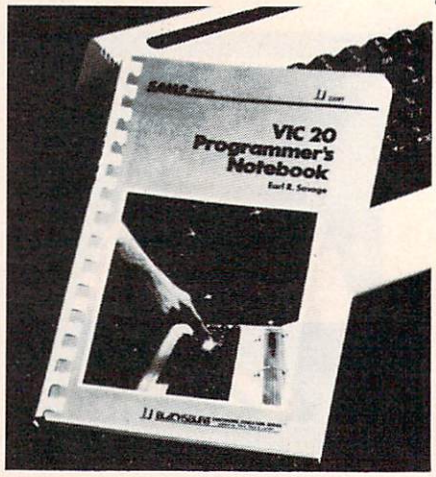
Discwasher has introduced a product designed to ease the arm, wrist, and shoulder fatigue that results from extended periods of work with a computer. Entry Rest, made of solid oak, is mountable on Apple II series computers. It has a built-in multi-function calculator with memory, and a static grounding bar that protects the computer from static charges. *Entry Rest* can be purchased for $34.95.

Discwasher
1407 North Providence Road
Columbia, MO 65205

TIPS FOR VIC-20 USERS

Howard W. Sams & Co., Inc. has released the *VIC-20 Programmer's Notebook* by Earl R. Savage. The book provides subroutines and programming ideas for both novice and experienced programmers, and includes information on adding memory, dealing with data, troubleshooting, and accessories. Program statements are written in VIC-20 language, and flowcharts illustrate program operations. The *VIC-20 Programmer's Notebook* is $14.95.

Howard W. Sams & Co., Inc.
4300 West 62nd St.
Indianapolis, IN 46268
(317) 298-5400



CUSTOM PRINTING PACKAGE

A software package that enables its users to write, design, and print greeting cards, stationery, letterhead, signs, and banners has been introduced by Broderbund Software. *The Print Shop* produces messages that can be written in one of eight different typestyles, in two sizes, and in solid, outline, or three-dimensional formats. It has nine border designs, ten abstract patterns, and a built-in graphics editor to create original pictures and symbols. *The Print Shop* requires an Apple II or IIe computer and a printer. It is priced at $49.95, and comes with a colored assortment of pin-feed paper, matching envelopes, and a reference manual.

Broderbund Software
17 Paul Drive
San Rafael, CA 94903
(415) 479-1170

TRADING ON THE RUN

Redwick & Company has announced two software packages for the TI-99/4A, Apple IIe, and Commodore 64. *Cargo Run* is a non-graphic trade simulation game. The user is a galactic trader, buying and selling goods at various star systems, each with a distinct economic base. The trader must anticipate fuel use, taxes, market need, and availability while staying ahead of creditors. *Cargo Run*'s price is $29.95. The *Redwick Inventory/Invoice System* is a direct inventory update, invoice writing package that supports any number of inventory items, accessed by name. The system reports on stock levels, stock profits, taxes, shipping costs, daily sales, and other items. The *Inventory/Invoice System* is priced at $99.95.

Redwick Company
P.O. Box 45041
Winter Hill, MA 02145

CUSTOM FILING FOR THE HOME

Continental Software has introduced *The Home Cataloger*, a software program for the IBM PCjr, Apple II/IIe, and Commodore 64. This filing and cataloging program can create up to 1,500 individual entries, depending on the hardware used with it. The user can create customized filing systems, or select one of 10 predesigned cataloging lists, which include telephone, travel planner, insurance policies, studies, or books. *The Home Cataloger* can total numbers in any or all numeric categories, and can generate whole lists or lists of selected categories in any order. It is available for the suggested retail price of $49.95.

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

MATH DRILLS FOR KIDS

Stilwell Software Products has developed an educational program designed to help tutor and drill grade school and junior high school age children. *Math School* has 36 different types of arithmetic problems for drills, and three levels of difficulty. Each session has different math problems, since the numbers are randomly selected. Users can time themselves on problem sets, and can have their test results analyzed to find areas they are not mastering. The tutor section also has three levels of difficulty, and provides immediate feedback and time to study problem answers. *Math School* is priced at $15, and will run on the IBM PC, PCjr, XT, and Portable.

Stilwell Software Products
16403 North 43rd Drive
Glendale, AZ 85306

The Home Accountant

A review by Tom Green
HCM Staff

"When E.F. Hutton talks, people listen"—an adage that may have helped to fatten a few wallets, but how many of us can afford to take advantage of Hutton's—and other brokerage firms of similar ilk—golden advice? Do you often wonder why your "mad money" disappears long before the next paycheck? How many times have you initiated a budget or put some money in savings, only to have an overlooked or unexpected bill put you back in the red? If these scenarios are all too familiar, here is a software package that can perhaps make them distant memories.

Continental Software's *Home Accountant* presently tops many of the software best-seller lists under the *Home Management/Productivity* category. There is a good reason for this: the product sets the standard for home financial management programs, and appeals to both neophytes and seasoned users seeking to refine their personal financial affairs. It does this by introducing users to a comprehensive, yet easy-to-follow personal budgeting program that is versatile enough for a wide diversity in income levels. With a suitably equipped microcomputer and *Home Accountant*, you will be able to track income and expenses over a period of time, make budget forecasts for comparison against your actual spending and saving patterns, and set realistic financial projections or attainable monetary goals.

System Overview

Most complex tasks can be simplified by approaching the problem in a sequence of small steps. The *Home Accountant* is a case in point: it is structured in modules accessable from a central menu. A diagram of this "menu-driven" program's different modules and options is exhibited in Figure 1. A diagram of the *Home Accountant Plus* options for the IBM PC, shown in Figure 2, is similar to Figure 1 except that it also indicates additional options found only in the IBM PC version of the program.

To create a working model of the budget format, you need only to follow these three steps: (1) type in the name of your computer system and its periperals, (2) format your budget categories, and (3) enter your data into the budget categories.

The main menu's options are listed according to how often a module is accessed. Module 8, Hardware/Start New System, is a preliminary step used to identify each system's set-up. The information is used to properly interface the disk drive(s) and printer to the program. Thereafter, this module is accessed only to enter changes in equipment.

The Budget module contains an option for starting a new budget called New System. Its purpose is to initialize a data disk for interaction with the main program disk. Upon completion of this part of the formatting process, the Budget menu will appear, and your options will be limited to this menu until the data disk format is completed. With a model of the budget format ready, the next operation is to enter the data.

The Stage Is Set

Records of cash flow and expenses involving budget items are entered in the Transactions module. Refer to Figure 1 for a list of the options in this module. The first three options, Checkbook, Credit Card, and Cash access several different categories for data manipulation. Option 4, Start New Month, is for entering the first month's data, and is used to increment the months as you go along. Option 5, Select Person/Checkbook, permits access from one budget category to another. Option 6, Select Different Data Disk, allows you to make a data disk interchange when a disk is full, or at the end of a fiscal year. Option 7, Main Menu, returns you to the main menu.

To enter a single day's financial transactions—say, three expenditures (gas by credit card; lunch with cash; and groceries by check) on May 5, 1984—the process would go like this: Load *Home Accountant*, choose option 1 from the main menu to enter the Transaction module, and select the category number that contains the file for the food budget. When the Transactions menu is displayed, choose sub-option 1, Enter Checks/Deposit. Now enter the data for your groceries check: DATE 5/5/84; CHECK # xxx; PAID TO (market); AMOUNT $xx.xx.

When you finish transferring your data to files, the Transactions menu will be displayed. Choose option 3, Cash, for the sub-menu options display. Now choose option 1 from this sub-menu, Enter Payments/Receipts and enter your lunch expense: MODE Payment; DATE 5/5/84; MERCHANT (restaurant); AMOUNT $xx.xx.

Once this data is recorded, the Cash sub-menu will reappear. Select option 3 to go back to the Transactions menu. From this menu choose option 2 for the Credit Card submenu. Here, select option 1, Enter Purchases/Returns, and enter the amount you charged on your credit card: MODE Purchase; DATE 5/5/84; CARD (service co.); MERCHANT (station); AMOUNT $xx.xx.

Highlights

Start New Month (option 4) allows you to define and record automatic transactions such as monthly rent or utility bills. *Home Accountant* also includes a nice feature known as the "split transaction" which allows you to split an entry, such as

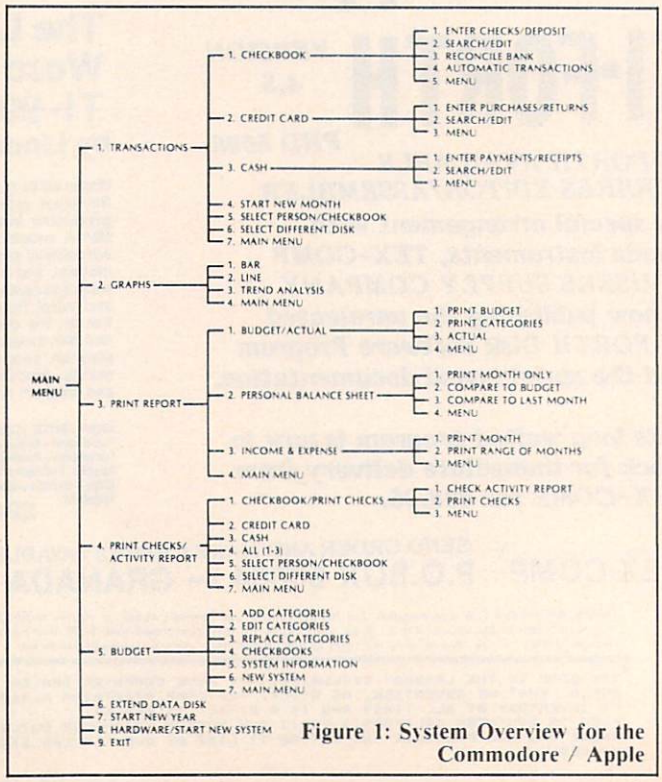


Figure 1: System Overview for the Commodore / Apple

HCM Review

Name:	Home Accountant
Program Type:	Personal budget
Machines:	Apple II family, Commodore 64, IBM PC; * PCjr forthcoming
Distributor:	Continental Software 11223 South Hindry Ave. Los Angeles, CA 90045
Price:	Apple disk $74.95; Commodore 64 disk $74.95; IBM PC disk $150

System Requirements:

Apple: 48K RAM, Apple DOS 3.3, Applesoft in ROM, one disk drive, 80-or 132-column printer with serial or parallel interface card.

IBM PC: 128K RAM, IBM DOS 1.1 or higher, one disk drive, 80-or 132-column printer with serial or parallel interface card.

Commodore 64: VIC-1541 disk drive, 80-or 132-column printer with suitable interface.

Note: Printers are required to obtain a comprehensive listing of all budget categories; video screen displays are limited to the graphs and individual check entries. Data from the Apple version can be merged with a separate *Tax Advantage* program to facilitate tax-return preparation.

	Poor	Fair	Good	Excellent
Performance			▬	
Ease of use			▬	
Documentation			▬	

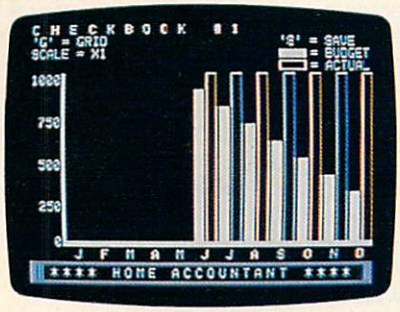
a loan payment, into two categories—perhaps placing the principle in one category and the interest paid into another.

As a part of the data entry format, you can identify any transaction as a tax deduction, or flag it as a check to be cleared at the end of the month with bank statements (this is referred to as Reconcile Bank).

Options 1, Checkbook; 2, Credit Card; and 3, Cash have a *Search/Edit* subroutine that is especially powerful. You can look for a single entry or call a range of transactions, depending on the criteria entered for the search. This function allows for changes and updates in the budget categories before incrementing to a new month.

Home Accountant maximizes your disk space with module 6, Extend Data Disk. If a disk is full or has been closed, you can use this option to erase all cleared (i.e., reconciled) transactions. You have the option of erasing all transactions, but it is advisable to keep two copies as backups.

The program also contains a Graphs module (number 2 from the main menu). Full, high-resolution graphs provide an informative visual aid—a useful feature when the program has accumulated a few months of data to work with. The Bar Graph



This sample screen depicts 7 months accumulated data in one catagory—actual input versus budget projection.

option compares your estimated budget with your actual transactions for any single budget category. The Line Graph will plot point-to-point the activities of one budget category, or three categories simultaneously. The Trend Analysis Graph shows the actual activity in a single budget category. By plotting the information, the graph projects a trend line. Continental Software offers supporting software to obtain a printout of the graphs, but it is not part of this package.

The Print Checks/Activity Report, option 4, provides a hard copy of your budget data. This function is quite versatile because you can (1) obtain summary reports for one month or a range of months; (2) print category titles and their corresponding numbers; and (3) print actual balances or the projected budget amounts (or the calculated difference between them). A 132-column printer is required for a 12-month list of the transactions in each financial category, or for a summary of the budgets for all budget categories (with projected net worth and final net income). [Many 80-column printers can print 132 columns if they are set to condensed mode.—Ed.]

As an aid to small business, the *Home Accountant* package also provides a sample form of pre-printed checks for direct use on a printer. Also, when using the Budget or Transactions modules, you can obtain a hard copy of any data input.

IBM PC Enhancements

The IBM version is dubbed *The Home Accountant Plus* due to enhancements that employ the added capabilities and memory of the PC. Allowances have been made for disk operations employing either single- or double-sided drives, or a hard disk interface. Also, the hardware configuration section for the printer is more flexible for the PC than in versions for the other systems.

Continued on page 64

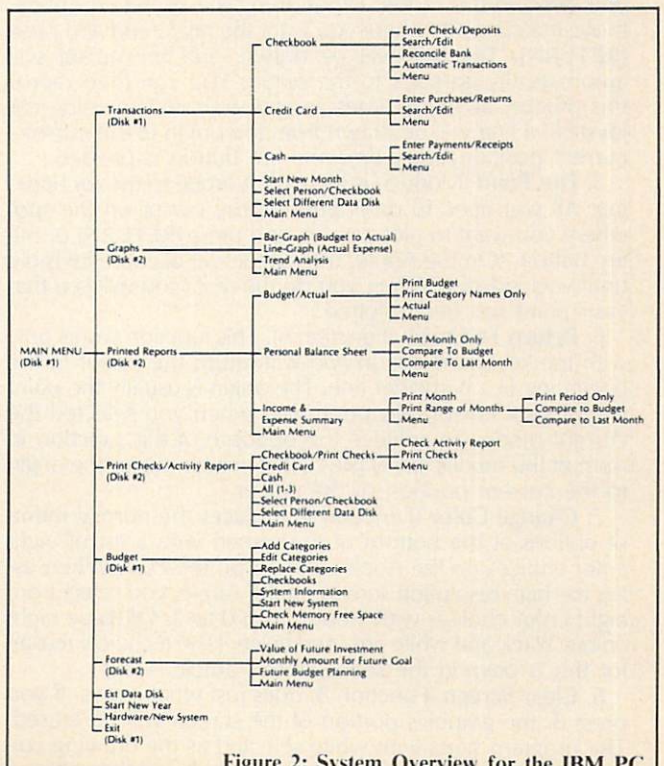


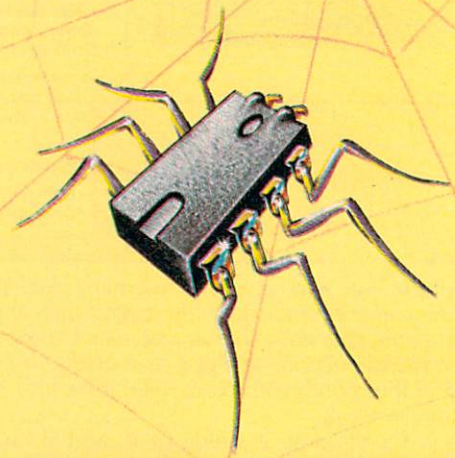
Figure 2: System Overview for the IBM PC

With the *Spider Graphics* program, it's easy to design elaborate graphics patterns on your Apple or IBM PCjr. You can draw lines and rays, plot points, and even erase your mistakes. The best part is that you can do all of this with simple BASIC commands. You don't have to master assembly language routines to create interesting graphics.

An Overview

When you run the program, you will see the title screen flash on for a moment before the graphics screen appears. The bottom of the screen displays two columns of options numbered from 1 to 8. The first five are "modes," and numbers 6 through 8 are specific "functions." These work as follows:

1. **Line Off** (Mode 1) lets you move the cursor anywhere on the screen without drawing or upsetting the rest of your picture. This mode is active when you first run the program.
2. **Line On** (Mode 2) puts you in the line-drawing mode. You have the option of using the keyboard or the joystick to move the cursor. When using the keyboard, movement of the cursor will result in a line drawn from the *origin*, or starting point of the line, to the cursor's new position. When used in keyboard mode, the line will move with the cursor until you press [RETURN]—thus saving the line and updating the origin to the current cursor position. With the joystick, a line will be drawn from the last spot the cursor occupied (when the fire button was pressed) to the current location of the cursor when you press the fire button again. Holding the fire button down while you move the joystick will draw a continuous line.
3. **Erase Line** (Mode 3) lets you erase parts of the screen by sweeping an "invisible line" across the screen. The line starts from the point where the cursor was positioned (when the Erase mode was selected) and will follow the cursor as it moves.To change the origin of the erase line, you can press 3 again. The cursor's position at that time will become the new origin.
4. **Draw Rays** (Mode 4) selects the starting point for the rays. Move the cursor to the position you want and press 4. The center of the rays will be the cursor's position when you selected this mode. When using the keyboard option, move the cursor to where you want the ray to end and press [RETURN]. The line will be drawn, and the cursor will automatically go back to the origin. You can then repeat this process as many times as you wish. When using the joystick, a line will be drawn from the origin to the cursor's current position every time the fire button is pressed.
5. **Plot Point** (Mode 5) is the easiest mode to use for drawing. All you need to do is position the cursor on the spot where you want to plot a point, and press [RETURN] or the fire button. (On the Apple, the cursor will become invisible until you move it. When you do move it, you will see that your point has been plotted.)
6. **Return To Origin** (Function 6). This function works only with the keyboard option and will return the cursor to the beginning of a particular line. The origin is usually the point where your cursor was positioned when you selected the current mode. To achieve the opposite of this function in most of the modes, press [RETURN], which moves the origin to the current position of the cursor.
7. **Change Color** (Function 7) replaces the normal menu of options at the bottom of the screen with a list of eight color choices on the Apple, and 16 on the PCjr. When using the high-resolution screen on the Apple, you select from eight color choices with values from 0 to 7. Of those eight colors, black and white are used twice. [The technical reason for this is beyond the scope of this article.—Ed.]
8. **Clear Screen** (Function 8) does just what it says. If you press 8, the graphics portion of the screen will be erased. The program starts with white selected as the drawing color. To use a different color, select option 7, and then choose the color of your choice from the new menu. You can then move the cursor around to put your graphics on the screen. If you're using joysticks, the fire-button instead of the [RETURN] key causes lines to be drawn or erased. [You should be aware that on the Apple, whatever your choice of colors, some pixels or some vertical lines may be invisible if you're plotting points or drawing lines.You can avoid this by moving your cursor over one pixel.—Ed.]



One of the most endearing characteristics of the Apple *II* family of computers are their powerful graphics capability. The following program was written in Applesoft BASIC on an Apple IIe computer. It should work on any Apple with Applesoft BASIC.

The Apple computer has two different graphics modes: low- and high-resolution. The *Spider Graphics* program uses the high-resolution mode. In this mode, you can draw on the screen with a resolution of 280 pixels by 200 pixels. In other words, the screen is divided up into 280 small dots from left to right, and 200 dots from top to bottom. Screen characters and graphics patterns are created by turning these dots on and off and giving them color.

You *could* attempt to draw a line in Applesoft BASIC by calculating each pixel position on the screen, but you wouldn't want to. The problem is not the complicated math or the loops you would have to program, but the speed of the program. While Applesoft is fairly fast as BASICs go, it is much too slow for this purpose. With that in mind, the designers of Applesoft provided us with the HPLOT command. This command allows you to plot points or lines on the screen. If you supply the command with one set of coordinates, it will plot a single point at those coordinates. If you supply two sets of coordinates, the command plots a line from the first set to the second set. For example, look at the following line:

```
HPLOT 30,60 TO 100,120
```

In this example a line will be drawn from the first coordinate (30,60) to the second coordinate (100,120). If the second coordinate and the word TO were left out of the command, then a single pixel would be plotted at the screen coordinate 30,60.

The Apple computer is a color computer, however, so color plays an important role in the outcome of the HPLOT command on the screen. The screen background in the *Spider Graphics* program is black, so if you try to draw a black line, you will not see any effect unless you draw that line over a different color somewhere on the screen.

This program can move a line around the screen like a piece of string. When you're in the Draw mode (Mode 2) and using the keyboard for input, you will be able to see the line move with the cursor. This is possible because the line is erased just before it is redrawn at the new cursor position.

SPIDER GRAPHICS

by W. K. Balthrop
HCM Staff

This creates the illusion that the line is really moving. The line is effectively erased by redrawing the line in black, the background color. This same effect is used in the Erase mode (Mode 3), except that black is the only color used to draw the line. When the line is passed over an area of the screen which has been drawn on with color, the color is wiped out by the black lines.

You can see the same effect if you select black to draw with when using Function 7. As a result of this erasing before redrawing, the lines will also erase existing lines, even when a colored line passes over them. So to avoid this, you must use caution when creating your graphics.

***A SPECIAL NOTE TO IBM PC OWNERS: If you have an RGBI color monitor with color graphics adapter and BASICA, Spider Graphics is for you too!**

*PCjr

The IBM PCjr version of *Spider Graphics* is similar to the Apple version, with a few refinements. The IBM program gives you 16 colors to work with and a screen width of 320 pixels, as opposed to 8 colors and 280 pixels on the Apple. The IBM version can plot any of its colors in all pixel positions. (The Apple program can plot only certain colors in even or odd pixels.)

The keyboard option of the IBM PCjr version uses the E, S, D, and X keys to move the small diamond-shaped cursor around the screen:

E Moves cursor up.
S Moves cursor left.
D Moves cursor right.
X Moves cursor down.

Within the program, the PUT statement uses input from the keyboard or joystick to move the cursor.

The DRAW command in line 290 draws the shape on the screen; then the GET command puts the shape in an array. Later, when the shape is moved about on the screen, the PUT statement will use the array to place the shape at the coordinates specified.

The XOR option of the PUT statement lets us move the shape over other objects without disturbing them. When this option is used, the shape placed on the screen will convert any color at that position to its opposite. Using a PUT statement twice at the same location will erase the object and restore the original screen.

To use the analog input option, you may have to calibrate the program to fit the device at hand. *Spider Graphics* was written for the PCjr using the IBM analog joystick. If you use a pressure-sensitive pad like the Koala Pad or a different brand of joystick, you can modify the program as follows. [If you do, please report it to us in a Letter to the Editor so that others may share your results.—Ed.]

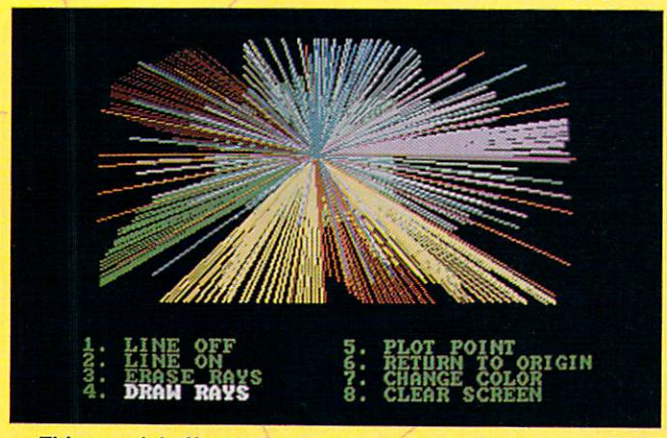
Line 910 reads the joystick ports. The variable PX is assigned to channel 0 of the joystick port, and PY is assigned to channel 1. The command STICK(0) reads all of the joystick ports and returns information for channel 0. STICK(1) returns information for channel 1. Both of these are multiplied by a scaling factor in line 910. In our listings, STICK(0) is multiplied by 2.5. To test this factor for your joystick, run the program with the analog option selected. Move the stick to the farthest right position, or touch the right edge of the pad. The cursor should move to the right edge of the screen. If the cursor goes off the side, then the value 2.5 should be made smaller. If the cursor doesn't reach the edge, then this value should be made larger. The scaling factor for STICK(1) is 1.3. Use the same procedure to check the bottom of the screen, and adjust the value until the cursor moves to a position just above the mode display area.

Continued on next page

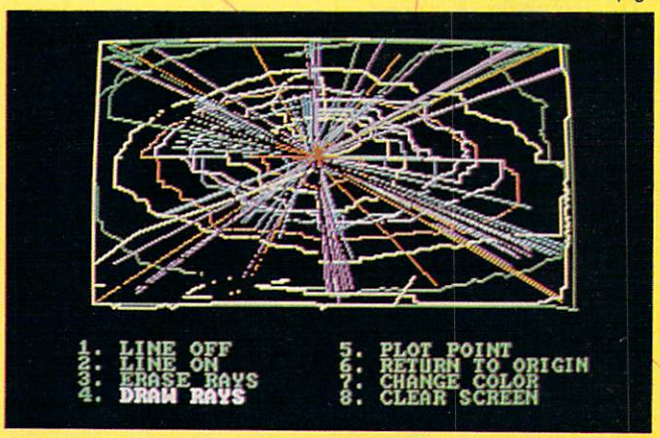
Spider Graphics (Apple) Explanation of the Program

Line Nos.	
100-170	Program header.
180-190	Place screen in text mode; display title.
200	Set up shape table.
210-250	Display screen options.
260-280	Input response.
290-320	Update cursor coordinates.
330	Branch to mode routines for [RETURN] key.
340	Get next input from keyboard.
350-370	Draw a line.
380-390	Erase a line.
400	Move the cursor.
410	Move origin to current cursor position.
420	Draw line; return to origin.
430	Plot a single point.
440-490	Set up the five modes.
500	Return to origin.
510-560	Input color routine.
570	Clear the graphics screen.
580-640	Turn off mode indicator for previous mode.
650	Shape table DATA.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.



This special-effects screen was created with the program's DRAW RAYS option, using a joysick for input.



This colorful spider's web was created by using the DRAW RAYS and LINE ON options together.

Circle 64 on Reader Service Card.

Circle 65 on Reader Service Card.

Circle 66 on Reader Service Card.

Spider

The built-in keyboard buffer lets you type ahead of the computer; you don't have to wait for the computer to finish an action before entering the next step. This works fine until you're trying to control motion on the screen with the cursor, as we do in *Spider Graphics*. In the keyboard mode, you must move the cursor by pressing keys. If the buffer is being used, the computer may read more keypresses than you actually intended. The key input routine has the buffer activated in the program because key input is jerky without it. But for those who would prefer not to have the buffer activated, here is the code to eliminate it:

```
1040 A$=INKEY$:DEF SEG=0:POKE 1050, PEEK(1052):RETURN
```

If you use this line instead of line 1040 as listed, the keyboard buffer will not work and the cursor will move only while you hold down the keys. One drawback of this is that cursor motion will be slower and a little jerkier. You may also have to hold the key down a little longer so that line 1040 has time to scan the keyboard.

Spider Graphics (IBM PCjr)
Explanation of the Program

Line Nos.	
100-180	Program header.
190	Display title screen.
200-290	Input option for keyboard or analog; initialize variables, shapes, and screen mode.
300-350	Display modes and function at bottom of screen.
360-500	Input routine for keyboard option.
510-560	[ENTER] key subroutines.
570-650	Change mode subroutines.
660-780	Functions subroutines.
790-870	Turn off old mode indicator.
880-930	Analog option routine.
940-1000	Fire button subroutine (analog mode.)
1010-1040	Key input subroutine.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

MAKE SOMEONE HAPPY!

AND GET FREE SOFTWARE TO BOOT!

Give a gift-subscription to

HOME COMPUTER™ magazine

And we'll send *YOU* 2 issues of ON TAPE™ or ON DISK™ ABSOLUTELY FREE!

Comments From Our Happy Readers

"I purchased my computer without knowing about you, but after seeing many other "computer" magazines, you are light years ahead of the pack."
Tim Williams, Gig Harbor, WA

"...your publication is better than Compute, Byte, Popular Computing and Personal Computing. We have cancelled our other subscriptions..."

This is the first time I've ever written a magazine publication and it's only because I'm impressed with your magazine."
Cindy Eckhardt, Palatka, FL

"...I had considered BYTE the most useful of the genre until I saw yours. Absolutely the most valuable source of ideas..."
Dennis E. Clouse, Concord, CA

"I am a programmer/analyst and I just subscribed to your magazine.

The question is where have you been? I was looking for just the magazine you and your staff publish. I must say it is the best. I have subscribed to Byte, Softside, Creative Computing and Kilobaud, and nothing—can compare!!"
Sal Melillo, Scotch Plains, NJ

"...Your magazine is a help as well as a great source for software games, books, and hardware. For me the super-novice, this information is invaluable! Keep up the good work!"
David Hurd, Lubbock, TX

"...After spending a small fortune subscribing to computer magazines...the investment has finally realized some fantastic gains with your magazine."
Robert L. Nelson, Suring, WI

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.

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** will receive **2 months** of our magazine program service—ON TAPE™ or ON DISK™—**FREE!!**

PLEASE ENTER A 1-YEAR GIFT-SUBSCRIPTION FOR:

Name _____

Address _____

City _____ State _____ Zip _____

Does recipient have a home computer?

☐No, not yet

☐Yes — What Brand? _____

FROM: (Gift-giver must complete this section)

Name _____

Address _____

City _____ State _____ Zip _____

Subscriber label code number of gift giver

SATISFACTION GUARANTEED

—or the unfilled portion of the subscription will be refunded, less the cost of any premium items you have received.

FREE SOFTWARE FOR GIFT-GIVERS

Indicate your choice of media: ☐ON TAPE™ ☐ON DISK™

Indicate which computer media is for:(pick one)

☐Apple ☐C-64 ☐VIC-20 ☐IBM PC ☐IBM PCjr ☐TI-99/4A

Select your 2 FREE software issues

☐Vol. 4, No. 1 ☐Vol. 4, No. 2 ☐Vol. 4, No. 3

☐Vol. 4, No. 4 ☐Vol. 4, No. 5 ☐Vol. 4, No. 6

☐Enclosed is $25 for each 1-year gift-subscription

Canada add $7; Foreign Surface add $18 for 1-yr magazine subscription only. Free software not available outside USA.

☐Check or Money Order Enclosed

MUST BE IN US FUNDS DRAWN ON A US BANK

Bill my ☐VISA ☐MasterCard

Account No. _____

Signature _____ Exp. Date _____

Enclose payment or credit card information and mail with completed form to:

Home Computer Magazine
P.O. Box 5537 • Eugene, OR 97405

Or use our TOLL-FREE ORDER LINE
FOR VISA/MasterCard orders only:
1-800-828-2212
In Oregon, Alaska, Hawaii Tel. (503)485-8796
Allow 6-8 weeks for your first issue.

HCM Review

MATRIX LAYOUT

	Spreadsheet Total		Screen Display	
	Rows	Cols.	Rows	Cols.
TI cassette	16	14	11	2
TI disk	24	15	11	2
C-64 disk	40	25	13	3
VIC-20 cassette	40	25	13	3

Name:	Count-Sil
Program Type:	Spreadsheet
Machines:	TI-99/4A, Commodore 64, VIC-20
Distributor:	Systems Interface 1511 Merivale Road Lancaster Mall Nepean, Ontario K2G 3J3
Price:	TI-99/4A disk $49.95; cassette $29.95 Commodore 64 disk $29.95 VIC-20 cassette $29.95

System Requirements:
Commodore 64: VIC-1541 disk drive
VIC-20: 16K Memory Expansion, cassette recorder.
TI-99/4A: TI Extended BASIC, 32K Memory Expansion, disk system or cassette recorder.

	Poor	Fair	Good	Excellent
Performance			■	
Ease of Use				■
Documentation				■

Count-Sil

A review by Tom Green
HCM Staff

Some pay hundreds of dollars for a fancy spreadsheet program. Here's an inexpensive alternative from Systems Interface. *Count-Sil* is relatively slow—being written in BASIC—and uncomplicated; but for limited home applications, it may be well worth the money. Those who can make most use of the program, however, may need some introduction to the subject.

What's A Spreadsheet?

Imagine a business form with, say, a matrix of 20 columns by 30 rows. If each column has a width of 10 characters, the form would be too wide and too deep to be displayed on conventional monitors. To simulate this type of "spreadsheet" business form —and even larger ones with hundreds of columns and rows—spreadsheet programs allocate computer memory for storing complete *templates* (each row and column of the matrix labeled with headers) and associated data; the screen is just used as a floating window or viewfinder to display various sections of the form.

The matrix layout and maximum spreadsheet size for *Count-Sil*, reviewed here, differs somewhat from one system implementation to another:

First Appearances

The disk versions all have the same main menu selections: 1. Run Count-Sil; 2. Catalog Disk; 3. Delete File; 4. Exit.

The cassette version has three options: 1. Count-Sil; 2. Print Spreadsheet; 3. and Exit.

Once the program has completed its initializing process, the matrix is displayed. The areas within the row and column line divisions are called "cells." Below the matrix is the program's "work area" for command input. The word COMMAND with a flashing prompt indicator is presented when the program is ready for input.

Template Design

Count-Sil has 19 commands that aid in creating templates. Seven of these commands affect structure and cursor control; the remaining 12 commands are connected with handling data.

The TI version allows titles only at the head of a column or start of a row, but the C-64 version permits labeling inside the cells.

To display a list of the commands available on the disk version you can press H for Help. This feature lists each option with a summary of its function. (See Chart 1.) Four of these commands are not available on the TI cassette version of *Count-Sil*: the (H)elp, (I)nsert, (K)opy, and cursor (M)ovement commands. Within the framework of the template created, data input and manipulation will complete the spreadsheet.

Customizing Your Spreadsheet

Arithmetic manipulation of data is one of the most important assets of spreadsheet programs. *Count-Sil* allows you to define formulas or "expressions" that will perform computations using data from particular cell locations. The format is as follows: X=A+B+C, where X can be a column, row, or designated cell to display the result; A, B, and C can be a column, row, or designated cell used for computation; + represents any operand function of addition, subtraction, multiplication, division, or exponential. One hundred such expressions can be defined on the disk versions and the VIC-20 cassette version; 69 expressions are allowed for the TI cassette version.

Using the Z command you can define a sequence of "work registers." The values stored in these registers can be used as numeric constants in the equations you set up. The disk and the VIC-20 cassette versions have up to 100 such registers; the TI cassette version has up to 71 work registers available.

Chart 1

HELP FILE

A- Select calculation decimal ACCURACY
B- Set BACKGROUND/BORDER color
C- COMPUTE expressions, all or partial
D- Change number of displayed DECIMALS
E- Enter EXPRESSIONS
H- Display HELP file
I- INSERT/DELETE rows or columns
J- JUMP to new spreadsheet location
K- COPY rows or columns of values
L- LOAD data files from storage
M- Change automatic cursor M0VEMENT
P- PRINT spreadsheet data
R- Enter ROW titles
S- SAVE spreadsheet data
T- Enter column TITLES
V- Select VALUE MODE for value entry
W- WIPES out selected data
X- EXIT MODE to system, or main menu
Z- Define work REGISTERS

As the spreadsheet takes shape, certain expressions may require computation results to be used as inputs for other cell locations. Pressing C (for Compute) allows you to perform expression computations one at a time or globally (all at once in sequence).

Upon completion, the spreadsheet can be Saved to disk or cassette. The print option of the cassette version is on the main menu and is limited to printing the spreadsheet as contained in memory.

Documentation

The user's guide supplied with the package is simple, yet thorough enough to ensure proper program execution. Two tutorial spreadsheets are quite informative in structuring procedures, forming arithmetic equations, and manipulating the work registers. You can use the preformatted spreadsheets to practice command manipulation, and to print hard copies as visual aids for template formatting.

Summary

Count-Sil's performance can be attributed to the language used to write the program—BASIC. Its command response is slow and certain data and format procedures take too much time to complete. And because the program was written for relatively small memory capacity requirements, its features are limited. The commands that are offered, however, provide enough power and flexibility to produce quality spreadsheets. With some practice in planning formats and a creative approach to their application, *Count-Sil* could introduce you the powerful world of computerized spreadsheets.

HCM

Home Accountant . . . from p. 57

Home Accountant Plus's main menu has a unique feature entitled Forecast Module. This subroutine is like having a mini-spreadsheet at your immediate disposal. You can project to a future year the value of an investment at a specified rate of return—while allowing for an anticipated inflation rate. You can also figure out how much you need to save every month (week, year, etc.) to reach a certain sum, and adjust it for variables like inflation. This option also has full print and graph functions.

Continental Software's soon-to-be-released version of *Home Accountant* for the PCjr is expected to have all of the features found on its PC version.

Documentation

Every *Home Accountant* package comes with a user's guide that is written with the beginner in mind. It is always a good idea to heed the precautions found in any instructional material, and one warning mentioned in this manual deserves special attention: For all three versions, it is of utmost importance to follow the exact method of exiting the program. Otherwise, you may have to go through the set-up procedures again, or you may lose your data.

By using diagrams of the different screen menus and explaining each input line, the manual takes the guess-work out of preparing your budget. Routines for initializing data disks and for loading the program are discussed at length. The manual also contains a code chart needed to interface the more popular printers with this program.

Even with the extensive tutorial, many will find *Home Accountant* a big fish to swallow. After taking a good look at this review, you may even decide *not* to go fishing. A program of this size and complexity is bound to make some beginners recoil in horror. The many subroutines involved with every transaction contribute to a considerable investment of time, and can be overwhelming to a new user. Those with more computer experience—and with real dedication to detailing their personal finances—will naturally gain the most from this package.

Nothing Ventured . . .

The flexibility and nominal price of *Home Accountant* explains why it has stayed on top of the charts for so long. The program handles such a broad spectrum of information, however, that it will take a while for any user to realize the full potential of the software. Nevertheless, many computer owners will discover that the rewards for this expended effort into home money management can be fruitful, indeed. And for consumers without a computer of their own, a demonstration of the benefits of a package like *Home Accountant* might be just the reason they're looking for to take the home computer plunge.

HCM

HCM Review

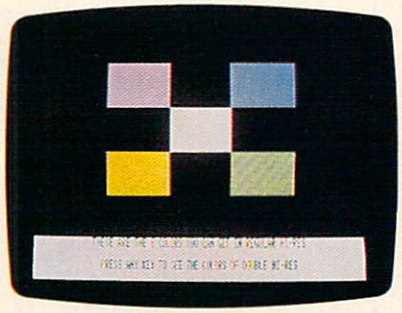
Name:	Doublestuff
Program Type:	Language/Graphics Enhancer
Authors:	Louis Bonfiglio and Peter Joselow
Machine:	Apple IIe
Distributor:	Doublestuff Software Inc. 2053 West 11th Street Brooklyn, New York 11223
Price:	$39.95, Apple diskette

System Requirements:
80-column card for double lo-res, or
Extended 80-column card for both double lo-res and double hi-res with color monitor.

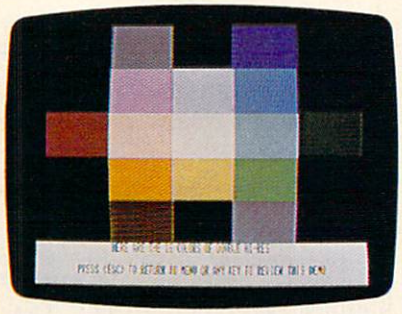
	Poor	Fair	Good	Excellent
Performance				
Ease of use				
Documentation				

STUFF STUFF DOUBLESTUFF

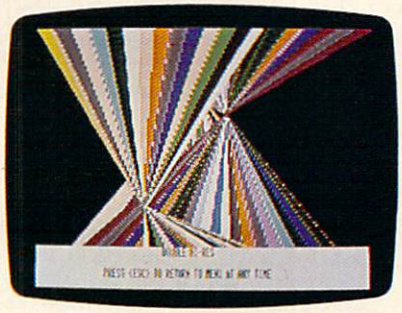
A Review by
Roger Wood and Wayne Koberstein
HCM Staff



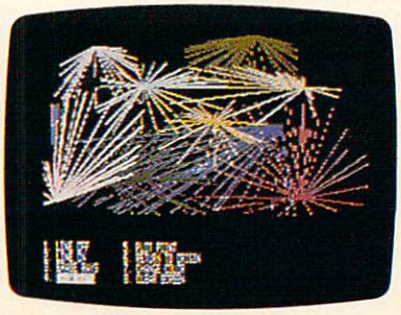
Apple's hi-res screen BEFORE *Doublestuff*.
This shows the six regular colors (including black and white) of normal hi-res graphics.



Apple's hi-res screen AFTER *Doublestuff*.
The pallette is now increased—through a process called "dithering"—to include sixteen different colors.



This spectacular demonstration screen shows a multi-hued pattern featuring both diagonal and near vertical lines. Notice that the striation lines are increasingly visible in lines approaching true vertical.



Doublestuff and *Spider Graphics*.
A full array of colors has now become available on the Apple for this BASIC program translated into the *Doublestuff* language.

Have you noticed something missing from Apple graphics? A lo-resolution (lo-res) screen, although it provides 16 colors, is indeed *very* lo-res—you may plot only in monochrome squares, each the size of one 40-column text character. A hi-resolution (hi-res) screen, with its better definition, allows you only six different colors. *Doublestuff*, from Doublestuff Software Development, Inc., promises to overcome these limitations of color choice and resolution—and it *delivers*.

This software is more than a simple utility. It is actually a language—a form of Applesoft BASIC stored in the same memory location reserved for Integer BASIC. In a sense, it is also a learning tool. You can easily modify all of your Applesoft BASIC programs to take advantage of *Doublestuff*'s improved graphics.

But before you can use the software, you must have an 80-column card in the auxillary slot. In addition, pins 50 and 55 must be jumpered together with a small plastic connecter on the J1 jumper on the card. You should receive one of these when you purchase an 80-column card; if you don't have the jumper, you can usually obtain one from your dealer. The regular 80-column card only supports double *lo-res* graphics. For double *hi-res* graphics, the Extended 80-column card (the card with the additional 64K of RAM) is required.

Your Screen Is Easel And Canvas . . .

Once installed, *Doublestuff* provides twice the normal screen resolution, in both lo-res and in hi-res modes. It increases the lo-res screen from 40 to 80 columns, each column half as wide as its original. This literally doubles the amount of color-blocks available in low-res mode. The number of dot-columns on the hi-res screen increases from 280 to 560. In addition to doubling the resolution, *Doublestuff* "dithers" (mixes) different hi-res colors, forming a total of 16 color choices. Then the fun really begins.

Although *Doublestuff* does deliver on its promises, it does not completely solve the problem of color placement in Apple graphics—nothing can. The columner structure of the hi-res Apple graphics screen limits where you can place any given color: Color dots in *even* columns can be only black, purple, or blue. Dots in *odd* columns can be only black, green, or orange. (White can appear in any column as long as two adjacent dots are "on," with a bit-value of 1.) This limitation is imbedded in the Apple hardware itself.

Doublestuff creates its 16-color palette by dithering Applesoft's regular hi-res colors. For example, yellow is dithered from purple, orange, and green. Due to Apple graphic's columner structure, dithering works horizontally, but not vertically—i.e., if you draw a horizontal yellow line, you will get a yellow line. But if you draw a one-pixel-wide vertical yellow line, you will get only one of its component colors, because any one dot-column will show only the color assigned to it.

Thus, as the pixel-wide lines become more vertical, dithering becomes less effective. Thin diagonal lines will not be a solid color, but will instead show striations of the component colors. You can, however, color-in areas by close-packing diagonal

Continued on next page

Doublestuff

lines. The component colors will then be horizontally adjacent, and will blend into one solid hue.

Don't let these details drive you away. With *Doublestuff*, you still come out ahead. Even with its hardware-related obstacles, this software's fine resolution and improved palette still bring out the colorful best in your Apple IIe.

Doublestuff takes you right to the heart of Applesoft BASIC. And, if you are a beginner, it will teach you a great deal about BASIC programming concepts. This software features an excellent on-disk demonstration with knock-out graphics. And its instructions are presented as a very lucid tutorial. Even experienced programmers can learn from the clear and useful documentation.

Programming in *Doublestuff*

To see how *Doublestuff* could be used as a programming tool, we modified the *Spider Graphics* program (in this issue of HCM) to use *Doublestuff* BASIC. The modifications required were surprisingly few—they mostly involved changing limit checks and expanding the Change Color option to include all 16 colors available with double hi-res. Here's a list of the changes:

```
160 REM APPLE II SERIES DOUBLESTUFF
270 PX = 280:PY = 100:MODE = 1:PDX = 280:PDY = 100
280 HCOLOR= 0: HGR : HCOLOR= 3: SCALE= 1:IF KB = 1 THEN XDRAW 1 AT PX,PY
410 IF (A = 2 OR A = 32) AND PX < 559 THEN MX = 1: GOTO 460
610 HOME : VTAB 21: PRINT "0. BLACK" TAB( 7)"1. RED" TAB( 9)"2. VIOLET" TAB( 6)"3. DK. BLUE" TAB( 4)"4. DK. GREEN" TAB( 2)"5. GRAY(1)"
620 PRINT "6. MED. BLUE" TAB(3)"7. BLUE" TAB( 8)"8. BROWN" TAB( 7)"9. ORANGE" TAB( 6)"10. GRAY(2)
630 PRINT "11. PINK" TAB( 7)"12. LT. GREEN" TAB( 2)"13. YELLOW" TAB( 5)"14. AQUA" TAB( 7)"15. WHITE"
650 INPUT A$:A = VAL (A$): IF A < 0 OR A > 15 THEN GOTO 650
670 HCOLOR= 0: HGR : GOTO 330
760 PX = PDL (0) * 2.1
```

Lines 280 and 670 demonstrate a unique feature of *Doublestuff*: Normally, in Applesoft BASIC, an HGR statement clears the screen to black—not so with *Doublestuff*. Instead, the screen is cleared to the color named in the last HCOLOR= statement.

With the minor changes listed above, *Spider Graphics* has access to all 16 hi-res colors available with *Doublestuff*. This modified *Spider Graphics* program is an ideal tool for visually exploring the doubled graphics capabilities of the Apple. So if you're tired of that small palette, don't stuff it—*Doublestuff* it! **HCM**

[Note: Doublestuff Software Development Inc. recently released an expanded version of this software called Doublestuff + *($69.95), which includes a complete drawing package. Also, they soon will release new versions that will run on the Apple IIc as well as the Apple IIe. We were unable to obtain copies of these versions in time for review.]*

HOME COMPUTER™
PROGRAM LISTINGS

PROGRAM LISTING

CONTENTS

Your Guide to Typing in Programs from *HCM*

Within these pages is a software bonanza: entertainment, education, home and business applications, utilities, and tutorials—just for you. All you need to do is type them into your computer. *HCM* has taken most of the strain out of this process with:

- Typeset listings with numbers in boldface type.
- A bold, double vertical bar separating the line numbers from the program statements in BASIC listings.
- A vertical background grid to aid entry of the spaces.

Look at the Key-in Reference (Figure 1 below) to see how each character actually appears in the listing. By checking any questionable characters with the Key-in Reference, you can reduce errors to a minimum.

Figure 1: KEY-IN-REFERENCE

Before You Begin

Since *HCM* publishes programs for several different computers, the first thing you should do is make sure that you are looking at the listing designed for your machine. If, for example, you have an Apple IIe, make sure you look for the following black bar above the listing: APPLE II Series The computer model name will likewise appear on each subsequent page of each listing, so always look for the name before you begin typing from a new page of listings.

Before you begin typing in the program, you will want to set up a system to save your program. Whether you are using a cassette or diskette storage system, now is the time to be certain it is properly connected, powered up, and loaded with a blank cassette or an initialized disk. As you type in your program, you should get in the habit of saving your work after every twenty or so lines.

One of the most common errors in entering a listing is typing one symbol for another. These transpositions include substituting the letter O for the number 0, the letter I for the number 1, the letter S for the $, and the uppercase B for the number 8. This last error is especially likely when working in hexadecimal numbers which are composed of 0-9 and the uppercase letters A-F.

The listings in *HCM* are always the same number of characters wide, but, the number of characters put on any line of the video display will vary from computer to computer. Don't try to make your listing *look like* the type-set listing—instead make sure you key in the listings character for character and space for space.

A Special Note on Listings for the C-64 and VIC-20

Commodore uses more than 90 special symbols to represent various keyboard operations: for instance, the symbol [♥] in a program represents the operation of holding down the [SHIFT] key and pressing the key which has CLR on its upper half (second key from the right on the top row). This operation clears the screen.

Rather than reproducing these symbols, *HCM*'s listings include key-stroke instructions between two hands with pointing fingers. For example, when you find ☛ SHIFT CLR ☚ in an *HCM* listing, you will know to hold down the [SHIFT] key and press the key with CLR on it.

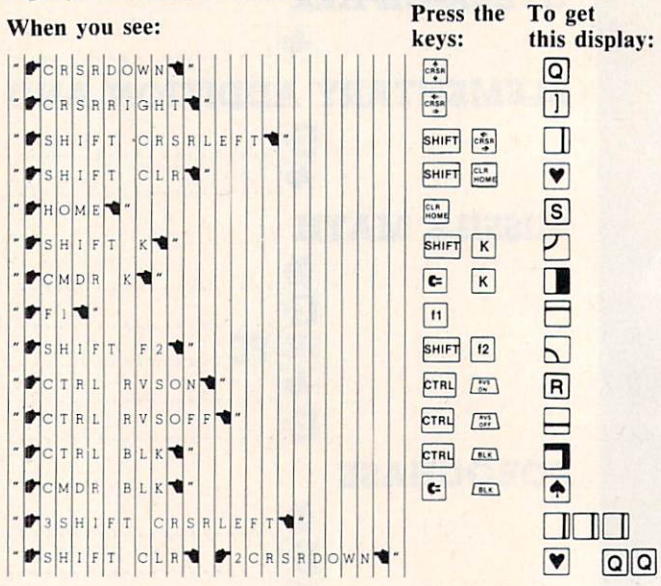
A number is included if you need to repeat the operation: ☛ 8SHIFT CRSRLEFT ☚ tells you to hold the [SHIFT] key down and press the cursor left key (on the bottom right of the keyboard) eight times.

When you come to the hand symbols, remember:

- Each operation is enclosed in its own set of hand symbols.
- If any key action requires you to press two keys, press the control key or the Commodore key or the shift key. *first* and hold it down before pressing the second key.
- Not to enter spaces or anything else *within* a set of hands.
- Everything between a pair of hand symbols is set in a different typeface.

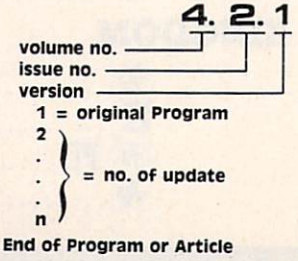
In Figure 2 below, we have included a chart showing you a representative sample of the symbols that appear when you use keystrokes enclosed by the hand symbols.

Figure 2: SPECIAL SYMBOLS CHART FOR C-64 and VIC 20



Program Identification

Each program header (the first few lines of the program) contains information giving the language the program is written in (e.g., TI Extended BASIC, Applesoft, etc.) and any special system components that are required (special memory cards, Speech Synthesizer, etc.). The first two digits of the version number tell you in which volume and issue of *HCM* the program *initially* appeared. The third digit of the version number indicates the version of the program. When a program initially appears, in *HCM*, it is version 1. Any subsequent revisions to the program if later published in the magazine or in the software available on magnetic medium from *HCM* will bear a revised version number.



BARS AND PLOTS

TI-99/4A

```
100 REM  **********************
110 REM  *  BARS  AND  GRAPHS  *
120 REM  **********************
130 REM  BY JOHN GUNTER
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.3.1
160 REM  TI BASIC
170 CALL CLEAR
180 PRINT TAB(7);"BARS AND GRAPHS":::::
    :::
190 INPUT "USE A PRINTER (Y/N)?":P$
200 IF (P$<>"Y")*(P$<>"N")THEN 190
210 IF P$="N" THEN 290
220 PRNT=1
230 INPUT "PRINTER DEVICE NAME?":DEV$
240 GOTO 290
250 CALL KEY(0,K,S)
260 IF (S=0)+(K<>13)THEN 250
270 CALL CLEAR
280 RETURN
290 PRINT TAB(10);"PLEASE WAIT"::
300 PRINT TAB(12);"GOTTA REST"
310 DIM CO(32)
320 DIM A$(11)
330 A$(1)="0000000000000000007E"
340 A$(2)="000000000000007E7E"
350 A$(3)="00000000007E7E7E7E"
360 A$(4)="0000000007E7E7E7E7E"
370 A$(5)="00000007E7E7E7E7E7E"
380 A$(6)="000007E7E7E7E7E7E7E"
390 A$(7)="007E7E7E7E7E7E7E7E"
400 A$(8)="7E7E7E7E7E7E7E7E7E"
410 A$(9)="0101010101010101"
420 A$(10)="01010101010101FF"
430 A$(11)="FF01010101010101"
440 CALL SCREEN(13)
450 CALL COLOR(9,5,1)
460 CALL COLOR(10,9,1)
470 CALL COLOR(11,11,1)
480 CALL COLOR(12,15,1)
490 COL(1)=5
500 COL(2)=9
510 COL(3)=11
520 COL(4)=15
530 FOR X=95 TO 158 STEP 8
540 FOR CH=1 TO 8
550 CALL CHAR(X+CH,A$(CH))
560 NEXT CH
570 NEXT X
580 CALL CHAR(42,"935539FF395593FF")
590 CALL CHAR(91,"00000000000000FF")
600 CALL CHAR(92,A$(9))
610 CALL CHAR(93,A$(10))
620 CALL CHAR(94,A$(11))
630 CALL CLEAR
640 SET=0
650 CALL VCHAR(1,7,92,21)
660 FOR X=1 TO 21
670 CALL HCHAR(X,8,91,25)
680 NEXT X
690 CALL HCHAR(21,8,93,24)
700 FOR TIC=5 TO 20 STEP 5
710 CALL HCHAR(TIC,7,93)
720 NEXT TIC
730 CALL HCHAR(1,7,94)
740 ST$="MAXIMUM DATA VALUE ? "
750 GOSUB 770
760 GOTO 1190
770 D1$=D$
780 D$=""
790 FOR ST=1 TO LEN(ST$)
800 CALL HCHAR(24,2+ST,ASC(SEG$(ST$,ST,
    1)))
810 NEXT ST
820 KI=1
830 CALL SOUND(100,800,2)
840 CALL HCHAR(23,7,42)
850 CALL KEY(0,K,S)
860 IF S=0 THEN 850
870 IF K=13 THEN 1140
880 IF K=7 THEN 1080
890 IF K=8 THEN 980
900 IF K=9 THEN 930
910 IF K<>32 THEN 1030
920 IF (ST$="SIDE LABEL ? ")+(ST$="BOTT
    OM LABEL ?")+(ST$="LEGEND 1")+(ST$=
    "LEGEND 2")THEN 1030
930 2340+1>32 THEN 840
940 CALL HCHAR(21,M,93)
950 M=M+1
960 CALL HCHAR(21,M,42)
970 GOTO 840
980 IF M-1<8 THEN 840
990 CALL HCHAR(21,M,93)
1000 M=M-1
1010 CALL HCHAR(21,M,42)
1020 GOTO 840
1030 IF KI>25 THEN 840
1040 CALL HCHAR(23,7+KI,K)
1050 D$=D$&CHR$(K)
1060 KI=KI+1
1070 GOTO 840
1080 CALL SOUND(300,300,2)
1090 FOR ES=KI TO 1 STEP -1
1100 CALL HCHAR(23,7+ES,32)
1110 NEXT ES
1120 D$=""
1130 GOTO 820
1140 CALL HCHAR(23,1,32,32)
1150 CALL HCHAR(24,1,32,32)
1160 IF D$<>"" THEN 1180
1170 D$=D1$
1180 RETURN
1190 FOR X=1 TO LEN(D$)
1200 IF (ASC(SEG$(D$,X,1))<48)+(ASC(SEG$
    (D$,X,1))>57)THEN 1240
1210 NEXT X
1220 IF (LEN(D$)<1)+(LEN(D$)>5)THEN 1240
1230 GOTO 1260
1240 GOSUB 1870
1250 GOTO 740
1260 MD=VAL(D$)
1270 SCA=(INT(MD/20)*20)+20
1280 IF INT(MD/20)<>MD/20 THEN 1300
1290 SCA=SCA-20
1300 DIV=100
1310 LA$=STR$(SCA)
1320 FOR MK=1 TO 21 STEP 5
1330 FOR LAB=1 TO LEN(LA$)
1340 CALL HCHAR(MK,2+LAB,ASC(SEG$(LA$,LA
    B,1)))
1350 NEXT LAB
1360 DIV=DIV-25
1370 SCA1=SCA*(DIV/100)
1380 LA$=STR$(SCA1)
1390 NEXT MK
1400 ST$="SIDE LABEL ? "
1410 GOSUB 770
1420 IF LEN(D$)<25 THEN 1440
1430 D$=SEG$(D$,1,24)
1440 FOR LAB=1 TO LEN(D$)
1450 CALL VCHAR(LAB,2,ASC(SEG$(D$,LAB,1)
    ))
1460 NEXT LAB
1470 ST$="BOTTOM LABEL ?"
1480 GOSUB 770
1490 R=22
1500 Y=7
1510 GOSUB 1530
1520 GOTO 1570
1530 FOR LAB=1 TO LEN(D$)
1540 CALL HCHAR(R,Y+LAB,ASC(SEG$(D$,LAB,
    1)))
1550 NEXT LAB
1560 RETURN
1570 M=8
1580 ST$="COLOR?1-BLU,2-RED,3-YEL,4-GY"
1590 GOSUB 770
1600 IF (ASC(D$)<49)+(ASC(D$)>52)+(LEN(D
    $)>1)THEN 1580
1610 C=VAL(D$)-1
1620 IF SET=0 THEN 1770
1630 CH=127
1640 CH1=135
1650 SE=SET-1
1660 CST=13+SET-1
1670 IF CST<17 THEN 1790
1680 CST=CST-1
1690 SE=SE-1
1700 CALL SOUND(300,300,2)
1710 D$="COLOR FULL-ENTER ERASE OR END"
1720 Y=1
1730 R=24
1740 GOSUB 1530
1750 ST$=""
1760 GOTO 1800
1770 CH=95
1780 CH1=103
1790 ST$="ENTER DATA"
1800 CALL HCHAR(21,M,42)
1810 CALL HCHAR(21,M-1,93)
1820 GOSUB 770
1830 IF D$<>"C" THEN 1900
1840 IF SET=0 THEN 1580
1850 SET=SET+1
1860 GOTO 1580
1870 CALL SOUND(300,300,2)
1880 ST$="BAD DATA TRY AGAIN"
1890 RETURN
1900 IF D$="END" THEN 2330
1910 IF D$<>"ERASE" THEN 1960
1920 CALL HCHAR(21,M,93)
1930 CALL VCHAR(2,M,91,19)
1940 CALL HCHAR(1,M,95)
1950 GOTO 1770
1960 IF (D$="P")*(PRNT=1)THEN 2560
1970 IF D$="NEW" THEN 630
1980 FOR X=1 TO LEN(D$)
1990 D=ASC(SEG$(D$,X,1))
2000 IF (D<47)+(D>57)THEN 2030
2010 NEXT X
2020 GOTO 2050
2030 GOSUB 1870
2040 GOTO 1820
2050 D=VAL(D$)
2060 IF D>SCA THEN 2080
2070 GOTO 2100
2080 GOSUB 1870
```

Continued

PROGRAM LISTING

BARS AND PLOTS

Continued

TI-99/4A

```
2090 GOTO 1820
2100 BAR=D/(SCA/20)
2110 HT=INT(BAR)
2120 RE=BAR-HT
2130 TP=1+INT((RE*8)+.5)
2140 IF TP<9 THEN 2160
2150 TP=8
2160 SE=C
2170 CALL CHAR(CH+TP+(SE*8),A$(TP))
2180 CALL CHAR(CH1+(SE*8),A$(8))
2190 CALL VCHAR(21-HT,M,CH1+(SE*8),HT)
2200 CO(M)=C
2210 IF HT>19 THEN 2250
2220 CALL HCHAR(20-HT,M,CH+TP+(SE*8))
2230 CALL VCHAR(1,M,91,19-HT)
2240 GOTO 2260
2250 CALL HCHAR(21-HT,M,CH1)
2260 IF M<32 THEN 2310
2270 CALL SOUND(300,300,2)
2280 ST$="DATA FULL"
2290 GOSUB 770
2300 GOTO 1790
2310 M=M+1
2320 GOTO 1790
2330 CALL SOUND(500,800,2)
2340 ST$="LEGEND 1"
2350 GOSUB 770
2360 D2$=D$
2370 ST$="LEGEND 2"
2380 GOSUB 770
2390 D3$=D$
2400 D$=D2$
2410 R=23
2420 Y=7
2430 GOSUB 1530
2440 D$=D3$
2450 R=24
2460 Y=7
2470 GOSUB 1530
2480 CALL KEY(0,K,S)
2490 IF S=0 THEN 2480
2500 IF K=80 THEN 2560
2510 IF K=78 THEN 630
2520 IF K=83 THEN 2540
2530 GOTO 2480
2540 CALL CLEAR
2550 END
2560 OPEN #1:DEV$
2570 FOR X=1 TO 24
2580 FOR P=1 TO 32
2590 CALL GCHAR(X,P,Z)
2600 IF Z<96 THEN 2620
2610 Z=35
2620 IF (Z<91)+(Z>94)THEN 2640
2630 Z=95
2640 PRINT #1:CHR$(Z);
2650 NEXT P
2660 PRINT #1:""
2670 NEXT X
2680 CLOSE #1
2690 GOTO 1820
```

HCM

BARS AND PLOTS

VIC-20

```
100 REM ***********************
110 REM * BARS AND GRAPHS *
120 REM ***********************
130 REM BY JOHN GUNTER
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.3.1
170 REM V20 BASIC
180 POKE52,28:POKE56,28:PRINT"■SHIFT CLR■"
190 FORI=7168TO7679:POKEI,PEEK(I+25600):NEXT:POKE36869,255:POKE36879,25
200 FORI=7432TO7431+(9*8):READA:POKEI,A:NEXT
210 CL=30720
220 GOSUB 630
230 DATA 0,0,0,0,0,0,0,254
240 DATA 0,0,0,0,0,0,254,254
250 DATA 0,0,0,0,0,254,254,254
260 DATA 0,0,0,0,254,254,254,254
270 DATA 0,0,0,254,254,254,254,254
280 DATA 0,0,254,254,254,254,254,254
290 DATA 0,254,254,254,254,254,254,254
300 DATA 254,254,254,254,254,254,254,254
310 DATA 255,129,129,129,129,129,255,0
320 GOSUB 660::INPUT"■5SHIFT CRSRUP■MAX. DATA VALUE";D$:GOSUB 670
330 IFE<>0ORLEN(D$)>5THEN 320
340 MD=VAL(D$):SC=(INT(MD/16)*16)+16:IFINT(MD/16)<>MD/16THEN 360
350 SC=SC-16
360 DI=100:C$=STR$(SC):SC$=RIGHT$(C$,LEN(C$)-1):PRINT"■HOME■■5CRSRDOWN■"TAB(6-LEN(SC$))SC$:DI=DI-25
370 S1=SC*DI/100:SC$=STR$(S1):PRINTTAB(6-LEN(SC$))"■3CRSRDOWN■";SC$
380 DI=DI-25:IFDI>=25THEN 370
390 GOSUB 660::INPUT"■3SHIFT CRSRUP■SIDE LABEL";A$:IFLEN(A$)>15ORLEN(A$)<1THEN 390
400 GOSUB 660::GOSUB 700
410 INPUT"■3SHIFT CRSRUP■BOT. LABEL";A$:GOSUB 660::IFLEN(A$)>15ORLEN(A$)<1THEN 410
420 GOSUB 700
430 PRINT"■3SHIFT CRSRUP■SELECT CLR: 2 FOR RED, 6 FOR BLU., 5 FOR GREEN"
440 GOSUB 990::A=A-48:IFA<>2ANDA<>5ANDA<>6THEN 440
450 CO=A:GOSUB 660
460 POKE8148,42:M=1
470 INPUT"■3SHIFT CRSRUP■ENTER DATA";D$:GOSUB 660:IFD$="END"THEN1010
480 IFD$="ERASE"THEN 760
490 IFD$="C"THEN 770
500 IFD$="L"THEN 800
510 IFD$="R"THEN 800
520 IFD$="P"THEN 840
530 IFD$="NEW"THEN 220
540 GOSUB 670::IFE<>0THEN 470
550 M1=8147+M:D1=VAL(D$):IFD1>SCTHENEND1=SC
560 DI=SC/16:DF=INT((D1/DI)):DP=D1-(DF*DI):DP=INT((DP/DI)*8)+1
570 IFDF=0THENX=1:GOTO 590
580 FORX=1TODF:POKEM1-(X*22),40:POKECL+M1-(X*22),CO:NEXT
590 POKEM1-(X*22),32+DP:POKECL+M1-(X*22),CO:IFDF=16THENDF=15
600 FORX=DF+2TO17:POKEM1-(X*22),33:POKECL+M1-(X*22),0:NEXT
610 IFM>=15THEN 470
620 POKE8147+M,41:POKE8147+M+1,42:M=M+1:GOTO 470
630 PRINT"■SHIFT CLR■■4CRSRDOWN■■CTRL BLK■":FORI=1TO17
640 PRINTTAB(6)"!!!!!!!!!!!!!!!!!":NEXT
650 PRINTTAB(6)")))))))))))))))))":RETURN
660 PRINT"■HOME■
    ":RETURN
670 FORI=1TOLEN(D$)
680 IFASC(MID$(D$,I,1))<48ORASC(MID$(D$,I,1))>57THENE=1:RETURN
690 NEXT:E=0:RETURN
700 FORC=1TOLEN(A$)
710 C$=MID$(A$,C,1):F1=ASC(C$):IFF1<65THEN 730
720 F1=F1-64
730 IFCT=0THENPOKE(7768+22*C),F1:POKECL+7768+(22*C),0:GOTO 750
740 POKE8169+C,F1:POKECL+8169+C,0
750 NEXT:CT=1:RETURN
760 M1=8147+M:FORI=1TO17:POKEM1-(I*22),33:POKECL+M1-(I*22),0:NEXT:GOTO 470
770 PRINT"■3SHIFT CRSRUP■SELECT CLR: 2 FOR RED, 6 FOR BLU., 5 FOR GREEN"
780 GOSUB 990::A=A-48:IFA<>2ANDA<>5ANDA<>6THEN 780
790 CO=A:GOSUB 660::GOTO 470
800 IF(M=1ANDD$="L")+(M=15ANDD$="R")THEN 470
810 POKE8147+M,41:IFD$="L"THENM=M-1:GOTO 830
820 M=M+1
830 POKE8147+M,42:GOTO 470
840 X=7768:OPEN4,4
850 FORN=1TO18:PRINT#4,CHR$(13);
860 FORI=0TO21:A=PEEK(X+I)
870 IFA=33THENA=164
880 IFA=34THENA=175
890 IFA=35THENA=185
900 IFA=36THENA=162
910 IFA=37THENA=184:PRINT#4,CHR$(18);
920 IFA=38THENA=183:PRINT#4,CHR$(18);
930 IFA=39THENA=163:PRINT#4,CHR$(18);
940 IFA=40THENA=166
950 IFA=41THENA=32
960 IFA<32THENA=A+64
970 PRINT#4,CHR$(A)+CHR$(146);CHR$(157);:NEXT:X=X+22:NEXT
980 PRINT#4:CLOSE4:GOTO 470
990 GETA$:IFA$=""THEN 990
1000 A=ASC(A$):RETURN
1010 END
```

HCM

THE BOOLEAN BRAIN

APPLE II Family

```
100 REM  **********************
110 REM  *  THE BOOLEAN BRAIN  *
120 REM  **********************
130 REM  BY W.K. BALTHROP
140 REM  AND THE HCM STAFF
150 REM  VERSION 4.3.1
160 REM  APPLE II SERIES APPLESOFT
170 REM
180 LOMEM: 24576
190 DIM CO(1,14),CA(14),LO(2,9),RM(10,4),L(3)
200 W1=7:W0=3:PB=2:BP=6:OG=5:GO=1:B1=4:B0=0:L(0)=42:L(1)=85:L(2)=170:L(3)=213
210 FOR I=1 TO 10: READ N$(I),RM(I,1),RM(I,2),RM(I,3),RM(I,4): NEXT
220 FOR I=0 TO 14: READ CO(0,I),CO(1,I): NEXT I: FOR I=0 TO 14: READ CO(I): NEXT I: FOR I=1 TO 9: READ LO(2,I): NEXT I
230 POKE 232,0: POKE 233,3: FOR PA=768 TO 932: READ PV: POKE PA,PV: NEXT PA
240 FOR I=1 TO 10: READ NL(I): NEXT I
250 POKE 34,20: HOME : GOSUB 1120: FOR I=1 TO 3: HCOLOR=OG: ON I GOSUB 1270,1280,1290: NEXT I
260 DI=1:R=1:SC=0
270 POKE 230,32: POKE 49236,0: POKE 49235,0: HOME : PRINT "DIRECTION: ": VTAB 21: HTAB 12: PRINT "LOCATION:": VTAB 21: HTAB 32: ON DI GOSUB 930,940,950,960
280 PRINT CHR$(7);"WHICH WAY? (N,S,E,W)";: K=PEEK(-16368): POKE -16384,0
290 GET K$: IF NOT ((K$="E") OR (K$="W") OR (K$="S") OR (K$="N")) THEN 290
300 DI=(K$="E")+(2*(K$="W"))+(3*(K$="N"))+(4*(K$="S"))
310 IF RM(R,DI)=R THEN GOTO 1040
320 IF RM(R,DI)>10 THEN R=(RM(R,DI)-10): GOTO 270
330 ND=ND+1: GOSUB 1360: FOR I=1 TO 10: GT(I)=0: NEXT I: K=PEEK(-16368): POKE -16384,0
340 GET K$: IF K$<"0" OR K$>"9" THEN 350
360 VAL(K$): IF G=0 THEN G=10
370 IF GT(G)=1 THEN 340
380 SC=SC+1: HCOLOR=GO: I=G: GOSUB 1110,480,450,480,540,570,600,630,660
390 GT(1)=1: IF GT(2)=1 THEN 690
400 IF T(1)=2 THEN 690
410 GOTO 350
420 GT(2)=1: IF GT(1)=1 THEN 690
430 IF T(1)=2 THEN 690
440 GOTO 350
450 GT(3)=1: IF GT(4)=1 THEN 710
460 IF T(2)=2 THEN 710
470 GOTO 350
480 GT(4)=1: IF GT(3)=1 THEN 710
490 IF T(2)=2 THEN 710
500 GOTO 350
510 GT(5)=1: IF GT(6)=1 THEN 730
520 IF T(3)=2 THEN 730
530 GOTO 350
540 GT(6)=1: IF GT(5)=1 THEN 730
550 IF T(3)=2 THEN 730
560 GOTO 350
570 GT(7)=1: IF GT(8)=1 THEN 750
580 IF T(4)=2 THEN 750
590 GOTO 350
600 GT(8)=1: IF GT(7)=1 THEN 750
610 IF T(4)=2 THEN 750
620 GOTO 350
630 GT(9)=1: IF GT(10)=1 THEN 770
640 IF T(5)=2 THEN 770
650 GOTO 350
660 GT(10)=1: IF GT(9)=1 THEN 770
670 IF T(5)=2 THEN 770
680 GOTO 350
690 I=1: ON T(1) GOSUB 1430,1440: GOSUB 1450:T(1)=0: IF T(6)=2 OR T(2)=0 THEN 790
700 GOTO 350
710 I=2: ON T(2) GOSUB 1430,1440: GOSUB 1450:T(2)=0: IF T(6)=2 OR T(1)=0 THEN 790
720 GOTO 350
730 I=3: ON T(3) GOSUB 1430,1440: GOSUB 1450:T(3)=0: IF T(7)=2 OR T(4)=0 THEN 810
740 GOTO 350
750 I=4: ON T(4) GOSUB 1430,1440: GOSUB 1450:T(4)=0: IF T(7)=2 OR T(3)=0 THEN 810
760 GOTO 350
770 I=5: ON T(5) GOSUB 1430,1440: GOSUB 1450:T(5)=0: IF T(8)=2 OR T(7)=0 THEN 830
780 GOTO 350
790 I=6: ON T(6) GOSUB 1430,1440: GOSUB 1450:T(6)=0: IF T(9)=2 OR T(8)=0 THEN 850
800 GOTO 350
810 I=7: ON T(7) GOSUB 1430,1440: GOSUB 1450:T(7)=0: IF T(8)=2 OR T(5)=0 THEN 830
820 GOTO 350
830 I=8: ON T(8) GOSUB 1430,1440: GOSUB 1450:T(8)=0: IF T(9)=2 OR T(6)=0 THEN 850
840 GOTO 350
850 I=9: ON T(9) GOSUB 1430,1440: GOSUB 1450: FOR DE=1 TO 1000: NEXT DE:OM=R:R=RM(R,DI): IF DI=1 OR DI=3 THEN RM(RM(OM,DI),DI+1)=RM(RM(RM(OM,DI),DI+1)+10: GOTO 870
860 RM(RM(OM,DI),DI-1)=RM(RM(OM,DI),DI-1)+10
870 RM(OM,DI)=RM(OM,DI)+10: IF R<>10 THEN 270
880 POKE 230,32: POKE 49236,0: POKE 49235,0: HOME : GOSUB 1180: PRINT "CONGRATULATIONS -- YOU HAVE REACHED THE CENTRAL PROCESSING UNIT"
890 FOR H=1 TO 2: FOR I=0 TO 7068 STEP 1024: FOR J=8244 TO 8638 STEP 128: FOR K=0 TO 15: POKE I+J+K,L(INT(RND(1)*3)):M=PEEK(-16336): NEXT : PRINT CHR$(7);: NEXT : NEXT : NEXT
900 SC=INT(ND/SC/ND*10000): IF SC<600 THEN PRINT "HOWEVER, THE COMPUTER IS DAMAGED DUE TO DATA LOSS": GOTO 920
910 PRINT "YOU REPAIR THE COMPUTER AND ESCAPE": PRINT "YOUR SCORE=";SC;" PLAY AGAIN? (Y/N)";: GOTO 1080
920 PRINT "EAST": HTAB 11: PRINT N$(R): CT=2:LT=4:RT=3: GOSUB 970: RETURN
930 PRINT "WEST": HTAB 11: PRINT N$(R): CT=1:LT=3:RT=4: GOSUB 970: RETURN
940 PRINT "NORTH": HTAB 11: PRINT N$(R): CT=3:LT=2:RT=1: GOSUB 970: RETURN
950 PRINT "SOUTH": HTAB 11: PRINT N$(R): CT=4:LT=1:RT=2: GOSUB 970: RETURN
960 IF RM(R,CT)<11 THEN HCOLOR=OG:
970 GOTO 990
980 HCOLOR=B1
990 GOSUB 1280: IF RM(R,LT)<11 THEN HCOLOR=OG: GOTO 1010
1000 HCOLOR=B1
1010 GOSUB 1270: IF RM(R,RT)<11 THEN HCOLOR=OG: GOTO 1030
1020 HCOLOR=B1
1030 GOSUB 1290: RETURN
1040 HOME : POKE 49235,0: PRINT "YOU ARE ZAPPED BY A BAD DISK SECTOR"
1050 HGR2 :W1=W0:OG=GO:BP=PB: GOSUB 1130: FOR I=1 TO 200: POKE 49234,0: POKE 49237,0: POKE 49236,0: POKE 49235,0:K=PEEK(-16336)
1060 NEXT I
1070 PRINT "THERE IS NO ESCAPE FROM. . . 'CRASH'": PRINT "WANT TO PLAY AGAIN (Y/N)";
1080 GET K$: IF K$<>"Y" AND K$<>"N" THEN 1080
1090 IF K$="Y" THEN RUN
1100 TEXT : HOME : END
1110 HPLOT 14,NL(I)+3 TO 27,NL(I)+3: RETURN
1120 HGR : HCOLOR=W1:AN=25/70
1130 HPLOT 0,20 TO 70,45 TO 210,45 TO 279,20: HPLOT 0,140 TO 70,115 TO 210,115 TO 279,140
1140 HCOLOR=W1:CS=70: FOR RW=116 TO 140: HPLOT CS-(RW-115)/AN,RW TO (CS+140)+(RW-115)/AN,RW: NEXT RW
1150 HCOLOR=OG:CS=72:RW=20: FOR I=25 TO 1 STEP -1: HPLOT CS-I/AN,RW TO (CS+137)+I/AN,RW:RW=RW+1: NEXT I
1160 HCOLOR=W1: FOR CL=1 TO 2: HPLOT 70+CL,44 TO 70+CL,114: HPLOT 210+CL,44 TO 210+CL,114: NEXT CL
1170 GOSUB 1180: GOSUB 1250: RETURN
1180 HCOLOR=BP: FOR RW=21 TO 44: HPLOT 0,RW TO (RW-20)/AN,RW: NEXT RW
1190 FOR RW=45 TO 115: HPLOT 0,RW TO 70,RW: NEXT RW
1200 FOR RW=116 TO 139: HPLOT 0,RW TO (140-RW)/AN,RW: NEXT RW
1210 FOR RW=46 TO 114: HPLOT 72,RW TO 210,RW: NEXT RW
```

Continued

PROGRAM LISTING

THE BOOLEAN BRAIN

APPLE II Family

Continued

```
1220 RW = 21: FOR I = 44 TO 21 STEP -1
     : HPLOT 211 + ((I - 20) / AN),RW TO
     279,RW:RW = RW + 1: NEXT I
1230 FOR RW = 45 TO 115: HPLOT 212,RW TO
     279,RW: NEXT RW
1240 FOR RW = 116 TO 139: HPLOT 211 + ((
     RW - 115) / AN),RW TO 279,RW: NEXT
     RW: RETURN
1250 HCOLOR= W0: FOR RW = 75 TO 88: HPLO
     T 168,RW TO 181,RW: NEXT RW: HCOLOR
     = PB: FOR RW = 78 TO 85: HPLOT 171,
     RW TO 179,RW: NEXT RW
1260 HCOLOR= W0: FOR RW = 80 TO 93: HPLO
     T 49,RW TO 62,RW: NEXT RW: HCOLOR=
     PB: FOR RW = 83 TO 90: HPLOT 51,RW
     TO 59,RW: NEXT RW: RETURN
1270 GOSUB 1320: HCOLOR= W1: HPLOT 11,50
     TO 41,61: RETURN
1280 GOSUB 1310: HCOLOR= W1: HPLOT 126,6
     2 TO 160,62: RETURN
1290 GOSUB 1340: HCOLOR= W1: HPLOT 238,6
     1 TO 270,50: RETURN
1300 END
1310 FOR RW = 63 TO 114: HPLOT 127,RW TO
     159,RW: NEXT RW: RETURN
1320 FOR RW = 50 TO 61: HPLOT 11,RW TO 1
     1 + ((RW - 50) / AN),RW: NEXT RW: F
     OR RW = 61 TO 124: HPLOT 11,RW TO 4
     1,RW: NEXT RW
1330 FOR RW = 125 TO 135: HPLOT 11,RW TO
     11 + ((136 - RW)) / AN,RW: NEXT RW
     : RETURN
1340 RW = 50: FOR I = 61 TO 50 STEP -1
     : HPLOT 238 + ((I - 49) / AN),RW TO
     269,RW:RW = RW + 1: NEXT I: FOR RW
     = 62 TO 124: HPLOT 239,RW TO 269,R
     W: NEXT RW
1350 FOR RW = 125 TO 135: HPLOT 239 + ((
     RW - 124) / AN),RW TO 269,RW: NEXT
     RW: RETURN
1360 FOR I = 1 TO 9:T(I) = INT (( RND (
     1) * 2) + 1): NEXT I
1370 HGR2 : SCALE= 1: ROT= 0
1380 FOR I = 1 TO 10: HCOLOR= W1: DRAW I
     AT 3,NL(I)
1390 HCOLOR= OG: GOSUB 1110: NEXT I
1400 FOR I = 1 TO 9: ON T(I) GOSUB 1430,
     1440: GOSUB 1450: NEXT I
1410 RETURN
1420 END
1430 FOR J = 0 TO 14:RW = LO(2,I) + J: H
     PLOT LO(1,I),RW TO LO(1,I) + CA(J),
     RW: NEXT J: RETURN
1440 FOR J = 0 TO 14:RW = LO(2,I) + J: H
     PLOT LO(1,I) + CO(0,J),RW TO LO(1,I
     ) + CO(1,J),RW: NEXT J: RETURN
```

```
1450 ON I GOSUB 1460,1470,1480,1490,1500
     ,1510,1520,1530,1540: RETURN
1460 HPLOT 49,23 TO 69,23 TO 69,39 TO 90
     ,39: RETURN
1470 HPLOT 49,57 TO 69,57 TO 69,47 TO 90
     ,47: RETURN
1480 HPLOT 49,95 TO 69,95 TO 69,107 TO 9
     0,107: RETURN
1490 HPLOT 49,127 TO 69,127 TO 69,115 TO
     90,115: RETURN
1500 HPLOT 49,165 TO 133,165 TO 133,141
     TO 153,141: RETURN
1510 HPLOT 112,43 TO 195,43 TO 195,91 TO
     216,91: RETURN
1520 HPLOT 112,111 TO 133,111 TO 133,133
     TO 153,133: RETURN
1530 HPLOT 175,137 TO 195,137 TO 195,99
     TO 216,99: RETURN
1540 HPLOT 238,95 TO 251,95: FOR I = 88
     TO 102: HPLOT 249,I TO 261,I: NEXT
     I: RETURN
1550 DATA KEYBOARD INTERFACE,5,4,8,7,IN
     PUT PORT,4,3,6,9,VIDEO PROCESSING,2
     ,5,7,4,SOUND CONTROL ROOM,1,2,3,5,R
     AM ROOM,3,1,4,8,DISK CONTROLLER S R
     OOM,7,8,9,2,DISK DRIVE,7,6,1,3,ROM
     ROOM,6,9,5,1,PORT CONTROL,8,10,2,6
1560 DATA CENTRAL PROCESSING CONTROL,9
     ,10,10,10
1570 DATA 0,13,1,15,2,15,0,17,4,17,4,1
     9,4,19,4,20,4,19,4,19,4,17,0,17,2,1
     5,1,15,0,13
1580 DATA 15,16,17,18,19,19,20,20,20,19
     ,19,18,17,16,15
1590 DATA 28,16,28,50,28,88,28,120,28,
     158,91,36,91,104,154,130,217,88
1600 DATA 10,0,22,0,30,0,41,0,52,0,63,0
     ,76,0,88,0,124,0,137,0,150,0,73,62,
     14,54,190,45,5,0,17,12,45,21,190,19
     1,23,46,45,45,0,41,45,53,23,23,173,
     246,63,28,7,0,73,49,191,23,46,181,3
     3,44,28,36,0,41,45,245,219,46,45,21
     ,54,23,63,28,4,0
1610 DATA 73,45,222,243,46,45,21,190,63,
     28,36,0,41,45,53,23,23,54,6,0,27
     ,63,23,182,190,63,28,44,40,229,27,3
     6,0,9,45,21,254,3,55,14,45,53,23,23
     ,63,7,0,9,45,21,182,190,63,28,44,40
     ,229,27,36,0,9,45,21,254,195,118,45
     ,53,23,23,63,7,0
1620 DATA 9,45,21,54,54,30,63,28,36,36,7
     7,186,23,6,0
1630 DATA 16,24,50,58,88,96,120,128,158,
     166
```

HCM

THE BOOLEAN BRAIN

IBM PCjr

```
100 REM ********************
110 REM * THE BOOLEAN BRAIN *
120 REM ********************
130 REM BY W. K. BALTHROP
140 REM HOME COMPUTER MAGAZINE
150 REM VERRSION 4.3.1
160 REM IBM PCjr
170 REM CARTRIDGE BASIC
180 REM
190 CLS:CLEAR ,,,32768!:SCREEN 5:DIM A(
    2,10),RM(10,4):RANDOMIZE TIMER:KEY
    OFF
200 FOR Z=1 TO 9:READ A(1,Z),A(2,Z):NEX
    T:FOR Z=1 TO 10:READ N$(Z),RM(Z,1),
    RM(Z,2),RM(Z,3),RM(Z,4):NEXT
210 C=0:GOSUB 900:GOSUB 910
220 DIR=1:SC=0:R=1
230 ON DIR GOSUB 260,300,340,380:GOSUB
    990:DEF SEG=0:POKE 1050,PEEK(1052)
240 A$=INKEY$:IF A$="" THEN 240 ELSE IF
    A$="E" THEN DIR=1 ELSE IF A$="W" T
    HEN DIR=2 ELSE IF A$="N" THEN DIR=3
    ELSE IF A$="S" THEN DIR=4 ELSE GOT
    O 240
250 IF RM(R,DIR)=R THEN 700 ELSE IF RM(
    R,DIR)>10 THEN R=RM(R,DIR)-10:GOSUB
    910:GOTO 230 ELSE GOTO 840
260 LOCATE 7,20:PRINT "E";:IF RM(R,1)<1
    1 THEN GOSUB 940
270 IF RM(R,3)<11 THEN GOSUB 950
280 IF RM(R,4)<11 THEN GOSUB 960
290 RETURN
300 LOCATE 7,20:PRINT "W";:IF RM(R,2)<1
    1 THEN GOSUB 940
310 IF RM(R,4)<11 THEN GOSUB 950
320 IF RM(R,3)<11 THEN GOSUB 960
330 RETURN
340 LOCATE 7,20:PRINT "N";:IF RM(R,3)<1
    1 THEN GOSUB 940
350 IF RM(R,2)<11 THEN GOSUB 950
360 IF RM(R,1)<11 THEN GOSUB 960
370 RETURN
380 LOCATE 7,20:PRINT "S";:IF RM(R,4)<1
    1 THEN GOSUB 940
390 IF RM(R,1)<11 THEN GOSUB 950
```

```
400 IF RM(R,2)<11 THEN GOSUB 960
410 RETURN
420 G=VAL(A$):IF G=0 THEN G=10
430 IF GT(G)=1 THEN D=0:RETURN
440 IF RM(R,DIR)>10 THEN R=RM(R,DIR):D=
    1:RETURN
450 ON G GOTO 460,470,480,490,500,510,5
    20,530,540,550
460 SC=SC+1:LINE (10,16)-(24,16),2:GT(1
    )=1:IF T(1)=2 THEN 560 ELSE IF T(1)
    =1 AND GT(2)=1 THEN 560 ELSE D=0:RE
    TURN
470 SC=SC+1:LINE (10,24)-(24,24),2:GT(2
    )=1:IF T(1)=2 THEN 560 ELSE IF T(1)
    =1 AND GT(1)=1 THEN 560 ELSE D=0:RE
    TURN
480 SC=SC+1:LINE (10,56)-(24,56),2:GT(3
    )=1:IF T(2)=2 THEN 570 ELSE IF T(2)
    =1 AND GT(4)=1 THEN 570 ELSE D=0:RE
    TURN
490 SC=SC+1:LINE (10,64)-(24,64),2:GT(4
    )=1:IF T(2)=2 THEN 570 ELSE IF T(2)
    =1 AND GT(3)=1 THEN 570 ELSE D=0:RE
    TURN
500 SC=SC+1:LINE (10,96)-(24,96),2:GT(5
    )=1:IF T(3)=2 THEN 580 ELSE IF T(3)
    =1 AND GT(6)=1 THEN 580 ELSE D=0:RE
    TURN
510 SC=SC+1:LINE (10,104)-(24,104),2:GT
    (6)=1:IF T(3)=2 THEN 580 ELSE IF T(
    3)=1 AND GT(5)=1 THEN 580 ELSE D=0:
    RETURN
520 SC=SC+1:LINE (10,136)-(24,136),2:GT(
    7)=1:IF T(4)=2 THEN 590 ELSE IF T(
    4)=1 AND GT(8)=1 THEN 590 ELSE D=0:
    RETURN
530 SC=SC+1:LINE (10,144)-(24,144),2:GT
    (8)=1:IF T(4)=2 THEN 590 ELSE IF T(
    4)=1 AND GT(7)=1 THEN 590 ELSE D=0:
    RETURN
540 SC=SC+1:LINE (10,176)-(24,176),2:GT(
    9)=1:IF T(5)=2 THEN 600 ELSE IF T(
    5)=1 AND GT(10)=1 THEN 600 ELSE D=0
    :RETURN
```

Continued

IBM PCjr

```
550 SC=SC+1:LINE (10,184)-(24,184),2:GT
    (10)=1:IF T(5)=2 THEN 600 ELSE IF T
    (5)=1 AND GT(9)=1 THEN 600 ELSE D=0
    :RETURN
560 Y=A(1,1):Z=A(2,1):COL=2:ON T(1) GOS
    UB 970,980:T(1)=0:GOSUB 730:IF T(6)
    =2 OR T(6)=1 AND T(2)=0 THEN GOTO 6
    10 ELSE D=0:RETURN
570 Y=A(1,2):Z=A(2,2):COL=2:ON T(2) GOS
    UB 970,980:T(2)=0:GOSUB 740:IF T(6)
    =2 OR T(6)=1 AND T(1)=0 THEN GOTO 6
    10 ELSE D=0:RETURN
580 Y=A(1,3):Z=A(2,3):COL=2:ON T(3) GOS
    UB 970,980:T(3)=0:GOSUB 750:IF T(7)
    =2 OR T(7)=1 AND T(4)=0 THEN GOTO 6
    20 ELSE D=0:RETURN
590 Y=A(1,4):Z=A(2,4):COL=2:ON T(4) GOS
    UB 970,980:T(4)=0:GOSUB 760:IF T(7)
    =2 OR T(7)=1 AND T(3)=0 THEN GOTO 6
    20 ELSE D=0:RETURN
600 Y=A(1,5):Z=A(2,5):COL=2:ON T(5) GOS
    UB 970,980:T(5)=0:GOSUB 770:IF T(8)
    =2 OR T(8)=1 AND T(7)=0 THEN GOTO 6
    30 ELSE D=0:RETURN
610 Y=A(1,6):Z=A(2,6):COL=2:ON T(6) GOS
    UB 970,980:T(6)=0:GOSUB 780:IF T(9)
    =2 OR T(9)=1 AND T(8)=0 THEN GOTO 6
    40 ELSE D=0:RETURN
620 Y=A(1,7):Z=A(2,7):COL=2:ON T(7) GOS
    UB 970,980:T(7)=0:GOSUB 790:IF T(8)
    =2 OR T(8)=1 AND T(5)=0 THEN GOTO 6
    30 ELSE D=0:RETURN
630 Y=A(1,8):Z=A(2,8):COL=2:ON T(8) GOS
    UB 970,980:T(8)=0:GOSUB 800:IF T(9)
    =2 OR T(9)=1 AND T(6)=0 THEN GOTO 6
    40 ELSE D=0:RETURN
640 Y=A(1,9):Z=A(2,9):COL=2:ON T(9) GOS
    UB 970,980:T(9)=0:GOSUB 810:FOR TD=
    1 TO 2500:NEXT:D=1:ORM=R:R=RM(R,DIR
    ):IF DIR=1 OR DIR=3 THEN RM(RM(ORM,
    DIR),DIR+1)=RM(RM(ORM,DIR),DIR+1)+1
    0 ELSE RM(RM(ORM,DIR),DIR-1)=RM(RM(
    ORM,DIR),DIR-1)+10
650 RM(ORM,DIR)=RM(ORM,DIR)+10:RETURN
660 LINE(95,55)-(223,95),0,BF:FOR Z=1 T
    O 200:X=INT(RND*125)+95:Y=INT(RND*3
    7)+55:LINE (X,Y)-(X+3,Y+3),INT(RND*
    5)+5,BF:NEXT
670 FOR Z=1 TO 200:PALETTE INT(RND*5)+5
    ,INT(RND*16):SOUND RND*10000+110,1,
    15:FOR TD=1 TO 50:NEXT:NEXT:SCORE=I
    NT(ND/SC/ND*10000)
680 LINE (0,155)-(319,199),0,BF:LOCATE
    20,1:PRINT "CONGRATULATIONS -- YOU
    HAVE FOUND THE CENTRAL PROCCESSIN
    G UNIT.":IF SCORE<600 THEN PRINT "T
    HE COMPUTER IS DAMMAGED. YOU LOOSE
     BECAUSE OF DATA LOSS." ELSE PRI
    NT "YOU REPAIR THE COMPUTER AND ESC
    APE."
690 FOR TD=1 TO 3000:NEXT:PRINT "YOUR S
    CORE=";SCORE;" PLAY AGAIN (Y/N)?";:
    GOTO 720
700 FOR Z=1 TO 15:FOR Y=1 TO 15:PALETTE
     Y,4:SOUND Y*100+1000,2,15:NEXT:PAL
    ETTE:NEXT
710 CLS:LOCATE 16,1:PRINT "YOU'VE BEEN
    ZAPPED BY A BAD DISK SECTOR.":PRINT
    :PRINT "THERE IS NO ESCAPE FROM....
    .....'CRASH'":PRINT:PRINT "PLAY AGAI
    N (Y/N)?"
720 A$=INKEY$:IF A$="" THEN 720 ELSE IF
     A$="Y" THEN RUN ELSE IF A$<>"N" TH
    EN 720 ELSE END
730 DRAW "C=COL;BM56,20R20D16R17":RETUR
    N
740 DRAW "C=COL;BM56,60R20U16R17":RETUR
    N
750 DRAW "C=COL;BM56,100R20D16R17":RETU
    RN
760 DRAW "C=COL;BM56,140R20U16R17":RETU
    RN
770 DRAW "C=COL;BM56,180R90U26R17":RETU
    RN
780 DRAW "C=COL;BM126,40R90D51R17":RETU
    RN
790 DRAW "C=COL;BM126,120R20D26R17":RET
    URN
800 DRAW "C=COL;BM196,150R20U51R17":RET
    URN
810 LINE (266,85)-(286,105),COL,BF:RETU
    RN
820 L=0:FOR Z=2 TO 22 STEP 5:L=L+1:IF L
    =9 THEN LOCATE Z,1:PRINT "9";:LOCAT
    E Z+2,1:PRINT "0";:ELSE LOCATE Z,1:
    PRINT RIGHT$(STR$(L),1);:LOCATE Z+2
    ,1:L=L+1:PRINT RIGHT$(STR$(L),1);
830 NEXT:RETURN
840 ND=ND+1:FOR Z=1 TO 10:GT(Z)=0:NEXT:
    CLS:DRAW "S8":COL=4:FOR B=1 TO 9:T(
    B)=INT(RND*2)+1:Y=A(1,B):Z=A(2,B):O
    N T(B) GOSUB 970,980:NEXT:DRAW "S4"
850 FOR Z=16 TO 176 STEP 40:LINE (10,Z)
    -(24,Z),4:LINE (10,Z+8)-(24,Z+8),4:
    NEXT
860 COL=4:DRAW "S4":GOSUB 730:GOSUB 740
    :GOSUB 750:GOSUB 760:GOSUB 770:GOSU
    B 780:GOSUB 790:GOSUB 800:GOSUB 810
    :GOSUB 820:COL=0
870 COL=COL+1:IF COL=6 THEN COL=4
880 A$=INKEY$:IF A$>="0" AND A$<="9" TH
    EN GOSUB 420:IF D=1 AND R<>10 THEN I
    GOSUB 900:GOSUB 910:GOTO 230 ELSE I
    F D=1 THEN GOSUB 900:GOSUB 990:GOTO
     660
890 PALETTE 4,COL:GOTO 870
900 COLOR 15,0:CLS:PALETTE:DRAW "C1BM0,
    23M79,47M239,47M319,23BM319,143M239
    ,119NU72M79,119NU72M0,143BM47,87P10
    ,1BM159,87P11,1BM287,87P10,1BM159,2
    3P3,1BM159,160P2,1":RETURN
910 DRAW "C1BM143,119U56R32D56BH5P=C;,1
    BM183,79C1R16D16L16U16BF2D5R12U5L12
    BD9D3R12U3L12BFP4,1BU6P9,1BD3P15,1"
920 DRAW "C1BM7,140M7,51M47,59M47,129BH
    3P=C;,1C1BM71,79L16D16R16U16BG2D5L1
    2U5R12BD9D3L12U3R12BGP4,1BU2P15,1BU
    4P9,1"
930 DRAW "C1BM271,129M271,59M311,51M311
    ,140BM287,119P=C;,1":RETURN
940 PAINT (159,65),4,1:RETURN
950 PAINT (8,52),4,1:RETURN
960 PAINT (310,52),4,1:RETURN
970 DRAW "C=COL;BM=Y;,=Z;S8U2H2LHL9D10R
    9ERE2U2NR6BL2S4P=COL;,=COL;":RETURN
980 DRAW "C=COL;BM=Y;,=Z;S8H4LHL9FRF2NL
    4RFD2GLNL4G2LGR9ERE4R4BL6S4P=COL;,=
    COL;":RETURN
990 LOCATE 25,3:PRINT LEFT$("
    ",(36-LEN(N$(R)))/2);N$(R)
    ;RIGHT$("
    ",(36-LEN(N$(R)))/2);:RETURN
1000 DATA 50,20,50,60,50,100,50,140,50,1
    80,120,40,120,120,190,150,260,95
1010 DATA KEYBOARD INTERFACE,5,4,8,7,INP
    UT PORT,4,3,6,9,VIDEO PROCCESSING,2
    ,5,7,4,SOUND CONTROL ROOM,1,2,3,5,R
    AM ROOM,3,1,4,8,DISK CONTROLERS ROO
    M,7,8,9,2,DISK DRIVE,7,6,1,3,ROM ROO
    M,6,9,5,1,PORT CONTROL,8,10,2,6,CE
    NTRAL PROCCESSING CONTROL,9,10,10,1
    0
```

HCM

Continued

VIC-20

```
100 REM  *********
110 REM  *  COLOR FUN  *
120 REM  *********
130 REM BY TED MARTINO
140 REM HOME COMPUTER MAGAZINE
150 REM VERSION 4.3.1
160 REM VIC-20 BASIC
170 REM
180 POKE36879,25:POKE788,194:PRINT"■SHI
    FT CLR■":GOTO 220
190 X=X+1:Y=Y+1:IFY>7THENY=0
200 IFY=1THENY=Y+1
210 RETURN
220 POKE7680+X,102:POKE38400+X,Y:GOSUB1
    90:IFX>43THEN 240
230 GOTO 220
240 X=0
250 POKE8142+X,102:POKE38862+X,Y:GOSUB
    190:IFX>43THENGOTO 270
260 GOTO 250
270 PRINT"■HOME■7 CRSRDOWN■■CTRL RVSON■
    ■CTRL RED■
280 PRINT"■CTRL RVSON■■CTRL RED■ CO
    LOR-FUN
290 PRINT"■CTRL RVSON■■CTRL RED■
300 PRINT" PRESS ANY KEY TO BEGIN"
310 GET K$:IF K$="" THEN 310
320 PRINT"■SHIFT CLR■■CTRL BLK■■8 CRSRD
    OWN■LEVEL 1 OR 2 ?"
330 GETB$:IFB$=""THEN 330
340 V=VAL(B$):IFB$<"1"ORB$>"2" THEN 330
350 INPUT"■SHIFT CLR■■8 CRSRDOWN■■CTRL
    PURPLE■WHAT'S YOUR NAME";N$:PRINT"■SHI
    FT CLR■■CTRL BLK■■8 CRSRDOWN■HI THER
    E, "N$:X=RND(-TI)
360 FORT=1TO1000:NEXT:POKE36879,25:GOSU
    B 370:GOTO 650
370 D1=INT(RND(1)*5+1)
380 D2=INT(RND(1)*5+1):IFD2=D1THEN 380
390 D3=INT(RND(1)*5+1):IFD3=D2ORD3=D1TH
    EN 390
400 D4=INT(RND(1)*5+1):IFD4=D1ORD4=D2OR
    D4=D3THEN 400
410 D5=INT(RND(1)*5+1):IFD5=D4ORD5=D3OR
    D5=D2ORD5=D1THEN 410
420 ON-(D1=1)-2*(D1=2)-3*(D1=3)-4*(D1=4
    )-5*(D1=5)GOTO430,450,470,490,510
```

Continued

COLOR FUN

Continued

VIC-20

```
430 C1=0:IFB=1THENC1$="-BLACK ■CTRL RVS
    ON■CTRL BLK■2SHIFT Q■CTRL RVSOFF
    ■CTRL BLK■":GOTO 530
440 C1$="-BLACK":GOTO 530
450 C1=2:IFB=1THENC1$="-RED ■CTRL RVSON
    ■CTRL RED■2SHIFT S■CTRL RVSOFF■
    CTRL BLK■":GOTO 530
460 C1$="-RED":GOTO 530
470 C1=5:IFB=1THENC1$="-GREEN ■CTRL RVS
    ON■CTRL GRN■2SHIFT A■CTRL RVSOFF
    ■CTRL BLK■":GOTO 530
480 C1$="-GREEN":GOTO 530
490 C1=6:IFB=1THENC1$="-BLUE ■CTRL RVSO
    N■CTRL BLU■2SHIFT Z■CTRL RVSOFF■
    ■CTRL BLK■":GOTO 530
500 C1$="-BLUE":GOTO 530
510 C1=7:IFB=1THENC1$="-YELLOW■CTRL RVS
    ON■CTRL YEL■2SHIFT X■CTRL RVSOFF
    ■CTRL BLK■":GOTO 530
520 C1$="-YELLOW"
530 ON-(D2=1)-2*(D2=2)-3*(D2=3)-4*(D2=4
    )-5*(D2=5)GOTO540,560,580,600,620
540 C2=0:IFB=1THENC2$="-BLACK ■CTRL RVS
    ON■2SHIFT Q■CTRL RVSOFF■":GOTO 64
    0
550 C2$="-BLACK":GOTO 640
560 C2=2:IFB=1THENC2$="-RED ■CTRL RVSON
    ■CTRL RED■2SHIFT S■CTRL RVSOFF■■
    CTRL BLK■":GOTO 640
570 C2$="-RED":GOTO 640
580 C2=5:IFB=1THENC2$="-GREEN ■CTRL RVS
    ON■CTRL GRN■2SHIFT A■CTRL RVSOFF
    ■CTRL BLK■":GOTO 640
590 C2$="-GREEN":GOTO 640
600 C2=6:IFB=1THENC2$="-BLUE ■CTRL RVSO
    N■CTRL BLU■2SHIFT Z■CTRL RVSOFF■
    ■CTRL BLK■":GOTO 640
610 C2$="-BLUE":GOTO 640
620 C2=7:IFB=1THENC2$="-YELLOW■CTRL RVS
    ON■CTRL YEL■2SHIFT X■CTRL RVSOFF
    ■CTRL BLK■":GOTO 640
630 C2$="-YELLOW
640 RETURN
650 PRINT"■SHIFT CLR■CTRL RVSON■CTRL
    RED■ TYPE THE COLORS #":PRINT"FO
    R ■CTRL RVSON■NE■CTRL RVSOFF■EW PLAY
    ER/■CTRL RVSON■E■CTRL RVSOFF■ND"
660 PRINT"■CTRL BLK■HOME■4CRSRDOWN■WH
    AT'S THIS COLOR":POKE7786,160:POKE7
    787,160:POKE7808,160:POKE7809,160
670 E=RND(1):IFE<=.5THENF=C1:G=1:GOTO69
    0
680 F=C2:G=2
690 POKE38506,F:POKE38507,F:POKE38528,F
    :POKE38529,F
700 PRINT"■11CRSRDOWN■1"C1$,"2"C2$
710 GETZ$:IFZ$<>""THEN 710
720 GETZ$:IF NOT(Z$="1" OR Z$="2" OR Z$
    ="E" OR Z$="N")THEN 720
730 Q=Q+1:Z=VAL(Z$):ON-(Z=G)-2*(Z$="N")
    -3*(Z$="E")GOTO 770, 880, 890
740 PRINT"■HOME■7CRSRDOWN■■CTRL RVSON■
    ■CTRL PUR■ WRONG "N$" ":PRINT:PRINT
    "■CTRL RVSON■TRY AGAIN■CTRL BLK■"
750 FOR DE=1 TO 500:NEXT DE:IFV=1THENB=
    1:W=0:GOTO 710
760 GOTO 710
770 K=26:POKE36879,59:READA$:PRINT"■SHI
    FT CLR■9CRSRDOWN■"A$" "N$:POKE3687
    8,15:FORL=1TO4
780 FORM=254TO240STEP-1
790 POKE36876,M:POKE36879,K
800 NEXTM
810 POKE36876,0
820 FORM=0TOINT(RND(1)*100)+120
830 NEXTM
840 K=K+1:NEXTL
850 P=P+1:IFP=11THENRESTORE:P=0:GOTO 86
    0
860 I=I+1:W=W+1:IFW=10THENB=2:GOTO 360
870 GOTO 360
880 Q=Q-1:GOSUB 900:I=0:Q=0:W=0:GOTO32
    0
890 Q=Q-1:GOSUB 900:PRINT"■SHIFT CLR■"
    :POKE36879,27:PRINT"■CTRL BLU■":POK
    E788,191:END
900 PRINT"■SHIFT CLR■3CRSRDOWN■WELL "N
    $:PRINT"■CRSRDOWN■HERE'S YOUR SCORE
    ":PRINT"■2CRSRDOWN■YOU GOT "I" CORR
    ECT"
910 PRINT"■CRSRDOWN■OUT OF"Q
920 PRINT"■5CRSRDOWN■■CTRL RVSON■■CTRL
    RED■ HIT ANY KEY ■CTRL RVSOFF■
    "
930 GETE$:IFE$=""THEN 930
940 RETURN
950 DATAGOOD,RIGHT,ALL RIGHT,GOOD JOB,F
    ABULOUS,SUPER,RIGHT AGAIN,TERRIFIC,
    RIGHT ON
960 DATAOK,GOOD WORK
```

HCM

CYBER-CIPHER

TI-99/4A

```
100 REM  **********
110 REM  * CYBER-CIPHER *
120 REM  **********
130 REM  BY BUD DAVIS
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.3.1
160 REM  TI BASIC
170 REM
180 L=10
190 A$="0000C3C3C3C3C"
200 B$="1818181818181818"
210 C$="1824428181422418"
220 D$="183C7EFFFF7E3C18"
230 E$="FFFFFFFFFFFFFFFF"
240 CALL CHAR(97,B$)
250 CALL CHAR(98,A$)
260 CALL CHAR(99,C$)
270 CALL CHAR(104,E$)
280 CALL CHAR(105,A$)
290 FOR X=112 TO 152 STEP 8
300 CALL CHAR(X,D$)
310 NEXT X
320 CALL COLOR(9,2,1)
330 CALL COLOR(10,16,1)
340 CALL COLOR(11,8,16)
350 CALL COLOR(12,4,16)
360 CALL COLOR(13,11,16)
370 CALL COLOR(14,9,16)
380 CALL COLOR(15,14,16)
390 CALL COLOR(16,13,16)
400 REM  TITLE
410 CALL CLEAR
420 CALL SCREEN(3)
430 M$="CYBER-CIPHER"
440 R=5
450 C=10
460 GOSUB 2850
470 GOSUB 2820
480 REM  LEVEL OF DIFFICULTY
490 CALL CLEAR
500 PRINT "ENTER YOUR CHOICE:"::"1) EAS
    Y"::"2) MODERATE"::"3) HARD"::"4) P
    RO'S ONLY"
510 CALL KEY(0,K,S)
520 IF (S=0)+(K<49)+(K>52)THEN 510
530 LEV=K-48
540 NI=11-LEV
550 REM  DRAW GAME BOARD
560 CALL CLEAR
570 R=3
580 M$="CYBER CIPHER"
590 FOR D=1 TO LEN(M$)
600 CALL VCHAR(R+D-1,14,ASC(SEG$(M$,D,1
    )))
610 NEXT D
620 CALL VCHAR(1,13,97,24)
630 CALL VCHAR(1,15,97,24)
640 R=2
650 C=4
660 M$="KEYBOARD"
670 GOSUB 2850
680 R=3
690 M$="CODES"
700 GOSUB 2850
710 C=3
720 RESTORE 2890
730 FOR R=5 TO 15 STEP 2
740 READ X,A$
750 M$=CHR$(X)&" "&A$
760 GOSUB 2850
770 NEXT R
780 FOR R=1 TO 3
790 CALL HCHAR(R,18,104,9)
800 NEXT R
810 R=5
820 C=16
830 FOR X=1 TO NI
840 M$=STR$(X)
850 GOSUB 2850
860 R=R+2
870 NEXT X
880 CALL HCHAR(16,3,95,10)
890 C=5
900 R=17
910 M$="LOW"
920 GOSUB 2850
```

Continued

```
 930 R=18
 940 M$="SCORE"
 950 GOSUB 2850
 960 CALL HCHAR(19,7,104)
 970 CALL HCHAR(20,3,95,10)
 980 REM SET COLORS
 990 C=4
1000 R=22
1010 M$="YOU HAVE"
1020 GOSUB 2850
1030 R=23
1040 M$=STR$(NI)&" TRIES"
1050 GOSUB 2850
1060 FOR X=1 TO 4
1070 RANDOMIZE
1080 CM(X)=((INT(6*RND)+1)*8)+104
1090 NEXT X
1100 CALL SOUND(200,1047,2)
1110 FOR C=19 TO 25 STEP 2
1120 CALL HCHAR(2,C,99)
1130 NEXT C
1140 REM PLAYER'S GUESSES
1150 R=5
1160 X=1
1170 C=19
1180 IF R=NI*2+5 THEN 2270
1190 CALL KEY(0,KY,ST)
1200 IF ST=0 THEN 1190
1210 IF KY=66 THEN 1280
1220 IF KY=76 THEN 1310
1230 IF KY=89 THEN 1340
1240 IF KY=82 THEN 1370
1250 IF KY=77 THEN 1400
1260 IF KY=71 THEN 1430
1270 GOTO 1190
1280 P(X)=112
1290 SND=440
1300 GOTO 1450
1310 P(X)=120
1320 SND=494
1330 GOTO 1450
1340 P(X)=128
1350 SND=523
1360 GOTO 1450
1370 P(X)=136
1380 SND=587
1390 GOTO 1450
1400 P(X)=144
1410 SND=659
1420 GOTO 1450
1430 P(X)=152
1440 SND=740
1450 ON LEV GOTO 1460,1490,1460,1510
1460 CALL HCHAR(R,C,P(X))
1470 CALL SOUND(100,SND,0)
1480 GOTO 1520
1490 CALL HCHAR(R,C,P(X))
1500 GOTO 1520
1510 CALL SOUND(100,SND,0)
1520 IF X=4 THEN 1590
1530 X=X+1
1540 C=C+2
1550 FOR D=1 TO 50
1560 NEXT D
1570 GOTO 1190
1580 REM EVALUATE COLOR
1590 IF LEV<>3 THEN 1610
1600 CALL HCHAR(R,19,32,8)
1610 S=0
1620 T=0
1630 U=0
1640 V=0
1650 C=27
1660 W=0
1670 FOR X=1 TO 4
1680 IF P(1)=CM(X)THEN 1710
1690 NEXT X
1700 GOTO 1720
1710 GOSUB 2060
1720 W=0
1730 FOR X=1 TO 4
1740 IF P(2)=CM(X)THEN 1750 ELSE 1770
1750 GOSUB 2060
1760 IF W=1 THEN 1780
1770 NEXT X
1780 W=0
1790 FOR X=1 TO 4
1800 IF P(3)=CM(X)THEN 1810 ELSE 1830
1810 GOSUB 2060
1820 IF W=1 THEN 1840
1830 NEXT X
1840 W=0
1850 FOR X=1 TO 4
1860 IF P(4)=CM(X)THEN 1870 ELSE 1890
1870 GOSUB 2060
1880 IF W=1 THEN 1910
1890 NEXT X
1900 REM EVALUATE COLOR
     ND LOCATION
1910 Z=0
1920 IF P(1)=CM(1)THEN 1930 ELSE 1940
1930 GOSUB 2220
1940 IF P(2)=CM(2)THEN 1950 ELSE 1960
1950 GOSUB 2220
1960 IF P(3)=CM(3)THEN 1970 ELSE 1980
1970 GOSUB 2220
1980 IF P(4)=CM(4)THEN 1990 ELSE 2000
1990 GOSUB 2220
2000 IF R>5 THEN 2030
2010 CALL HCHAR(22,4,32,8)
2020 CALL HCHAR(23,4,32,8)
2030 IF Z=4 THEN 2420
2040 R=R+2
2050 GOTO 1160
2060 ON X GOTO 2070,2100,2130,2160
2070 IF S<>0 THEN 2210
2080 S=1
2090 GOTO 2180
2100 IF T<>0 THEN 2210
2110 T=1
2120 GOTO 2180
2130 IF U<>0 THEN 2210
2140 U=1
2150 GOTO 2180
2160 IF V<>0 THEN 2210
2170 V=1
2180 CALL HCHAR(R,C,105)
2190 W=1
2200 C=C+1
2210 RETURN
2220 C=C-1
2230 CALL HCHAR(R,C,98)
2240 Z=Z+1
2250 RETURN
2260 REM A LOSER
2270 C=4
2280 R=22
2290 M$="SORRY--"
2300 GOSUB 2850
2310 R=23
2320 M$="YOU LOSE"
2330 GOSUB 2850
2340 X=140*80
2350 CALL SOUND(-99,X,2)
2360 X=X/2
2370 IF X=55 THEN 2390
2380 GOTO 2350
2390 GOSUB 2820
2400 GOTO 2610
2410 REM A WINNER
2420 FOR X=4 TO 16 STEP 2
2430 CALL COLOR(10,X,1)
2440 FOR D=1 TO 50
2450 NEXT D
2460 NEXT X
2470 IF ((R-5)/2)+1<L THEN 2480 ELSE 2500
2480 L=((R-5)/2)+1
2490 CALL HCHAR(19,7,ASC(STR$(L)))
2500 T=60
2510 CALL SOUND(T,1047,2)
2520 CALL SOUND(T,110,30)
2530 CALL SOUND(T,1047,2)
2540 CALL SOUND(T,110,30)
2550 CALL SOUND(T,1047,2)
2560 CALL SOUND(T,110,30)
2570 CALL SOUND(T,1047,2)
2580 CALL SOUND(T*2,110,30)
2590 CALL SOUND(T*4,1319,2)
2600 CALL SOUND(T*6,2094,1)
2610 CALL HCHAR(2,19,CM(1))
2620 CALL HCHAR(2,21,CM(2))
2630 CALL HCHAR(2,23,CM(3))
2640 CALL HCHAR(2,25,CM(4))
2650 R=22
2660 C=4
2670 M$="REPLAY?"
2680 GOSUB 2850
2690 R=23
2700 M$="Y OR N"
2710 GOSUB 2850
2720 CALL KEY(0,KY,ST)
2730 IF ST=0 THEN 2720
2740 IF KY=89 THEN 490
2750 IF KY=78 THEN 2770
2760 GOTO 2720
2770 CALL CLEAR
2780 STOP
2790 FOR D=1 TO 2500
2800 NEXT D
2810 RETURN
2820 FOR D=1 TO 1000
2830 NEXT D
2840 RETURN
2850 FOR D=1 TO LEN(M$)
2860 CALL HCHAR(R,C+D-1,ASC(SEG$(M$,D,1)))
2870 NEXT D
2880 RETURN
2890 DATA 112,"BLUE=B",120,"LIME=L",128,"YEL.=Y"
2900 DATA 136,"RED =R",144,"MAG.=M",152,"GRN.=G"
```

HCM

ELEMENTARY ADDITION AND SUBTRACTION

COMMODORE 64

PROGRAM LISTING

```
100 REM  ********************
110 REM  * ADD SUBTRACT PROGRAM *
120 REM  ********************
130 REM  BY MARK DEWESE
140 REM  AND THE HCM STAFF
150 REM  HOME COMPUTER MAGAZINE
160 REM  VERSION 4.3.1
170 REM  C-64 BASIC
180 FOR I=15360 TO 15359+(2*64):READ A:
    POKE I,A:NEXT
190 POKE 52,60:POKE 56,60
200 POKE 53280,1:POKE 53281,1
210 S=54272:FOR I=0TO24:POKE S+I,0:NEXT
220 FOR I=1TO6:READ S2(I):NEXT
230 DATA 0,0,0,0,0,0,0,0,0,0,0,8,0,0,28,0
    ,0
240 DATA 62,0,0,127,0,127,255,255,63,25
    5,254,31,255,252,15,255
250 DATA 248,7,255,240,3,255,224,7,247,
    240,15,193,248,31,0,124
260 DATA 60,0,30,48,0,6,0,0,0,0,0,0,0,0
    ,0,0
270 DATA 192,0,56,224,0,120,240,0,240,1
    20,1,224,60,3,192,30
280 DATA 7,128,15,15,0,7,158,0,3,252,0,
    1,248,0,1,248
290 DATA 0,1,248,0,3,252,0,7,254,0,15,1
    5,0,30,7,128
300 DATA 60,3,192,120,1,224,240,0,240,2
    24,0,120,192,0,56,0
310 DATA 131,175,196,131,196,131
320 DATA 1047,988,880,784,698,659,587,5
    23
330 PRINT"■SHIFT CLR■CTRL BLU■3CRSRDO
    WN■15CRSRRIGHT■B A S I C":PRINT"■3
    CRSRDOWN■16CRSRRIGHT■ADDITION"
340 PRINT"■CRSRDOWN■"TAB(17)"A N D":PRI
    NT TAB(15)"■CRSRDOWN■SUBTRACTION"
350 PRINT "■4CRSRDOWN■                PRESS ANY
    KEY TO CONTINUE"
360 WG=33:FOR I=1TO8:READ S1(I):F1=S1(I
    ):GOSUB 950::NEXT
370 IF PEEK(197)=64 THEN 370
380 PRINT"■SHIFT CLR■4CRSRDOWN■ENTER 1
    FOR ADDITION":PRINT"                2 FOR SU
    BTRACTION"
390 GOSUB 920::IF A<>49 AND A<>50 THE
    N 390
400 OV=A-48
410 PRINT"■SHIFT CLR■3CRSRDOWN■ENTER:■
    2CRSRDOWN■":PRINT"1 FOR GRAPHICS AN
    D NUMBERS"
420 PRINT"  WITH GRAPHIC ANSWER GIVEN."
    :PRINT"■CRSRDOWN■2 FOR GRAPHICS AND
    NUMBERS"
430 PRINT"  WITHOUT GRAPHIC ANSWER.":PR
    INT"■CRSRDOWN■3 FOR NUMBERS ONLY."
440 GOSUB 920::IF A<49 OR A>51 THEN
    440
450 GV=A-48
460 PRINT"■SHIFT CLR■"
470 N=INT(RND(0)*9)+1
480 IF OV=1 THEN 520
490 LN=INT(RND(0)*N)+1:IF LN>=N/2 THEN
    RN=N-LN:GOTO 510
500 RN=LN:LN=N-RN
510 N=LN-RN:GOTO 530
520 LN=INT(RND(0)*N)+1:RN=N-LN
530 PN=LN:TB=5:GOSUB 1260::GOSUB 1250
540 IF GV>2 THEN 560
550 NC=LN:AD=1352:GOSUB 1310
560 IF OV=1 THEN GOSUB 1270::GOSUB 12
    20::GOTO 590
570 GOSUB 1290::GOSUB 1240
580 IF GV<3 THEN GOSUB 1300::GOSUB 12
    40::GOTO 600
590 IF GV<3 THEN GOSUB 1280::GOSUB 12
    20
600 PN=RN:TB=18:GOSUB 1260::GOSUB 125
    0
610 IF GV>2 THEN 630
620 AD=1365:NC=RN:GOSUB 1370
630 GOSUB 1270::GOSUB 1230
640 IF GV<3 THEN GOSUB 1280::GOSUB 12
    30
650 IF GV=1 THEN SN=N:AD=1380:NC=SN:GOS
    UB 1370
660 PRINT"■HOME■2DCRSRDOWN■5CRSRRIGHT
    ■PRESS <M> KEY FOR MENU."
670 GOSUB 920::GOTO 830
680 POKE 53248,15:POKE 53249,157:POKE 5
    3264,1
690 IF SN<>N THEN 770
700 POKE 2040,240:POKE 53287,7:WG=33
710 FOR J=1TO8:F1=S1(J):GOSUB 1410
720 IF PEEK(53269)=0 THEN POKE 53269,1:
    GOTO 750
730 POKE 53269,0
740 TB=31:GOSUB 1260::PN=SN:GOSUB 125
    0
750 GOSUB 950::NEXT
760 FOR I=1 TO 2000:NEXT:POKE 53269,0:G
    OTO 460
770 POKE 2040,241:POKE 53287,0:WG=33
780 FOR J=1TO8:F1=S2(J):GOSUB 1410
790 IF PEEK(53269)=0 THEN POKE 53269,1:
    GOTO 810
800 POKE 53269,0:TB=31:GOSUB 1260::PN=
    SN:GOSUB 1250
810 GOSUB 950::NEXT:IF GV=2 THEN GOSU
    B 1400
820 POKE 53269,0:GOSUB 1410::GOTO 670
830 IF A<48 OR A>57 THEN 900
840 IF GV=1 THEN GOSUB 1410::GOTO 860
850 GOSUB 1400
860 SN=A-48:TB=31:PN=SN:GOSUB 1260::GO
    SUB 1250
870 IF GV<>2 THEN 890
880 AD=1380:NC=SN:GOSUB 1370
890 GOTO 680
900 IF A=77 THEN 380
910 GOTO 670
920 GET A$:IF A$<>"" THEN 920
930 GET A$:IF A$="" THEN 930
940 A=ASC(A$):RETURN
950 FD=INT(F1/.06097):H1=INT(FD/256):L1
    =FD-(256*H1)
960 POKE S+5,0:POKE S+6,224
970 POKE S+24,15
980 POKE S+1,H1:POKE S,L1
990 POKE S+4,WG
1000 FOR ZX=1TO100:NEXT
1010 POKE S+4,WG-1
1020 POKE S+24,0:RETURN
1030 PRINT TAB(TB)"■CTRL BLK■■CTRL RVSON
    ■ ■CTRL RVSON■ ■CTRL RVSOFF■■CTRL B
    LU■":RETURN
1040 PRINT TAB(TB)"■CTRL RVSON■ ■CTRL RV
    SOFF■■CTRL RVSON■ ":PRINT TAB(TB)"
    ■CTRL RVSON■ ■CTRL RVSOFF■■CTRL B
    LU■":RETURN
1050 FOR I=1TO5:PRINT TAB(TB)"■CTRL RED■
    ■CTRL RVSON■■CTRL RVSOFF■":NEXT:
    PRINT"■CTRL BLU■":RETURN
1060 PRINT TAB(TB)"■CTRL PUR■■CTRL RVSON
    ■ ■CTRL RVSOFF■":PRINT TAB(TB)"■C
    TRL RVSON■ ":PRINT TAB(TB)"■CTRL
    RVSON■ ":PRINT TAB(TB)"■CTRL RV
    SOFF■":PRINT TAB(TB)"■CTRL RVSON■
    ■CTRL BLU■":RETURN
1070 PRINT TAB(TB)"■CTRL GRN■■CTRL RVSON
    ■ ■CTRL RVSOFF■":PRINT TAB(TB)"■C
    TRL RVSON■ ":PRINT TAB(TB)"■CTRL
    RVSOFF■":PRINT TAB(TB)"■CTRL RVSON
    ■ ■CTRL BLU■":RETURN
1080 PRINT TAB(TB)"■CTRL CYN■■CTRL RVSON
    ■ ■CTRL RVSOFF■":PRINTTAB(TB)"■C
    TRL RVSON■ ":PRINT TAB(TB)"■CTRL
    RVSON■ "
1090 PRINT TAB(TB)"  ■CTRL RVSON■ ":PRIN
    T TAB(TB)"■CTRL RVSON■ ■CTRL BLU■
    ":RETURN
1100 PRINT TAB(TB)"■CTRL PUR■■CTRL RVSON
    ■ ■CTRL RVSOFF■":PRINT TAB(TB)"■C
    TRL RVSON■ ":PRINT TAB(TB)"■CTRL
    RVSOFF■":PRINT TAB(TB)"■CTRL RVSO
    N■ ":PRINT TAB(TB)"■CTRL BLU■":RETURN
1110 PRINT TAB(TB)"  ■CTRL RVSON■■CTRL
    RVSOFF■":PRINT TAB(TB)"  ■CTRL RVSO
    N■ ■CTRL BLU■":RETURN
1120 PRINT TAB(TB)"■CTRL GRN■■CTRL RVSON
    ■ ■CTRL RVSOFF■":PRINT TAB(TB)"■C
    TRL RVSON■ ":PRINT TAB(TB)"■CTRL
    RVSOFF■":PRINT TAB(TB)"■CTRL RVSON
    ■ ■CTRL BLU■":RETURN
1130 PRINT TAB(TB)"  ■CTRL RVSON■ ":PRIN
    T TAB(TB)"■CTRL RVSON■ ■CTRL RVSO
    FF■■CTRL BLU■":RETURN
1140 PRINT TAB(TB)"■CTRL BLU■■CTRL RVSON
    ■ ■CTRL RVSOFF■":PRINT TAB(TB)"■C
    TRL RVSON■ ":PRINT TAB(TB)"■CTRL
    RVSON■ "
1150 PRINT TAB(TB)"■CTRL RVSON■ ■CTRL RV
    SOFF■■CTRL RVSON■ ":PRINT TAB(TB)"
    ■CTRL RVSON■ ":RETURN
1160 PRINT TAB(TB)"■CTRL RED■■CTRL RVSON
    ■ ■CTRL RVSOFF■":FOR I=1TO4:PRINT
    TAB(TB)"  ■CTRL RVSON■ ":NEXT:PRIN
    T"■CTRL BLU■"
1170 RETURN
1180 PRINT TAB(TB)"■CTRL RED■■CTRL RVSON
    ■ ■CTRL RVSOFF■":PRINT TAB(TB)"■C
    TRL RVSON■ ":PRINT TAB(TB)"■CTRL
    RVSON■ ":PRINT TA
    B(TB)"■CTRL RVSON■
1190 PRINT TAB(TB)"■CTRL RVSON■ ■CTRL RV
    SOFF■■CTRL RVSON■ ":PRINT TAB(TB)"
    ■CTRL BLU■":RETURN
1200 PRINT TAB(TB)"■CTRL GRN■■CTRL RVSON
    ■ ■CTRL RVSOFF■":PRINT TAB(TB)"■C
    TRL RVSON■ ":PRINT TA
    B(TB)"■CTRL RVSON■
1210 PRINT TAB(TB)"■CTRL RVSON■ ■CTRL
    RVSOFF■":PRINT TAB(TB)"■CTRL RVSO
    N■ ■CTRL RVSOFF■■CTRL BLU■":RETURN
1220 PRINT TAB(13)"■CTRL RVSON■ ":PRINT
    TAB(12)"■CTRL RVSON■ ":RETURN
1230 PRINT TAB(25)"■CTRL RVSON■ ":PRIN
    T TAB(25)"■CRSRDOWN■■CTRL RVSON■
    ":RETURN
1240 PRINT TAB(12)"■CTRL RVSON■ ":RETU
    RN
1250 ON PN+1 GOTO 1030,1050,1060,1080
    ,1100,1120,1140,1160,1180,120
    0
1260 PRINT"■HOME■11CRSRDOWN■":RETURN
1270 PRINT"■HOME■12CRSRDOWN■":RETURN
1280 PRINT"■HOME■2CRSRDOWN■":RETURN
1290 PRINT"■HOME■13CRSRDOWN■":RETURN
1300 PRINT"■HOME■3CRSRDOWN■":RETURN
1310 CL=((INT(RND(0)*4)+1)*2:CH=INT(RND(0
    )*4)+1
```

Continued

ELEMENTARY ADDITION AND SUBTRACTION

Continued

COMMODORE 64

```
1320 ON CH GOTO 1330, 1340, 1350, 1360
1330 CH=83:GOTO 1370
1340 CH=88:GOTO 1370
1350 CH=90:GOTO 1370
1360 CH=65
1370 IF NC=0 THEN RETURN
1380 FOR I=0 TO NC-1:IF I/3=INT(I/3) THE
     N AD=AD-86
1390 POKE AD+2,CH:POKE S+AD+2,CL:AD=AD+2
     :NEXT:RETURN
1400 PRINT"■HOME■";:FOR I=1TO20:PRINT TAB
     (30)";:NEXT:RETURN
1410 PRINT"■HOME■■SCRSRDOWN■";:FOR I=1TO1
     1:PRINT TAB(30)";:NEXT:RETUR
     N
```

HCM

ELEMENTARY ADDITION AND SUBTRACTION

TI-99/4A

```
100 REM **********************
110 REM * BASIC ADDITION   *
120 REM * AND SUBTRACTION  *
130 REM **********************
140 REM BY MARK DEWESE
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.3.1
170 REM TI BASIC
180 REM TI EXTENDED BASIC
190 CALL CLEAR
200 CALL SCREEN(2)
210 PRINT TAB(10);"B A S I C"
220 PRINT
230 PRINT TAB(11);"ADDITION"
240 PRINT
250 PRINT TAB(12);"A N D"
260 PRINT
270 PRINT TAB(9);"SUBTRACTION":::::
    :
280 PRINT "PRESS THE [S] KEY IF YOU ARE
    USING EXTENDED BASIC AND A"
290 PRINT "SPEECH SYNTHESIZER."
300 PRINT :"OTHERWISE --
    PRESS ANY OTHER KEY TO BEGIN"
310 PRINT ::
320 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
330 CALL HCHAR(1,1,128,32)
340 CALL HCHAR(2,1,128,32)
350 CALL HCHAR(23,1,128,32)
360 CALL HCHAR(24,1,128,32)
370 CALL VCHAR(3,1,128,20)
380 CALL VCHAR(3,2,128,20)
390 CALL VCHAR(3,31,128,20)
400 CALL VCHAR(3,32,128,20)
410 CALL COLOR(13,6,6)
420 CALL SCREEN(10)
430 CALL SOUND(300,131,4,1047,4)
440 CALL SOUND(300,147,4,988,4)
450 CALL SOUND(300,165,4,880,4)
460 CALL SOUND(300,175,4,784,4)
470 CALL SOUND(300,196,4,698,4)
480 CALL SOUND(300,220,4,659,4)
490 CALL SOUND(300,247,4,587,4)
500 CALL SOUND(600,262,4,523,4)
510 CALL KEY(0,JWD,STATUS)
520 IF STATUS=0 THEN 510
530 IF JWD<>83 THEN 550
540 FL=1
550 CALL CLEAR
560 QWL=0
570 PRINT "PRESS"
580 PRINT
590 PRINT "1 FOR ADDITION"
600 PRINT
610 PRINT "2 FOR SUBTRACTION"
620 FOR K=1 TO 10
630 PRINT
640 NEXT K
650 CALL KEY(0,MODE,STATUS)
660 IF STATUS=0 THEN 650
670 IF (MODE<49)+(MODE>50)THEN 650
680 V=0
690 U=0
700 PRINT "PRESS"
710 PRINT
720 PRINT "1 FOR GRAPHICS AND NUMBERS"
730 PRINT
740 PRINT " WITH GRAPHIC ANSWER GIVEN"
750 PRINT
760 PRINT "2 FOR GRAPHICS AND NUMBERS"
770 PRINT
780 PRINT " WITHOUT GRAPHIC ANSWER"
790 PRINT
800 PRINT "3 FOR NUMBERS ONLY":::::
    ::
810 CALL KEY(0,COMB,STATUS)
820 IF STATUS=0 THEN 810
830 IF (COMB<49)+(COMB>51)THEN 810
840 CALL SCREEN(16)
850 CALL CLEAR
860 G$="FFFFFFFFFFFFFFFF"
870 CALL CHAR(130,G$)
880 CALL CHAR(60,G$)
890 CALL CHAR(34,G$)
900 CALL CHAR(123,G$)
910 CALL CHAR(41,G$)
920 PRINT "PRESS [M] TO RETURN TO MENU"
930 RANDOMIZE
940 PIC=INT(5*RND)+1
950 IF COMB=51 THEN 1160
960 ON PIC GOTO 980,1020,1060,1100,1140
970 REM HEARTS
980 CALL CHAR(136,"42E7FFFFFF7E3C18")
990 CALL COLOR(14,9,16)
1000 GOTO 1160
1010 REM SQUARES
1020 CALL CHAR(137,"FFFFFFFFFFFFFFFF")
1030 CALL COLOR(14,4,4)
1040 GOTO 1160
1050 REM BELLS
1060 CALL CHAR(138,"183C3C3C3C3C7EFF")
1070 CALL COLOR(14,10,16)
1080 GOTO 1160
1090 REM CIRCLES
1100 CALL CHAR(139,"3C7EFFFFFFFF7E3C")
1110 CALL COLOR(14,12,16)
1120 GOTO 1160
1130 REM DIAMONDS
1140 CALL CHAR(140,"183C7EFFFF7E3C18")
1150 CALL COLOR(14,6,16)
1160 RANDOMIZE
1170 LEFT=INT(5*RND)+1
1180 IF COMB=51 THEN 1270
1190 B=PIC+135
1200 ON LEFT GOTO 1250,1240,1230,1220,12
     10
1210 CALL VCHAR(10,5,B)
1220 CALL VCHAR(6,5,B)
1230 CALL VCHAR(6,7,B)
1240 CALL VCHAR(8,7,B)
1250 CALL VCHAR(8,5,B)
1260 REM LEFT NUMBER
1270 ON LEFT GOTO 1290,1330,1420,1490,15
     50
1280 REM #1
1290 CALL COLOR(13,6,6)
1300 CALL VCHAR(15,6,130,7)
1310 GOTO 1620
1320 REM #2
1330 CALL COLOR(13,10,10)
1340 CALL VCHAR(16,4,130,1)
1350 CALL HCHAR(15,4,130,4)
1360 CALL VCHAR(16,7,130,3)
1370 CALL HCHAR(18,4,130,3)
1380 CALL VCHAR(19,4,130,3)
1390 CALL HCHAR(21,5,130,3)
1400 GOTO 1620
1410 REM #3
1420 CALL COLOR(13,4,4)
1430 CALL HCHAR(15,4,130,4)
1440 CALL VCHAR(16,7,130,6)
1450 CALL HCHAR(18,5,130,2)
1460 CALL HCHAR(21,4,130,3)
1470 GOTO 1620
1480 REM #4
1490 CALL COLOR(13,12,12)
1500 CALL VCHAR(15,4,130,4)
1510 CALL HCHAR(18,5,130,2)
1520 CALL VCHAR(15,7,130,7)
1530 GOTO 1620
1540 REM #5
1550 CALL COLOR(13,8,8)
1560 CALL HCHAR(15,4,130,4)
1570 CALL VCHAR(16,4,130,1)
1580 CALL HCHAR(17,4,130,4)
1590 CALL VCHAR(18,7,130,4)
1600 CALL VCHAR(20,4,130,1)
1610 CALL HCHAR(21,4,130,3)
1620 IF MODE=50 THEN 1720
1630 REM +SIGN
1640 CALL COLOR(4,2,16)
1650 CALL VCHAR(17,11,60,3)
1660 CALL HCHAR(18,10,60,3)
1670 IF COMB=51 THEN 1760
1680 CALL VCHAR(6,11,60,3)
1690 CALL HCHAR(7,10,60,3)
1700 GOTO 1760
1710 REM -SIGN
1720 CALL COLOR(4,2,16)
1730 CALL HCHAR(18,10,60,3)
1740 IF COMB=51 THEN 1760
1750 CALL HCHAR(7,10,60,3)
1760 RANDOMIZE
1770 IF MODE=49 THEN 1800
1780 RIGHT=INT(LEFT*RND)+1
1790 GOTO 1830
1800 RIGHT=INT(5*RND)+1
1810 IF (RIGHT=5)*(LEFT=5)THEN 1820 ELSE
     1830
1820 RIGHT=4
1830 IF COMB=51 THEN 1910
1840 ON RIGHT GOTO 1890,1880,1870,1860,1
     850
1850 CALL VCHAR(10,15,B)
1860 CALL VCHAR(6,15,B)
1870 CALL VCHAR(6,17,B)
1880 CALL VCHAR(8,17,B)
```

Continued

PROGRAM LISTING

ELEMENTARY ADDITION AND SUBTRACTION

Continued

TI-99/4A

PROGRAM LISTING

```
1890 CALL VCHAR(8,15,B)
1900 REM RIGHT NUMBER
1910 ON RIGHT GOTO 1930,1970,2060,2130,2
     190
1920 REM #1
1930 CALL COLOR(1,6,16)
1940 CALL VCHAR(15,16,34,7)
1950 GOTO 2270
1960 REM #2
1970 CALL COLOR(1,10,16)
1980 CALL VCHAR(16,15,34,1)
1990 CALL HCHAR(15,15,34,4)
2000 CALL VCHAR(16,18,34,2)
2010 CALL HCHAR(18,15,34,4)
2020 CALL VCHAR(19,15,34,2)
2030 CALL HCHAR(21,15,34,4)
2040 GOTO 2270
2050 REM #3
2060 CALL COLOR(1,4,16)
2070 CALL HCHAR(15,15,34,4)
2080 CALL VCHAR(16,18,34,6)
2090 CALL HCHAR(18,16,34,2)
2100 CALL HCHAR(21,15,34,3)
2110 GOTO 2270
2120 REM #4
2130 CALL COLOR(1,12,16)
2140 CALL VCHAR(15,15,34,4)
2150 CALL HCHAR(18,16,34,2)
2160 CALL VCHAR(15,18,34,7)
2170 GOTO 2270
2180 REM #5
2190 CALL COLOR(1,8,16)
2200 CALL HCHAR(15,15,34,4)
2210 CALL VCHAR(16,15,34,1)
2220 CALL HCHAR(17,15,34,4)
2230 CALL VCHAR(18,18,34,4)
2240 CALL VCHAR(20,15,34,1)
2250 CALL HCHAR(21,15,34,4)
2260 REM =SIGN
2270 CALL COLOR(12,2,16)
2280 CALL HCHAR(17,21,123,3)
2290 CALL HCHAR(19,21,123,3)
2300 IF COMB=51 THEN 2330
2310 CALL HCHAR(6,21,123,3)
2320 CALL HCHAR(8,21,123,3)
2330 IF MODE=49 THEN 2370
2340 ANSW=LEFT-RIGHT
2350 IF ANSW=0 THEN 2550
2360 GOTO 2390
2370 ANSW=LEFT+RIGHT
2380 QWL=0
2390 IF (COMB=50)+(COMB=51)THEN 2550
2400 IF QWL<>0 THEN 2430
2410 QWL=ANSW
2420 IF QWL=0 THEN 3010
2430 ON QWL GOTO 2530,2520,2510,2500,249
     0,2480,2470,2460,2450,2440
2440 CALL VCHAR(11,26,B)
2450 CALL VCHAR(9,30,B)
2460 CALL VCHAR(7,30,B)
2470 CALL VCHAR(5,30,B)
2480 CALL VCHAR(9,28,B)
2490 CALL VCHAR(9,26,B)
2500 CALL VCHAR(5,26,B)
2510 CALL VCHAR(5,28,B)
2520 CALL VCHAR(7,28,B)
2530 CALL VCHAR(7,26,B)
2540 IF COMB=50 THEN 3010
2550 CALL SOUND(300,1200,4)
2560 IF FL=1 THEN 3740
2570 REM GET INPUT OF PLAYER
2580 CALL KEY(0,QWL,STATUS)
2590 IF STATUS=0 THEN 2580
2600 IF STATUS=-1 THEN 2580
2610 IF QWL=77 THEN 550
2620 IF (QWL<48)+(QWL>57)THEN 2580
2630 QWL=QWL-48
2640 RANDOMIZE
2650 FOR S=1 TO 10
2660 T=INT(3000*RND)+110
2670 CALL SOUND(5,T,4)
2680 NEXT S
2690 IF QWL=ANSW THEN 2950
2700 CALL SOUND(600,-3,3)
2710 GOTO 2990
2720 IF FL=1 THEN 4270
2730 FOR EC=26 TO 30
2740 IF (COMB=49)+(COMB=51)THEN 2760
2750 CALL VCHAR(4,EC,32,8)
2760 CALL VCHAR(15,EC,32,7)
2770 NEXT EC
2780 GOTO 2570
2790 U=U+1
2800 GOTO 4150
2810 V=V+1
2820 IF V=10 THEN 2870
2830 IF FL=1 THEN 4260
2840 FOR DELAY=1 TO 1000
2850 NEXT DELAY
2860 GOTO 840
2870 W=V-U
2880 CALL CLEAR
2890 PRINT "YOUR SCORE IS";W
2900 PRINT
2910 PRINT "OUT OF 10"
2920 FOR L=1 TO 1000
2930 NEXT L
2940 GOTO 550
2950 FOR Z=1 TO 6
2960 CALL SOUND(20,2000,3)
2970 CALL SOUND(20,1700,3)
2980 NEXT Z
2990 REM PRINT ANSWER
3000 IF COMB=50 THEN 2420
3010 ON QWL+1 GOTO 3020,3090,3130,3220,3
     290,3350,3440,3530,3590,3670
3020 CALL COLOR(2,15,16)
3030 CALL HCHAR(15,26,41,4)
3040 CALL VCHAR(16,26,41,6)
3050 CALL HCHAR(21,27,41,3)
3060 CALL VCHAR(16,29,41,5)
3070 GOTO 3720
3080 REM #1
3090 CALL COLOR(2,6,16)
3100 CALL VCHAR(15,27,41,7)
3110 GOTO 3720
3120 REM #2
3130 CALL COLOR(2,10,16)
3140 CALL VCHAR(16,26,41,1)
3150 CALL HCHAR(15,26,41,4)
3160 CALL VCHAR(16,29,41,3)
3170 CALL HCHAR(18,26,41,3)
3180 CALL VCHAR(19,26,41,3)
3190 CALL HCHAR(21,27,41,3)
3200 GOTO 3720
3210 REM #3
3220 CALL COLOR(2,4,16)
3230 CALL HCHAR(15,26,41,4)
3240 CALL VCHAR(16,29,41,6)
3250 CALL HCHAR(18,27,41,2)
3260 CALL HCHAR(21,26,41,3)
3270 GOTO 3720
3280 REM #4
3290 CALL COLOR(2,12,16)
3300 CALL VCHAR(15,26,41,4)
3310 CALL HCHAR(18,26,41,3)
3320 CALL VCHAR(15,29,41,7)
3330 GOTO 3720
3340 REM #5
3350 CALL COLOR(2,8,16)
3360 CALL HCHAR(15,26,41,4)
3370 CALL VCHAR(16,26,41,1)
3380 CALL HCHAR(17,26,41,4)
3390 CALL VCHAR(18,29,41,4)
3400 CALL VCHAR(20,26,41,1)
3410 CALL HCHAR(21,26,41,3)
3420 GOTO 3720
3430 REM #6
3440 CALL COLOR(2,10,16)
3450 CALL HCHAR(15,26,41,4)
3460 CALL VCHAR(16,29,41,1)
3470 CALL VCHAR(16,26,41,6)
3480 CALL HCHAR(18,27,41,3)
3490 CALL VCHAR(19,29,41,3)
3500 CALL HCHAR(21,27,41,2)
3510 GOTO 3720
3520 REM #7
3530 CALL COLOR(2,9,16)
3540 CALL VCHAR(16,26,41,1)
3550 CALL HCHAR(15,26,41,4)
3560 CALL VCHAR(16,29,41,6)
3570 GOTO 3720
3580 REM #8
3590 CALL COLOR(2,12,16)
3600 CALL HCHAR(15,26,41,4)
3610 CALL HCHAR(18,27,41,2)
3620 CALL HCHAR(21,26,41,4)
3630 CALL VCHAR(16,26,41,5)
3640 CALL VCHAR(16,29,41,5)
3650 GOTO 3720
3660 REM #9
3670 CALL COLOR(2,5,16)
3680 CALL HCHAR(15,26,41,4)
3690 CALL VCHAR(16,26,41,2)
3700 CALL HCHAR(18,26,41,3)
3710 CALL VCHAR(16,29,41,6)
3720 IF ANSW=QWL THEN 4030 ELSE 3930
3730 REM SPEACH
3740 CALL SAY("WHAT+IS")
3750 IF MODE=50 THEN 3830
3760 ON LEFT GOTO 3770,3780,3790,3800,38
     10
3770 CALL SAY("ONE"):: GOTO 3820
3780 CALL SAY("TWO"):: GOTO 3820
3790 CALL SAY("THREE"):: GOTO 3820
3800 CALL SAY("FOUR"):: GOTO 3820
3810 CALL SAY("FIVE")
3820 IF MODE=49 THEN 3900 ELSE 3920
3830 ON RIGHT GOTO 3840,3850,3860,3870,3
     880
3840 CALL SAY("ONE"):: GOTO 3890
3850 CALL SAY("TWO"):: GOTO 3890
3860 CALL SAY("THREE"):: GOTO 3890
3870 CALL SAY("FOUR"):: GOTO 3890
3880 CALL SAY("FIVE")
3890 IF MODE=50 THEN 3910 ELSE 3920
3900 CALL SAY("AND"):: GOTO 3830
3910 CALL SAY("FROM"):: GOTO 3760
3920 GOTO 2570
3930 IF FL=1 THEN 3950
3940 GOTO 2720
3950 WRONG=INT(5*RND)+1
3960 ON WRONG GOTO 3970,3980,3990,4000,4
     010
```

Continued

ELEMENTARY ADDITION AND SUBTRACTION

Continued

TI-99/4A

```
3970 CALL SAY("I+AM+SORRY"):: GOTO 4020
3980 CALL SAY("UHOH, THAT+IS+NOT+THE1+RI
     GHT ANSWER"):: GOTO 4020
3990 CALL SAY("THAT+IS NOT RIGHT"):: GOT
     O 4020
4000 CALL SAY("THAT IS IN+CORRECT"):: GO
     TO 4020
4010 CALL SAY("THAT IS+NOT+THE ANSWER")
4020 GOTO 2720
4030 IF FL<>1 THEN 4120
4040 CORR=INT(5*RND)+1
4050 ON CORR GOTO 4060,4070,4080,4090,41
     00
4060 CALL SAY("THAT+IS+RIGHT YOU+HAVE+TH
     E1+CORRECT ANSWER"):: GOTO 4110
4070 CALL SAY("GOOD WORK"):: GOTO 4110
4080 CALL SAY("THAT IS CORRECT"):: GOTO
     4110
4090 CALL SAY("VERY GOOD"):: GOTO 4110
4100 CALL SAY("YOU+ARE EXACTLY RIGHT")
4110 CALL SAY("THE ANSWER IS")
4120 REM
4130 IF FL<>1 THEN 2810
4140 ON ANSW+1 GOTO 4150,4160,4170,4180,
     4190,4200,4210,4220,4230,4240
4150 CALL SAY("ZERO"):: GOTO 2810
4160 CALL SAY("ONE"):: GOTO 2810
4170 CALL SAY("TWO"):: GOTO 2810
4180 CALL SAY("THREE"):: GOTO 2810
4190 CALL SAY("FOUR"):: GOTO 2810
4200 CALL SAY("FIVE"):: GOTO 2810
4210 CALL SAY("SIX"):: GOTO 2810
4220 CALL SAY("SEVEN"):: GOTO 2810
4230 CALL SAY("EIGHT"):: GOTO 2810
4240 CALL SAY("NINE"):: GOTO 2810
4250 CALL SAY("TEN"):: GOTO 2810
4260 CALL SAY("NOW+TRY+THIS"):: GOTO 840
4270 CALL SAY("TRY AGAIN"):: GOTO 2730
```

HCM

MISSILE MATH

APPLE II Family

```
100 REM  ****************
110 REM  * MISSILE MATH *
120 REM  ****************
130 REM  BY J.C. ADELMANN
140 REM  AND THE HCM STAFF
150 REM  HOME COMPUTER MAGAZINE
160 REM  VERSION 4.3.1
170 REM  APPLE II SERIES APPLESOFT
180 HOME
190 VTAB 10
200 INVERSE : PRINT "
                          ": NORMAL
210 PRINT "   M   I   S   S   I   L   E   M
    A   T   H"
220 INVERSE : PRINT "
                          ": NORMAL
230 PRINT "PLEASE WAIT WHILE I GET MY M
    ISSILES...   "
240 POKE 232,0: POKE 233,96: FOR K = 0
    TO 38: READ D: POKE (770 + K),D: NE
    XT K
250 DATA 172,1,3,174,1,3,169,4,32,168,2
    52,173,48,192,232,208,253,136,208,2
    39,206,0,3,208,231,96,32,32,112,8,2
    4,216,136,8,160,160,16,56,0
260 FOR K = 0 TO 568: READ Z: POKE (245
    76 + K),Z: NEXT K
270 REM ** 280,400 IS RESERVED     FOR
    SHAPE TABLE DATA **
280 DATA 14,0,30,0,51,0,71,0,91,0,112,0
    ,133,0,154,0,174,0,194,0,215,0,247,
    0,12,1,31,1,51,1,41,109,26,31,27,11
    0,41,21,59,59,51,109,41,26,31,27,14
    ,45,13,2,0,9,77,26,27,63,74,77,26,2
    7,31
290 DATA 74,77,26,27,31,10,45,13,2,0,41
    ,109,26,31,27,78,9,21,27,63,83,77,1
    7,27,27,51,45,45,21,0,45,45,21,59,2
    7,83,9,13,26,59,31,74,9,21,59,27,11
    5,45,13,2,0,73,13,26,59,31,10,13,13
    ,26,59
300 DATA 27,46,45,45,26,59,27,74,105,2,
    0,45,45,21,27,27,51,45,109,26,31,27
    ,74,9,21,59,27,115,45,13,2,0,41,109
    ,26,31,27,110,73,26,59,63,110,9,21,
    59,27,115,45,13,2,0,45,45,21,59,27,
    83,9,13,26
310 DATA 27,31,10,77,17,27,27,87,77,17,
    0,41,109,26,31,27,110,9,21,27,63,23
    ,77,41,26,31,27,14,45,13,2,0,41,109
    ,26,31,27,110,9,21,59,63,87,73,21,5
    9,27,115,45,13,2,0,59,63,87,73,21,5
    9,27,115,45
320 DATA 13,2,0,9,77,26,59,59,106,9,21,
    59,27,51,45,45,21,59,27,51,77,41,2,
    0,45,109,26,31,59,10,77,21,27,63,87
    ,77,21,59,27,23,45,109,2,0,9,109,26
    ,31,59,106,73,26,27,27,110,73,26,31
    ,59,74,109
330 DATA  2,0,63,63,63,63,63,63,60,60,6
    0,60,60,60,44,44,44,44,44,44,44,45,
    45,45,45,45,45,54,54,54,54,54,54,46
    ,45,45,45,45,45,45,45,45,45,45,45,4
    5,45,45,37,36,36,36,36,36,36,63,63,
    63,63,63,63
340 DATA 63,63,63,63,63,63,63,63,63,63,
    63,47,45,45,45,45,45,45,45,44,37,45
    ,44,37,45,44,37,37,37,37,37,45,45,4
    5,45,45,61,55,62,54,55,62,54,62,63,
    63,63,63,63,63,63,63,44,45,45,45,45
    ,45,45,45,37
350 DATA 63,63,63,63,63,63,44,45,45,45,
    45,45,45,60,63,63,63,63,39,45,45,45
    ,45,45,60,63,63,63,44,45,45,45,45,6
    2,36,61,63,63,191,146,146,210,219,2
    19,219,219,155,146,146,146,146
360 DATA 202,48,45,46,53,45,46,53,45,46
    ,46,46,46,46,45,45,45,45,45,63,36,3
    9,60,36,39,60,36,62,63,63,63,63,63,
    63,63,63,46,45,45,45,45,45,45,45,45
    ,62,63,63,63,63,63
370 DATA  63,46,45,45,45,45,45,45,53,61
    ,63,63,63,63,63,53,45,45,45,45,45,5
    3,63,63,63,63,63,46,45,45,45,45,45,
    62,63,63,63,63,7,0
380 GOSUB 1570: POKE - 16368,0
390 HGR :A = 179:B = 72
400 SCALE= 1: ROT= 12: DRAW 14 AT A,B
410 KYBD = PEEK ( - 16384): IF KYBD >
    127 THEN  GOTO 480
420 HCOLOR= 0: DRAW 14 AT A,B: HCOLOR=
    3
430 A = A - 2:B = B - 1
440 IF A = < 5 THEN  GOTO 390
450 VTAB 23: PRINT "PRESS ANY KEY TO CO
    NTINUE...          ": POKE 34,24
460 IF B = < 35 THEN  GOTO 390
470 GOTO 400
480 POKE 34,0: POKE - 16368,0
490 HOME : TEXT
500 PRINT "HELLO!"
510 INPUT "MY NAME IS APPLE. WHAT'S YOU
    RS? ";NAME$
520 PV = 32768: POKE PV, LEN (NAME$): F
    OR I = 1 TO  LEN (NAME$): POKE PV +
      I,( ASC ( MID$ (NAME$,I,1))): NEXT
      I
530 PRINT "GLAD TO MEET YOU, ";NAME$;".
    "
540 PRINT
550 PRINT "LET'S GET STARTED."
560 FOR DLY = 1 TO 1500: NEXT DLY
570 HOME : PRINT "DO YOU WANT INSTRUCTI
    ONS (Y/N)?";: GET YN$: IF YN$ = "Y"
      THEN  GOTO 1350
580 IF YN$ < > "N" GOTO 570
590 HOME : PRINT "WHICH NUMBER WOULD YO
    U LIKE TO PRACTICE MULTIPLYING? (1-
    9)";
600 X1 = 200:Y1 = 40:X2 = 200:Y2 = 80:X
    3 = 200:Y3 = 120
610 GET FF$:
620 IF  VAL (FF$) < 1 GOTO 590
630 FF =  VAL (FF$): HOME
640 NAME$ = "":PV = 32768:LN = ( PEEK (
    PV)): FOR I = 1 TO LN:NAME$ = NAME$
      + CHR$ ( PEEK (PV + I)): NEXT I
650 IF FF = 0 THEN  GOTO 590
660 PRINT : PRINT : PRINT : PRINT
670 LP = 0
680 PRINT "DO YOU WANT A TIMES TABLE,";
    NAME$;"?": PRINT "   (Y/N)";
690 GET YN$: IF YN$ = "N" THEN 780
700 IF YN$ < > "Y" THEN 690
710 HOME : PRINT " ======= T I M E S
    T A B L E ======     "
720 PRINT : PRINT
730 FOR I = 1 TO 9
740 PRINT I;".     ";I;" * ";FF;" = ";I *
      FF
750 PRINT
760 NEXT I
770 POKE - 16368,0: PRINT "PRESS A KEY
      WHEN YOU ARE READY TO START.";: GE
    T ANYKEY$
780 REM ** MAIN LOOP **
790 ROT= 0
800 LP = LP + 1
810 HGR
820 GOSUB 1490
830 B = LP - 1
840 C =  INT ( RND (1) * 10)
850 D(B) = C
860 HOME : VTAB 21: PRINT LP;". ";: PRI
    NT FF;" * ";D(B);" =": PRINT "YOUR
    CHOICE: ";
870 CR =  INT ( RND (1) * 3) + 1
880 IF CR = 1 THEN CX = X1:CY = Y1
890 IF CR = 2 THEN CX = X2:CY = Y2
900 IF CR = 3 THEN CX = X3:CY = Y3
```

Continued

PROGRAM LISTING

MISSILE MATH

Continued

APPLE II Family

```
910 R1 = INT ( RND (1) * 10) * FF:R2 = INT (( RND (1) * 10) * ( RND (1) * 10))
920 CA = FF * D(B)
930 IF R1 = CA OR R1 < 2 THEN GOTO 910
940 IF R2 = (CA) OR R2 < 2 THEN GOTO 910
950 CA$ = STR$ (CA):C1$ = LEFT$ (CA$,1):C2$ = RIGHT$ (CA$,1)
960 C1 = VAL (C1$):C2 = VAL (C2$)
970 IF C1 = C2 THEN GOTO 990
980 DRAW (C1 + 1) AT CX + 10,CY - 10
990 DRAW (C2 + 1) AT (CX + 17),CY - 10
1000 R1$ = STR$ (R1):R2$ = STR$ (R2):R
X$ = LEFT$ (R1$,1):RY$ = RIGHT$ (
R1$,1):RA$ = LEFT$ (R2$,1):RB$ =
RIGHT$ (R2$,1)
1010 RA = VAL (RA$):RB = VAL (RB$):RX
= VAL (RX$):RY = VAL (RY$)
1020 IF CR = 1 THEN XB = X2:YB = Y2
1030 IF CR = 1 THEN XC = X3:YC = Y3
1040 IF CR = 2 THEN XC = X3:YC = Y3
1050 IF CR = 2 THEN XB = X1:YB = Y1
1060 IF CR = 3 THEN XB = X2:YB = Y2
1070 IF CR = 3 THEN XC = X1:YC = Y1
1080 DRAW (RX + 1) AT XB + 10,YB - 10
1090 DRAW RY + 1 AT XB + 17,YB - 10
1100 DRAW RA + 1 AT XC + 10,YC - 10
1110 DRAW RB + 1 AT XC + 17,YC - 10
1120 ANS = 0
1130 POKE - 16368,0
1140 GET ANS
1150 IF ANS$ = "A" THEN ANS = 1
1160 IF ANS$ = "B" THEN ANS = 2
1170 IF ANS$ = "C" THEN ANS = 3
1180 IF ANS = 0 THEN GOTO 1140
1190 PRINT ANS$
1200 IF ANS = CR THEN GOSUB 1570
1210 IF ANS = CR THEN WIN = WIN + 1
1220 IF ANS = CR THEN PRINT "CORRECT!!"
1230 IF ANS < > CR THEN GOSUB 1630
1240 IF ANS < > CR THEN PRINT "NO, "; NAME$;", IS CORRECT.": PRINT FF;" * ";D(B);" = ";F
F * D(B): FOR DE = 1 TO 1500: NEXT DE
DE: GOTO 1270
1250 FOR DE = 1 TO 1000: NEXT DE
1260 GOSUB 1450
1270 IF LP < 10 THEN GOTO 780
1280 TEXT : HOME
1290 PRINT NAME$;",": PRINT "YOU GOT ";WIN;" RIGHT OUT OF 10 TRIES.": PRINT : PRINT " THIS IS A ";WIN * 10;"% RIGHT."
1300 POKE - 16368,0: PRINT : PRINT "PLAY AGAIN? (Y/N)";: GET YN$
1310 IF YN$ = "Y" THEN RUN 590
1320 HOME : END
1330 IF A = 11 THEN GOTO 790
1340 END
1350 REM ** INSTRUCTIONS **
1360 HOME : PRINT "INSTRUCTIONS:"
1370 PRINT " ";NAME$;","
1380 PRINT " YOU ARE GIVEN A MULTIPLICATION"
1390 PRINT " PROBLEM AT THE BOTTOM OF THE SCREEN."
1400 PRINT " YOU WILL ANSWER IT BY CHOOSING THE"
1410 PRINT "MISSLE WITH THE CORRECT ANSWER IN IT."
1420 PRINT : PRINT "IT'S AS SIMPLE AS THAT!"
1430 VTAB 24: PRINT "PRESS A KEY WHEN YOU WANT TO CONTINUE";: GET ANYKEY$
1440 GOTO 590
1450 REM ** MOVE ROUTINE **
1460 X1 = X1 - 14
1470 X2 = X2 - 14
1480 X3 = X3 - 14: RETURN
1490 HCOLOR= 5
1500 DRAW 14 AT X1,Y1: DRAW 14 AT X2,Y2: DRAW 14 AT X3,Y3
1510 HCOLOR= 3
1520 DRAW 11 AT (X1 - 8),(Y1 - 10)
1530 DRAW 12 AT (X2 - 8),(Y2 - 10)
1540 DRAW 13 AT (X3 - 8),(Y3 - 10)
1550 HCOLOR= 7
1560 RETURN
1570 REM RIGHT ANSWER SOUND
1580 POKE 768,4: POKE 769,30: CALL 770
1590 POKE 768,4: POKE 769,80: CALL 770
1600 POKE 768,4: POKE 769,30: CALL 770
1610 POKE 768,10: POKE 769,80: CALL 770
1620 RETURN
1630 REM WRONG ANSWER SOUND
1640 POKE 768,80: POKE 769,5: CALL 770: RETURN
```

HCM

MISSILE MATH

COMMODORE 64

```
100 REM ********************
110 REM * MISSILE MATH *
120 REM ********************
130 REM BY J.C. ADELMANN
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.3.1
170 REM C-64 BASIC
180 GOTO880
190 REM SOUND EFFECTS AND OTHER SUB-RTNS
200 POKESO+24,15:POKESO+5,17:POKESO+6,241:POKESO,100
210 FORX=1TO12:POKESO+1,T1(X):POKESO+4,17:FORD=1TO5:NEXT:NEXT:GOTO240
220 FORX=1TO12:POKESO+1,T2(X):POKESO+4,17:FORD=1TO5:NEXT:NEXT:GOTO240
230 FORX=1TO12:POKESO+1,T3(X):POKESO+4,17:FORD=1TO5:NEXT:NEXT:GOTO240
240 POKESO,16:RETURN
250 FORX=0TO24:POKESO+X,0:NEXT:POKESO+3,1:POKESO+24,143:POKESO+6,240
260 POKESO+4,65:FORX=1TO120:POKESO,8:POKESO+1,8:NEXT:POKES+24,0
270 FORX=0TO24:POKESO+X,0:NEXT:RETURN
280 AA=E:BB=B:CC=C:E$="A":RETURN
290 AA=C:BB=E:CC=B:E$="B":RETURN
300 AA=B:BB=C:CC=E:E$="C":RETURN
310 PRINT"■SHIFT CLR■17CRSRDOWN■";:FORX=1TO7
320 PRINT"■CTRL YEL■CTRL RVSON■ ";:NEXT:POKE646,0:RETURN
330 REM MAIN LOOP
340 FORJ=1TO10:A=INT(RND(0)*10):B=INT(RND(0)*12)+4:C=INT(RND(0)*24)+2
350 D=INT(RND(1)*3)+1:E=N*A:B=E+B:IFABS(E-C)>10THENC=ABS(E-C):GOTO370
360 C=E+C
370 IFD=1THENGOSUB280
380 IFD=2THENGOSUB290
390 IFD=3THENGOSUB300
400 PRINT"■CTRL RVSOFF■HOME■4CRSRDOWN■"SPC(M)"A"AA:PRINT"■4CRSRDOWN■"SPC(M)"B"BB:PRINT"■4CRSRDOWN■"SPC(M)"C"CC
410 PRINTSPC(10)"■3CRSRDOWN■"
420 PRINT"■SHIFT CRSRUP■"SPC(10)"A"="A"
430 PRINT"■CRSRDOWN■"SPC(10)"YOUR ANSWER?"
440 GETZ$:IFZ$<"A"ORZ$>"C"THEN440
450 PRINT"■3SHIFT CRSRUP■"SPC(10)"A"="E
460 PRINT"■CRSRDOWN■"SPC(10)"YOUR ANSWER? "Z$
470 PRINT"■CRSRDOWN■"SPC(10)"THE CORRECT ANSWER IS "E$
480 IFZ$=E$THEN500
490 GOSUB250:L=L+1:FORX=1TO2900:NEXT:GOSUB310:GOTO600
500 FORI=1TO4:POKESC,8:POKEBD,8:FORX=1TO100:NEXT
510 POKESC,7:POKEBD,7:FORX=1TO75:NEXT:NEXT
520 K=K+1:POKESC,14:POKEBD,14:GOSUB200
530 IFK>3THENGOSUB220
540 IFK>7THENGOSUB210
550 IFK<10THENPOKESC,14:GOTO570
560 GOSUB230:T5=T5-50:GOSUB200:GOSUB220:GOSUB210:GOSUB230:T5=T5+50:GOTO600
570 M=M-3:FORI=1TO8:XX=PEEK(X1):XX=XX-3:IFXX>0THEN590
580 POKEXS,0:XX=XX+255
590 POKEX1,XX:POKEX2,XX:POKEX3,XX:NEXT:GOSUB310
600 GETZ$:IFZ$<>""THEN600
610 NEXTJ:GOTO790
620 REM INITIALIZE MAIN LOOP
630 POKESC,14:POKEBD,14:POKEXS,7:POKEX1,32:POKEX2,32:POKEX3,32:M=34:K=0:L=0
640 GOSUB310:POKESE,7
650 POKE53283,7:POKE646,0
660 GOTO330
670 REM GAME BEGINS OR RESTARTS HERE
680 POKEBD,5:POKESC,5:POKE646,0:PRINT"■SHIFT CLR■6CRSRDOWN■3CRSRRIGHT■WHAT NUMBERS DO YOU WANT"
690 PRINT:PRINT"■3CRSRRIGHT■TO PRACTICE WITH? (1-9)"
700 GETN$:IFN$=""THEN 700
710 N=VAL(N$):IFN<1ORN>9THEN700
720 PRINT"■5CRSRDOWN■3CRSRRIGHT■DO YOU WANT A TIMES TABLE? (Y/N)"
```

Continued

MISSILE MATH

Continued

COMMODORE 64

```
730 GETZ$:IFZ$=""THEN730
740 IFZ$="N"THEN620
750 PRINT"■SHIFT CLR■":FORX=0TO9:PRINT:
    PRINTSPC(15)N"*"X"="N*X:NEXT
760 PRINT:PRINTSPC(10)"PRESS SPACE BAR
    TO CONTINUE..."
770 GETZ$:IFZ$=""THEN 770
780 GOTO620
790 REM SET FINISHED
800 PRINT"■SHIFT CLR■":POKESE,0
810 PRINT"■8CRSRDOWN■"SPC(8)"NUMBER RIG
    HT IS "K
820 PRINT:PRINTSPC(8)"NUMBER WRONG IS "
    L
830 PRINT:PRINTSPC(8)"PERCENT RIGHT IS
    "10*K
840 PRINT"■8CRSRDOWN■"SPC(7)"DO YOU WIS
    H TO PLAY AGAIN?■SPACE■(Y/N)"
850 GETZ$:IFZ$=""THEN850
860 IFZ$="Y"THEN670
870 END
880 REM DEFINE CONSTANTS AND VARIABLES
890 XS=53264:T5=120:FORI=0TO2:POKE2040+
    I,192:POKE53287+I,7:NEXT:X1=53248
900 POKE53249,64:POKE53251,105:POKE5325
    3,144:SE=53269:POKE53271,7:POKE5327
    7,7
910 POKE53275,7:POKE53276,7:BD=53280:PO
    KEBD,6:SC=53281:X3=53252:X2=53250
920 POKE53285,2:SO=54272:DIMT1(12),T2(1
    2),T3(12):POKESC,6:POKESE,0
930 REM PRINT **STAR-STUDDED** COMMERCI
    AL
940 PRINT"■SHIFT CLR■■CTRL WHT■■6CRSRDO
    WN■"SPC(5)"**********
    ******
950 PRINTSPC(5)"**********
    ******
960 PRINTSPC(5)"***"SPC(26)"***
970 PRINTSPC(5)"***"SPC(26)"***
980 PRINTSPC(5)"***"SPC(6)"M I S S I L E
    "SPC(6)"***
990 PRINTSPC(5)"***"SPC(26)"***
1000 PRINTSPC(5)"***"SPC(9)"M A T H "SPC(
     9)"***
1010 PRINTSPC(5)"***"SPC(26)"***
1020 PRINTSPC(5)"***"SPC(26)"***
1030 PRINTSPC(5)"**********
     **********
1040 PRINTSPC(5)"**********
     **********
1050 FORI=0TO63:READDT:POKE12288+I,DT:NE
     XT
1060 FORI=1TO12:READT1(I):NEXT
1070 FORI=1TO12:READT2(I):NEXT
1080 FORI=1TO12:READT3(I):NEXT
1090 PRINT"■5CRSRDOWN■■7CRSRRIGHT■DO YOU
     WANT INSTRUCTIONS? (Y/N)"
1100 GETZ$:IFZ$=""THEN1100
1110 IFZ$="Y"THEN1190
1120 GOTO670
1130 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,4,0
     ,0,20,0,0,0,20,0,0,84,9,85,84,42,170,
     169
1140 DATA170,170,169,42
1150 DATA170,169,9,85,84,0,0,84,0,0,20,0
     ,0,20,0,0,4,0,0,0,0,0,0,0,0,0,0,0,0
     ,0
1160 DATA33,44,0,44,50,0,50,56,67,56,44,
     0
1170 DATA33,44,0,44,50,0,50,56,0,0,44,0
1180 DATA0,67,0,0,50,0,59,56,0,0,44,0
1190 REM INSTRUCTIONS
1200 POKESC,5:POKEBD,5:POKE646,0
1210 PRINT"■SHIFT CLR■■6CRSRDOWN■1. MULT
     IPLICATION GAME"
1220 PRINT:PRINT"2. READ PROBLEM AT BOTT
     OM"
1230 PRINT:PRINT"3. EACH MISSILE HAS AN
     ANSWER"
1240 PRINT:PRINT"4. CHOOSE A, B OR C"
1250 PRINT"■11CRSRDOWN■■10CRSRRIGHT■PRES
     S SPACE BAR TO CONTINUE..."
1260 GETZ$:IFZ$=""THEN1260
1270 GOTO670
```

HCM

MISSILE MATH

IBM PC & PCjr

```
100 REM ***********
110 REM * MISSILE MATH *
120 REM ***********
130 REM BY J.C. ADELMANN
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.3.1
170 REM IBM PC CASSETTE BASIC
180 REM IBM PCjr CARTRIDGE BASIC
190 REM
200 CLS:SCREEN 1:COLOR 11,0:KEY OFF
210 P1$="MFL24T600 2GT120GT60AT120AB03DO
    2BT60GP16"
220 P2$="GGAT1200 3CO2T40BT60GP16"
230 P3$="GT120GT60AT120AB03DO2BT60GP16"
240 P4$="T400 3EO2T60AT120 03CO2T40BT60G"
250 DIM ROCKET(764):DRAW "BM100,100C1E3
    R2E2R2E2R25E10R5D8G2D6R2E2RERD10L
    HLH2L2D6F2D8L5H10L25H2L3HL2H2LH3BR1
    0P1,1":LINE (115,95)-(148,105),3,BF
    GET (100,81)-(175,118),ROCKET
260 LOCATE 9,13:PRINT "MISSILE MATH":LO
    CATE 20,5:PRINT "PRESS 'I' FOR INST
    RUCTIONS":PRINT "OR PRESS ENTER
    TO BEGIN";:IF A$="I" THEN GOSUB 590:
270 GOSUB 610:ELSE IF ASC(A$)<>13 THEN
    GOSUB 610
    GOTO 280
280 CLS:PRINT "WHAT NUMBER WOULD YOU LI
    KE TO PRACTICE WITH (1-9)?"
290 GOSUB 610:IF A$<"1" OR A$>"9" THEN
    300 ELSE S=VAL(A$)
300 PRINT:PRINT "WOULD YOU LIKE TO SEE
    THE TIMES TABLE (Y/N)?"
310 GOSUB 610:IF A$="N" THEN 350 ELSE I
    F A$<>"Y" THEN 320
320 FOR K=0 TO 9:LOCATE K+7,1:PRINT S "
    * " K "=" S*K:NEXT
330 LOCATE 23,1:PRINT "PRESS ENTER TO C
    ONTINUE":GOSUB 610
340 CLS:Q=0:E=0:R=240
350 RANDOMIZE TIMER:FOR X=1 TO 10
360 R(X)=INT(RND*10):ANS=INT(RND*3)+1:F
    OR J=1 TO X-1:IF R(X)=R(J) THEN J=1
    00
370 NEXT:IF J>11 THEN 370
380 B=R(X)*S
390 GOSUB 600:LINE (1,140)-(319,199),1,
    BF:LOCATE 5,INT(R/8)+3:IF ANS=1 THE
    N PRINT USING "\\###";A;:INT(RND*99)
400 LOCATE 10,INT(R/8)+3:IF ANS=2 THEN
    PRINT USING "\\###";B;:INT(RND*99)
410 LOCATE 15,INT(R/8)+3:IF ANS=3 THEN
    PRINT USING "\\###";C;:INT(RND*99)
420 LOCATE 15,INT(R/8)+3:IF ANS=3 THEN
    PRINT USING "\\###";C;:B ELSE PRIN
    T USING "\\###";C;:INT(RND*99)
430 LOCATE 20,17:PRINT S"x"R(X)
440 LOCATE 22,13:PRINT "YOUR ANSWER IS?
    ":GOSUB 610:IF A$<"A" OR A$>"C" THE
    N SOUND 110,2:GOTO 440
450 IF ASC(A$)-64=ANS THEN 480
460 LOCATE 23,13:PRINT "THAT IS WRONG!!
    ":SOUND 220,10:LOCATE 24,10:PRINT "
    THE RIGHT ANSWER IS ";CHR$(ANS+64);
470 FOR TD=1 TO 2000:NEXT:GOTO 540
480 LOCATE 23,1:PRINT "*****GOOD WORK
    ---THATS RIGHT*****";
    YOUR ANSWER WAS ";CHR$(ANS+64);"
    *******";
490 IF Q<3 THEN PLAY "XP1$;":GOTO 530
500 IF Q<6 THEN PLAY "XP1$;XP2$;":GOTO
    530
510 IF Q<9 THEN PLAY "XP1$;XP2$;XP3$;":
    GOTO 530
520 PLAY "XP1$;XP2$;XP3$;XP4$;"
530 Q=Q+1:R=R-16
540 NEXT
550 IF Q<10 THEN 570 ELSE FOR Z=0 TO 24
    0 STEP 80:FOR R=240 TO Z STEP -8:GO
    SUB 600:NEXT:NEXT
560 A$="L16T600 2F#O3DT120F#DEF#GET60DT1
    20F#DEF#GEDF#AO3DC#O2BAGF#GABAGF#ET
    60DT120F#DEF#GET60DT120F#DEF#GEDF#A
    O3DC#O2BAGF#GEF#T60D":PLAY "XA$;XA$
    ;"
570 LINE (0,140)-(319,199),0,BF:LOCATE
    20,1:PRINT "NUMBER RIGHT IS:";Q:PRI
    NT "NUMBER WRONG IS:";10-Q:PRINT "PE
    RCENT RIGHT IS:";(Q/10)*100:PRINT
    "WOULD YOU LIKE TO PLAY AGAIN (Y/
    N)?";
580 GOSUB 610:IF A$="N" THEN END ELSE I
    F A$<>"Y" THEN 580 ELSE GOTO 290
590 CLS:LOCATE 3,10:PRINT "MULTIPLICATI
    ON GAME":PRINT:PRINT "EACH MISSILE C
    ONTAINS AN ANSWER.":PRINT "YOU MUST
    CHOOSE THE ANSWER WHICH FITS THE
    PROBLEM SHOWN BELOW.":PRINT:PRINT
    "PRESS EITHER A B OR C FOR THE ANSW
    ER.":RETURN
600 PUT (R,16),ROCKET,PSET:PUT (R,56),R
    OCKET,PSET:PUT (R,96),ROCKET,PSET:R
    ETURN
610 A$=INKEY$:IF LEN(A$)=0 THEN 610 ELS
    E RETURN
```

HCM

MISSILE MATH

Continued

TI-99/4A

```
100 REM *********************
110 REM *  MISSILE MATH  *
120 REM *********************
130 REM BY J.C. ADELMANN
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.3.1
170 REM TI-BASIC
180 REM TI-EXTENDED BASIC
190 CALL CLEAR
200 DIM MS1(22),MS2(12)
210 RESTORE 2000
220 FOR I=1 TO 22
230 READ MS1(I)
240 NEXT I
250 FOR I=1 TO 12
260 READ MS2(I)
270 NEXT I
280 PRINT TAB(9);"MISSILE MATH"
290 PRINT TAB(3);"DO YOU NEED INSTRUCTI
    ONS?  (Y/N)"
300 INPUT R$
310 IF R$="Y" THEN 1800
320 CALL CLEAR
330 GOSUB 1060
340 CALL SCREEN(11)
350 INPUT "WHAT NUMBER DO YOU WANT TO
    PRACTICE WITH? (1-9)":S
360 IF (S<1)+(S>9)THEN 350
370 Y$=STR$(S)
380 INPUT "DO YOU WANT A TIMES TABLE?
    (Y/N) ":R$
390 IF (R$="N")+(R$="n")THEN 490
400 CALL CLEAR
410 FOR K=0 TO 9
420 PRINT "  ";S;" * ";K;" = ";S*K
430 PRINT
440 NEXT K
450 PRINT "  PRESS ENTER TO CONTINUE"
460 CALL KEY(0,RET,STA)
470 IF STA=0 THEN 460
480 IF RET<>13 THEN 460
490 REM DOMATH2
500 CALL CLEAR
510 Q=0
520 E=0
530 CALL CLEAR
540 CALL SCREEN(6)
550 CALL HCHAR(18,1,106,224)
560 FOR X=1 TO 10
570 RANDOMIZE
580 R(X)=INT(10*RND)
590 T=0
600 ANS=INT(3*RND)+1
610 FOR J=1 TO X-1
620 IF R(X)=R(J)THEN 640
630 GOTO 650
640 T=1
650 NEXT J
660 IF T=1 THEN 580
670 W$=STR$(R(X))
680 B=R(X)*S
690 C$=STR$(B)
700 L=LEN(STR$(B))
710 U=ASC(W$)
720 V=ASC(Y$)
730 GOSUB 1270
740 CALL HCHAR(21,19,106,2)
750 CALL HCHAR(19,24,106)
760 FOR I=6 TO 28
770 CALL HCHAR(22,I,106)
780 NEXT I
790 CALL HCHAR(21,15,U)
800 CALL HCHAR(21,16,42)
810 CALL HCHAR(21,17,V)
820 CALL HCHAR(21,18,61)
830 E=0
840 FOR I=11 TO 22
850 CALL HCHAR(19,I,MS2(I-10))
860 NEXT I
870 CALL KEY(3,RET,STA)
880 IF (RET<65)+(RET>67)THEN 870
890 CALL HCHAR(19,24,RET)
900 IF (RET-64)<>ANS THEN 920
910 E=1
920 GOSUB 1920
930 Q=Q+E
940 NEXT X
950 FOR DELAY=1 TO 500
960 NEXT DELAY
970 CALL SCREEN(11)
980 CALL CLEAR
990 PRINT "NUMBER RIGHT IS";Q
1000 PRINT "NUMBER WRONG IS";10-Q
1010 PRINT "PERCENT RIGHT IS";(Q/10)*100
1020 INPUT "DO YOU WISH TO PLAY AGAIN?
     (Y/N) ":D$
1030 IF (D$="Y")+(D$="y")THEN 330
1040 CALL CLEAR
1050 END
1060 CALL COLOR(3,2,11)
1070 CALL COLOR(4,2,11)
1080 CALL COLOR(7,2,11)
1090 CALL COLOR(2,2,11)
1100 CALL COLOR(5,2,11)
1110 CALL COLOR(6,2,11)
1120 CALL COLOR(8,2,11)
1130 CALL CLEAR
1140 CALL SCREEN(2)
1150 RESTORE 1780
1160 FOR I=97 TO 102
1170 READ A$
1180 CALL CHAR(I,A$)
1190 NEXT I
1200 FOR I=104 TO 106
1210 READ A$
1220 CALL CHAR(I,A$)
1230 NEXT I
1240 CALL COLOR(9,10,1)
1250 CALL COLOR(10,11,1)
1260 RETURN
1270 FOR G=0 TO 2
1280 IF G<>ANS-1 THEN 1360
1290 IF L<>1 THEN 1330
1300 N=ASC(C$)
1310 M=106
1320 GOTO 1420
1330 M=ASC(SEG$(C$,1,1))
1340 N=ASC(SEG$(C$,2,1))
1350 GOTO 1420
1360 RANDOMIZE
1370 M=INT(10*RND)+48
1380 IF M<>48 THEN 1400
1390 M=106
1400 N=INT(10*RND)+48
1410 IF M*10+N=B THEN 1370
1420 IF E=1 THEN 1450
1430 CALL SOUND(200,110,0)
1440 GOTO 1580
1450 CALL SOUND(150,262,15,330,15,392,15
     )
1460 CALL SCREEN(5)
1470 CALL SOUND(150,262,15,330,15,392,15
     )
1480 CALL SCREEN(6)
1490 CALL SOUND(150,262,15,330,15,392,15
     )
1500 CALL SCREEN(7)
1510 CALL SOUND(150,294,15,349,15,440,15
     )
1520 CALL SCREEN(8)
1530 CALL SOUND(150,262,15,330,15,392,15
     )
1540 CALL SCREEN(9)
1550 CALL SOUND(150,294,15,349,15,440,15
     )
1560 CALL SCREEN(6)
1570 CALL SOUND(200,330,15,415,15,494,15
     )
1580 CALL HCHAR(G*4+3,20-Q,97)
1590 CALL HCHAR(G*4+3,20-Q+1,32)
1600 CALL HCHAR(G*4+4,17-Q,104)
1610 CALL HCHAR(G*4+4,18-Q,99)
1620 CALL HCHAR(G*4+4,19-Q,106)
1630 CALL HCHAR(G*4+4,20-Q,G+65)
1640 CALL HCHAR(G*4+4,21-Q,101)
1650 CALL HCHAR(G*4+4,21-Q+1,32)
1660 CALL HCHAR(G*4+5,17-Q,105)
1670 CALL HCHAR(G*4+5,18-Q,100)
1680 CALL HCHAR(G*4+5,19-Q,M)
1690 CALL HCHAR(G*4+5,20-Q,N)
1700 CALL HCHAR(G*4+5,21-Q,102)
1710 CALL HCHAR(G*4+5,21-Q+1,32)
1720 CALL HCHAR(G*4+6,20-Q,98)
1730 CALL HCHAR(G*4+6,20-Q+1,32)
1740 FOR DELAY=1 TO 100
1750 NEXT DELAY
1760 NEXT G
1770 RETURN
1780 DATA "0103070F1F3F7FFF","FF7F3F1F0F
     0703010","03070F1F3F7FFFFF","FFFF7F3
     F1F0F0703","0000000020606E0E0"
1790 DATA "E0E06020","0000003838101FF",
     "FF1010383838","FFFFFFFFFFFFFFFF"
1800 REM
1810 REM *INSTRUCTIONS*
1820 REM
1830 CALL CLEAR
1840 PRINT "MULTIPLICATION GAME"
1850 PRINT
1860 PRINT "EACH MISSILE HAS AN ANSWER
     CHOOSE A B OR C"
1870 PRINT
1880 PRINT "PRESS ANY KEY TO CONTINUE"
1890 CALL KEY(0,KY,S)
1900 IF S=0 THEN 1890
1910 GOTO 320
1920 IF (LEN(C$)=1)THEN 1940 ELSE 1930
1930 CALL HCHAR(21,20,ASC(SEG$(C$,2,1)))
1940 CALL HCHAR(21,19,ASC(SEG$(C$,1,1)))
1950 FOR I=6 TO 27
1960 CALL HCHAR(22,I,MS1(I-5))
1970 NEXT I
1980 CALL HCHAR(22,28,ANS+64)
1990 RETURN
2000 DATA 84,72,69,106,67,79,82,82,69,67
     ,84,106,65,78,83,87,69,82,106,73,83
     ,106
2010 DATA 89,79,85,82,106,65,78,83,87,69
     ,82,63
```

HCM

MISSILE MATH

VIC-20

```
100 REM  ***************
110 REM  *  MISSILE MATH  *
120 REM  ***************
130 REM  BY J.C. ADELMANN
140 REM  AND THE HCM STAFF
150 REM  HOME COMPUTER MAGAZINE
160 REM  VERSION 4.3.1
170 REM  V-20 BASIC
180 POKE36879,13
190 PRINT"■SHIFT CLR■":PRINT"■CTRL GRN■
    ■6CRSRDOWN■■5CRSRRIGHT■MISSILE MATH
    ■"
200 PRINT"■2CRSRDOWN■    DO YOU NEED":
    PRINT"     INSTRUCTIONS?"
210 GETI$:IFI$=""THEN 210
220 IFI$<>"N"THENGOSUB 1080
230 NC=233:NC2=95:RS=96:F=123:V=126:SP=
    11
240 POKE36879,13
250 PRINT"■SHIFT CLR■■CTRL GRN■■4CRSRDO
    WN■WHAT NUMBER DO YOU WANT TO PR
    ACTICE":PRINT"WITH (1-9)?"
260 GETPN$:IFPN$=""THEN 260
270 PN=VAL(PN$)
280 IFPN<1ORPN>9THEN 250
290 Y$=STR$(PN)
300 PRINT"■2CRSRDOWN■DO YOU WANT A":PRI
    NT"TIMES TABLE (Y/N)?"
310 GETTT$:IFTT$=""THEN 310
320 IFTT$="N"THEN 380
330 PRINT"■SHIFT CLR■":FORK=0TO9
340 PRINT"  ";PN;" * ";K;" = ";PN*K:PR
    INT:NEXT
350 PRINT"■CRSRDOWN■HIT ■CTRL RVSON■RET
    URN■CTRL RVSOFF■ TO CONTINUE";
360 GETA$:IFA$=""THEN 360
370 REM **DO MATH 2**
380 PRINT"■SHIFT CLR■":POKE36879,108
390 FORZ=0TO8
400 T=INT(RND(1)*10)
410 FORN=0TO9
420 IFR(N)=TTHEN 400
430 NEXTN
440 R(Z)=T:NEXTZ
450 FOR Z=0TO9
460 AN=R(Z)*PN
470 O1=INT(RND(1)*10)*INT(RND(1)*10):IF
    O1=ANTHEN 470
480 O2=INT(RND(1)*10)*INT(RND(1)*10):IF
    O2=O1ORO2=ANTHEN 480
490 W=INT(RND(1)*3)+1
500 B(W)=AN
510 IFB(1)=ANTHENB(2)=O1:B(3)=O2:CA$="A
    "
520 IFB(2)=ANTHENB(1)=O1:B(3)=O2:CA$="B
    "
530 IFB(3)=ANTHENB(1)=O1:B(2)=O2:CA$="C
    "
540 GOSUB 770
550 POKE198,0
560 GETA$:IFA$=""THEN 560
570 IFA$<>"A"ANDA$<>"B"ANDA$<>"C"THEN
    560
580 PRINT A$
590 IFA$=CA$THENPRINT"■CRSRDOWN■■CTRL R
    VSON■THAT IS CORRECT!!":GOSUB 980:
    ::CK=CK+1:SP=SP-1:GOTO 640
600 PRINT"■CRSRDOWN■■CTRL RVSON■THAT'S
    NOT RIGHT!":GOSUB 1050:::WR=WR+1
610 FORD=1TO999:NEXTD
620 PRINT"■2SHIFT CRSRUP■■CTRL RVSON■TH
    E RIGHT ANSWER WAS ";CA$;
630 FORD=1TO1999:NEXTD
640 PRINT"■HOME■":NEXTZ
650 POKE36879,15
660 IFCK=10THENGOSUB 980
670 PRINT"■SHIFT CLR■■2CRSRDOWN■YOU GOT
    ";CK;"RIGHT"
680 PRINT"AND";WR;"WRONG OUT"
690 PRINT"OF 10 PROBLEMS."
700 PRINT "THAT'S ";INT(CK*1000/100);"
    PERCENT"
710 IFCK=10THENPRINT"IS THIS TOO
    EASY FOR YOU?"
720 PRINT"■5CRSRDOWN■WOULD YOU LIKE":PR
    INT"TO TRY AGAIN?"
730 GET A$:IFA$=""THEN 730
740 IFA$<>"Y"THEN 760
750 RUN 230
760 PRINT"THANKS FOR PLAYING!":END
770 O$=""
780 FORLN=1TOSP
790 O$=O$+" "
800 NEXTLN
810 PRINTO$;"■CTRL RVSON■■CTRL RED■■
    SHIFT £■■CTRL RVSOFF■ "
820 PRINTO$;"■CTRL RVSON■■CTRL RED■■SHI
    FT £■■CTRL GRN■■A ■CTRL CYAN■■CTRL
    RVSOFF■■CTRL BLK■■CMDR F■";B(1);"■S
    HIFT CRSRLEFT■ "
830 PRINTO$;"■CTRL RED■■CMDR ·■■CTRL GR
    N■■CTRL RVSON■ ■CTRL CYAN■■CTRL R
    VSOFF■■CTRL BLK■■CMDR V■ "
840 PRINTO$;"■CTRL RED■■CMDR ·■ "
850 PRINTO$;"■CTRL RVSON■■CTRL RED■■
    SHIFT £■■CTRL RVSOFF■ "
860 PRINTO$;"■CTRL RVSON■■CTRL RED■■SHI
    FT £■■CTRL GRN■■B ■CTRL CYAN■■CTRL
    RVSOFF■■CTRL BLK■■CMDR F■";B(2);"■S
    HIFT CRSRLEFT■ "
870 PRINTO$;"■CTRL RED■■CMDR ·■■CTRL GR
    N■■CTRL RVSON■ ■CTRL CYAN■■CTRL R
    VSOFF■■CTRL BLK■■CMDR V■ "
880 PRINTO$;"■CTRL RED■■CMDR ·■ "
890 PRINTO$;"■CTRL RVSON■■CTRL RED■■
    SHIFT £■■CTRL RVSOFF■ "
900 PRINTO$;"■CTRL RVSON■■CTRL RED■■SHI
    FT £■■CTRL GRN■■C ■CTRL CYAN■■CTRL
    RVSOFF■■CTRL BLK■■CMDR F■";B(3);"■S
    HIFT CRSRLEFT■ "
910 PRINTO$;"■CTRL RED■■CMDR ·■■CTRL GR
    N■■CTRL RVSON■ ■CTRL CYAN■■CTRL R
    VSOFF■■CTRL BLK■■CMDR V■ "
920 PRINTO$;"■CTRL RED■■CMDR ·■ "
930 FORCB=8010TO8141:POKECB,160:NEXTCB
940 FORCC=38730TO38861:POKECC,7:NEXTCC
950 PRINT"■HOME■■CTRL YEL■■16CRSRDOWN■■
    4CRSRRIGHT■■CTRL RVSON■WHAT IS";PN;
    "*";R(Z)
960 PRINT"■CRSRDOWN■■CTRL RVSON■CHOOSE
    A, B, OR C: ";
970 RETURN
980 RESTORE:POKE36878,15
990 READA,B
1000 IFA=-1THENPOKE36878,0:POKE36876,0:R
    ETURN
1010 POKE36876,A
1020 FORTD=1TOB:NEXT:
1030 GOTO 990
1040 RETURN
1050 POKE36878,15
1060 POKE36874,200
1070 FORTD=1TO500:NEXT:POKE36874,0:POKE3
    6878,0:RETURN
1080 PRINT"■SHIFT CLR■■4CRSRDOWN■ MULTIP
    LICATION GAME"
1090 PRINT"■CRSRDOWN■EACH MISSILE HAS AN
    ANSWER TO A PROBLEM."
1100 PRINT"■CRSRDOWN■CHOOSE THE CORRECT
    ANSWER A B OR C TO MOVE THE M
    ISSILES."
1110 PRINT"■CRSRDOWN■HIT ■CTRL RVSON■RET
    URN■CTRL RVSOFF■ TO CONTINUE";
1120 GETA$:IFA$=""THEN 1120
1130 RETURN
1140 DATA195,100,0,20,201,100,0,20,207,1
    00,0,20,201,100,0,20,195,100,0,20,2
    01,100,0,20
1150 DATA207,100,0,150,195,200,0,100,195
    ,200,-1,-1
```

HCM

ROBOCHASE

APPLE II Family

```
100 REM  **************
110 REM  *  ROBOCHASE  *
120 REM  **************
130 REM  BY GREG VAUGHAN AND HCM STAFF
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.3.1
160 REM  APPLE II SERIES APPLESOFT
170 IF PEEK(103)<>1 OR PEEK(104
    )<>64 THEN POKE 103,1: POKE 10
    4,64: POKE 16384,0: POKE 16385,0: P
    OKE 16386,0: PRINT CHR$(13) CHR$
    (4)"RUN ROBOCHASE"
180 GOSUB 1530
190 GOSUB 1600
200 TEXT : HOME : GOSUB 1020
210 GOSUB 980: IF ST = 0 THEN 210
220 POKE 863,KY
230 REM  START NEW GAME******
240 CLEAR : DIM SL%(27,17),RL%(10,1),ML
    %(1),LS%(1)
250 J$ = CHR$ ( PEEK (863))
260 R = 10: D = 1: LV = 1: BA = 24: DI = 6
270 REM  RE-ENTRY POINT*****
280 GOSUB 1170
290 GOSUB 1950
300 REM CHECK FOR MOVE ******
310 IF J$ <> "J" THEN 420
320 FB = PEEK (-16287)
330 IF FB > 127 AND D > 0 THEN GOSUB 1
    730: GOTO 620
340 M0% = PDL (0): FOR DE = 1 TO 10: N
    EXT DE
350 M1% = PDL (1)
360 IF M0% < 100 THEN MX = -1: GOTO 3
    80
370 IF M0% > 250 THEN MX = 1
380 IF M1% < 100 THEN MY = -1: GOTO 4
    90
390 IF M1% > 250 THEN MY = 1: GOTO 490
400 IF MY = 0 AND MX = 0 THEN 620
```

Continued

Continued

APPLE II Family

PROGRAM LISTING

```
410 GOTO 490
420 POKE -16368,0: FOR DE = 1 TO 100:
    NEXT DE: GOSUB 980: IF ST = 0 THEN
    620
430 IF KY = 32 AND D > 0 THEN GOSUB 17
    30: GOTO 620
440 IF KY = 81 THEN MY = -1: GOTO 490
450 IF KY = 65 THEN MY = +1: GOTO 490
460 IF KY = 79 THEN MX = -1: GOTO 490
470 IF KY = 80 THEN MX = +1: GOTO 490
480 GOTO 620
490 GOSUB 1150
500 IF MX = 0 THEN 520
510 ML%(0) = ML%(0) + MX:MX = 0: IF ML%
    (0) < 0 OR ML%(0) > 27 THEN 1700
520 IF MY = 0 THEN 540
530 ML%(1) = ML%(1) + MY:MY = 0: IF ML%
    (1) < 0 OR ML%(1) > 17 THEN 1700
540 IF SL%(ML%(0),ML%(1)) = 0 THEN 600
550 IF SL%(ML%(0),ML%(1)) = 14 THEN D =
    D + 1:SC = SC + 75:CR = 4:OB = 3:X
    = ML%(0):Y = ML%(1): GOSUB 960: GO
    TO 600
560 IF SL%(ML%(0),ML%(1)) = 13 THEN D =
    D + 1:SC = SC + 100:CR = 4:OB = 2:
    X = ML%(0):Y = ML%(1):LS%(0) = -1
    : GOSUB 960: GOTO 600
570 IF SL%(ML%(0),ML%(1)) = 11 THEN 170
    0
580 IF SL%(ML%(0),ML%(1)) = 12 THEN 170
    0
590 GOTO 1790
600 SL%(ML%(0),ML%(1)) = 15:X = ML%(0):
    Y = ML%(1):CR = 7:OB = 4: GOSUB 960
    :PT = 150:LN = 1: GOSUB 1010
610 REM MOVE ROBOTS **********
620 FOR RN = 1 TO 10
630 IF R = 0 THEN 1840
640 IF RL%(RN,0) = -1 THEN 830
650 IF RL%(RN,0) = ML%(0) THEN MX = 0:
    GOTO 680
660 IF RL%(RN,0) > ML%(0) THEN MX = -1
    : GOTO 680
670 MX = 1
680 IF RL%(RN,1) = ML%(1) THEN MY = 0:
    GOTO 710
690 IF RL%(RN,1) > ML%(1) THEN MY = -1
    : GOTO 710
700 MY = 1
710 IF MX = 0 THEN 760
720 IF MY = 0 THEN 760
730 HM = INT (RND (1) * 5) + 1: IF HM
    < LV THEN 760
740 IF INT (RND (1) + .5) = 0 THEN MY
    = 0: GOTO 760
750 MX = 0
760 X = RL%(RN,0):Y = RL%(RN,1):CR = 4:
    OB = 1: GOSUB 960:SL%(X,Y) = 0
770 X = X + MX:RL%(RN,0) = X:Y = Y + MY
    :RL%(RN,1) = Y:MX = 0:MY = 0
780 IF SL%(X,Y) = 0 THEN 820
790 IF SL%(X,Y) = 15 THEN 1790
800 IF SL%(X,Y) > 10 THEN RL%(RN,0) =
    -1: SC = SC + 25:R = R - 1: GOTO 83
    0
810 RL%(SL%(X,Y),0) = -1:RL%(RN,0) =
    -1:SL%(X,Y) = 12:CR = 7:OB = 5: GO
    SUB 960:SC = SC + 50:R = R - 2: GO
    TO 830
820 RL%(RN,0) = X:RL%(RN,1) = Y:SL%(X,Y
    ) = RN:CR = 6:OB = 1: GOSUB 960:PT
    = 4:L = 2: GOSUB 1010
830 NEXT RN
840 REM CHECK FOR SPUNKY AND MOVE*****
850 IF LV < 4 OR LS%(0) = -1 THEN 940
860 X = LS%(0):Y = LS%(1):CR = 4:OB = 2
    : GOSUB 960:SL%(X,Y) = 0
870 MX = INT (RND (1) * 3) - 1:MY =
    INT (RND (1) * 3) - 1:X = LS%(0) +
    MX:Y = LS%(1) + MY: IF X < 0 OR X
    > 27 OR Y < 0 OR Y > 17 THEN 870
880 IF SL%(X,Y) = 0 THEN 930
890 IF SL%(X,Y) < 11 THEN OB = 1:CR = 4
    :RL%(SL%(X,Y),0) = -1:R = R - 1::
    GOSUB 960: GOTO 930
900 IF SL%(X,Y) < 13 THEN CR = 4: GOSUB
    1940: GOTO 930
910 IF SL%(X,Y) = 14 THEN CR = 4:OB = 3
    : GOSUB 960: GOTO 930
920 LS%(0) = -1:SC = SC + 100:D = D +
    1: GOTO 940
930 CR = 7:OB = 2: GOSUB 960:SL%(X,Y) =
    13:LS%(0) = X:LS%(1) = Y:MX = 0:MY
    = 0
940 GOTO 290
950 REM DRAW OBJECT*******
960 HCOLOR= CR: DRAW OB AT (X * 8 + 28)
    ,(Y * 8 + 7): RETURN
970 REM READ KEYBOARD****
980 KY = PEEK (-16384): IF KY > 127
    THEN ST = 1:KY = KY - 128: POKE -
    16368,0: RETURN
990 ST = 0: POKE -16368,0: RETURN
1000 REM MAKE A NOISE******
1010 POKE 865,PT: POKE 864,LN: CALL 866:
     RETURN
1020 REM TITLE SCREEN******
1030 VTAB 10: HTAB 13: PRINT "*********
     ***": HTAB 13: PRINT "*
     *": HTAB 13: PRINT "* ROBOCHASE *":
     HTAB 13: PRINT "*
     *": HTAB 13: PRINT "**********"
1040 VTAB 23: HTAB 10: PRINT "PRESS J FO
     R JOYSTICKS": HTAB 10: PRINT "PRESS
     K FOR KEYBOARD": RETURN
1050 REM PLAY AGAIN?******
1060 PRINT "YOUR FINAL SCORE WAS ";SC
1070 GOSUB 1720
1080 PRINT " LIKE TO PLAY AGAIN? (Y/N)";:
     POKE -16368,0: FOR DE = 1 TO 100
     : NEXT : GOSUB 980
1090 IF ST = 0 THEN GOSUB 980: GOTO 109
     0
1100 IF KY = 89 THEN 240
1110 HOME : TEXT : END
1120 REM GET RANDOM******
1130 X = INT ( RND (1) * 28):Y = INT (
     RND (1) * 18): RETURN
1140 REM DELETE PERSON*****
1150 X = ML%(0):Y = ML%(1):CR = 4:OB = 4
     : GOSUB 960:SL%(X,Y) = 0: RETURN
1160 REM DRAW FIRST SCREEN******
1170 HGR : HOME : VTAB 21: HTAB 16: PRIN
     T "ROBOCHASE": PRINT "SCORE:";: HTA
     B 20: PRINT "LEVEL:": PRINT "TELEPO
     RTS:"
1180 VTAB 22: PRINT "SCORE:";: HTAB 20:
     PRINT "LEVEL:"
1190 VTAB 23: PRINT "TELEPORTS:"
1200 GOSUB 1950
1210 HCOLOR= 2
1220 FOR N = 0 TO 6: HPLOT 21,N TO 258,N
     : NEXT N
1230 FOR N = 21 TO 26: HPLOT N,7 TO N,15
     3: NEXT N
1240 FOR N = 252 TO 258: HPLOT N,7 TO N,
     153: NEXT N
1250 FOR N = 151 TO 157: HPLOT 21,N TO 2
     58,N: NEXT N
1260 HCOLOR= 5
1270 FOR I = 1 TO BA
1280 GOSUB 1130
1290 IF SL%(X,Y) > 0 THEN 1280
1300 SL%(X,Y) = 11
1310 GOSUB 1940
1320 NEXT I
1330 ROT= 0: SCALE= 1
1340 FOR K = 1 TO DI
1350 GOSUB 1130
1360 IF SL%(X,Y) > 0 THEN 1350
1370 OB = 3:CR = 7: GOSUB 960:SL%(X,Y) =
     14
1380 NEXT K
1390 FOR L = 1 TO R
1400 GOSUB 1130
1410 IF SL%(X,Y) > 0 THEN 1400
1420 OB = 1:CR = 6: GOSUB 960:SL%(X,Y) =
     L:RL%(L,0) = X:RL%(L,1) = Y
1430 NEXT L
1440 IF LV < 4 THEN 1470
1450 GOSUB 1130: IF SL%(X,Y) > 0 THEN 14
     50
1460 OB = 2:CR = 7: GOSUB 960:SL%(X,Y) =
     13:LS%(0) = X:LS%(1) = Y
1470 GOSUB 1130
1480 IF SL%(X,Y) > 0 THEN 1470
1490 FOR M = 1 TO 7: IF M / 2 = INT (M
     / 2) THEN CR = 4:PT = 134:LN = 2: G
     OSUB 1010: GOTO 1510
1500 CR = 7:PT = 85:LN = 4: GOSUB 1010
1510 OB = 4: GOSUB 960:SL%(X,Y) = 15:ML%
     (0) = X:ML%(1) = Y
1520 NEXT M: RETURN
1530 REM SPEAKER MACHINE LANGUAGE***
1540 FOR I = 864 TO 891
1550 READ J
1560 POKE I,J
1570 NEXT I
1580 RETURN
1590 DATA 0,115,172,97,3,174,97,3,232,2
     08,253,169,4,32,168,252,173,48,192,
     136,208,239,206,96,3,208,231,96
1600 REM LOAD SHAPE TABLE*****
1610 POKE 232,0: POKE 233,96
1620 FOR I = 24576 TO 24679: READ J: POK
     E I,J: NEXT I
1630 DATA 5,0,12,0,30,0,50,0,68,0,84,0,
     73,53,55
1640 DATA 63,54,54,46,36,36,45,53,54,46
     ,36,36,60,63,0,21
1650 DATA 190,53,53,118,45,32,63,44,45,
     5,56,31,63,40,45,37
1660 DATA 44,6,0,146,14,36,12,54,54,14,
     36,36,36,21,54,54
1670 DATA 5,32,172,6,0,145,14,37,32,53,
     110,21,31,55,21,46
1680 DATA 222,27,100,5,0,73,53,55,55,62
     ,54,46,36,37,44,54,62,46,45,60,46,6
     0,36,39,0
1690 RETURN
1700 REM PERSON IS ELECTRIFIED****
1710 HOME : VTAB 21: PRINT "YOU HAVE BEE
     N ELECTRIFIED ON LEVEL ";LV: GOTO 1
     060
```

Continued

ROBOCHASE

Continued

APPLE II Family

```
1720 PT = 24:LN = 8: GOSUB 1010:PT = 52:LN = 4: GOSUB 1010:PT = 63:LN = 6: GOSUB 1010:PT = 24:LN = 8: GOSUB 1010:PT = 97:LN = 10: GOSUB 1010: RETURN
1730 REM TELEPORT PERSON***
1740 GOSUB 1150
1750 FOR I = 1 TO 3: FOR J = 1 TO 3:PT = 250 + J:LN = 3: GOSUB 1010: NEXT J: NEXT I
1760 GOSUB 1130
1770 IF SL%(X,Y) > 0 THEN 1760
1780 ML%(0) = X:ML%(1) = Y:SL%(X,Y) = 15:CR = 7:OB = 4: GOSUB 960:D = D - 1:SC = SC - 50: RETURN
1790 REM PERSON CAPTURED BY ROBOT****
1800 HCOLOR= 6
1810 X = ML%(0):Y = ML%(1): HPLOT (X * 8 + 28),(Y * 8 + 7) TO (X * 8 + 36),(Y * 8 + 15)
1820 HOME : VTAB 22: PRINT "YOU HAVE BEEN CAPTURED BY A ROBOT": PRINT "ON LEVEL ";LV: GOTO 1060
1830 GOTO 1060
1840 REM NEXT LEVEL *****
1850 HOME : VTAB 21: PRINT "YOU HAVE COMPLETED LEVEL ";LV: PRINT "YOUR SCORE IS ";SC
1860 FOR N = 1 TO 2:PT = 85:LN = 4: GOSUB 1010: NEXT N:PT = 151:LN = 8: GOSUB 1010
1870 LV = LV + 1:R = 10:DI = DI - 1:BA = BA - 4
1880 IF D = 1 THEN 1900
1890 D = INT (D / 2)
1900 IF DI < 3 THEN DI = 3
1910 IF BA < 12 THEN BA = 12
1920 FOR I = 0 TO 27: FOR J = 0 TO 17:SL%(I,J) = 0: NEXT J: NEXT I
1930 GOTO 280
1940 FOR J = (X * 8 + 28) TO (X * 8 + 34): HPLOT J,(Y * 8 + 7) TO J,(Y * 8 + 13): NEXT J: RETURN
1950 VTAB 22: HTAB 7: PRINT SC;" ";: HTAB 26: PRINT LV;" "
1960 VTAB 23: HTAB 11: PRINT D;" ": RETURN
```

HCM

ROBOCHASE

COMMODORE 64

```
100 REM *************
110 REM * ROBOCHASE *
120 REM *************
130 REM BY GREG VAUGHAN
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.3.1
170 REM C-64 BASIC
180 PRINT "{SHIFT CLR}{2 CRSR DOWN}WORKING . . .":DIM BT(10,2),MN(1,2),SK(1,2)
190 POKE 52,56:POKE 56,56:CLR
200 POKE 56334,PEEK(56334)AND254
210 POKE1,PEEK(1)AND251
220 FOR I=0 TO 511:POKE I+14336,PEEK(I+53248):NEXT
230 POKE1,PEEK(1)OR4
240 POKE 56334,PEEK(56334)OR1
250 POKE 53272,(PEEK(53272)AND240)OR14
260 FOR I=14848 TO 16384:POKE I,255:NEXT
270 FOR I=14848 TO 14847+(29*8):READ A:POKE I,A:NEXT
280 DATA 120,124,110,120,108,108,108,110,126,126,102,102,102,102,126,126
290 DATA 124,126,102,124,124,102,126,124,126,126,102,96,96,102,126,126
300 DATA 102,102,102,126,126,102,102,102,60,126,102,102,126,126,102,102
310 DATA 62,62,50,56,28,76,124,124,126,126,96,124,124,96,126,126
320 DATA 24,24,255,189,189,60,36,102,129,66,126,187,126,60,36,24
330 DATA 16,56,124,254,126,56,16,0,255,255,255,255,255,255,255,255
340 DATA 16,56,120,120,124,254,254,0,56,56,74,61,8,21,38,32
350 DATA 255,255,255,255,255,248,248,248,255,255,255,255,255,31,31,31
360 DATA 128,192,224,240,248,252,254,255,255,255,255,255,0,0,0,0
370 DATA 255,255,255,255,255,0,0,0,248,248,248,255,255,255,255,255
380 DATA 31,31,31,255,255,255,255,255,254,252,248,240,240,248,252,254
390 DATA 0,0,0,255,255,255,0,0,255,127,63,31,15,7,3,1
400 DATA 0,0,255,255,255,255,0,0,15,15,15,15,15,15,15,15
410 DATA 0,0,0,0,255,255,255,255
420 DATA 1,3,7,15,31,63,127,255,255,254,252,248,240,224,192,128
430 PRINT"{SHIFT CLR}":POKE 53281,12
440 PRINT "{20 CRSR DOWN}{CTRL WHT}"
450 PRINT "{2 CRSR RIGHT}< PRESS I FOR INSTRUCTIONS >"
460 PRINT "{2 CRSR RIGHT}< HIT ANY OTHER KEY TO BEGIN >"
470 R=INT(RND(0)*6)+1:FOR I=1TO100:NEXT
480 ON R GOSUB 2020, 2030, 2040, 2050, 2060, 2070
490 GOSUB 1970
500 A=PEEK(197):IF A=64 THEN 470
510 IF A<>33 THEN 950
520 PRINT "{CTRL WHT}{SHIFT CLR}{2 CRSR DOWN}{7 CRSR RIGHT}YOU HAVE BEEN IMPRISONED"
530 PRINT "{2 CRSR RIGHT}IN A ROOM BY A MAD SCIENTIST AND YOU"
540 PRINT "{2 CRSR RIGHT}MUST AVOID THE BLUE ROBOTS AND CAUSE"
550 PRINT "{2 CRSR RIGHT}THEM TO CRASH INTO THE RED BARRIERS."
560 PRINT:PRINT:PRINT
570 PRINT "{2 CRSR RIGHT}        < HIT ANY KEY TO CONTINUE >"
580 FOR I=1TO1000:NEXT
590 A=PEEK(197):IF A=64 THEN 590
600 PRINT "{SHIFT CLR}{3 CRSR DOWN}{7 CRSR RIGHT}THE PLAYERS":PRINT:PRINT
610 PRINT "{2 CRSR RIGHT}{CTRL WHT}{SHIFT M}  YOU":PRINT "{CTRL BLU}{2 CRSR RIGHT}{SHIFT H}  {CTRL WHT}A ROBOT":PRINT "{2 CRSR RIGHT}{CTRL RED}{SHIFT K}  {CTRL WHT}A BARRIER"
620 PRINT "{2 CRSR RIGHT}{CTRL YEL}{SHIFT J}  {CTRL WHT}A TELEPORT RECHARGER"
630 PRINT "{2 CRSR RIGHT}{CTRL CYN}{SHIFT I}  {CTRL WHT}SPUNKY THE MARTIAN"
640 PRINT:PRINT:PRINT:PRINT "      < HIT ANY KEY TO CONTINUE >"
650 FOR I=1 TO 1000:NEXT
660 A=PEEK(197):IF A=64 THEN 660
670 PRINT "{SHIFT CLR}{3 CRSR DOWN}{5 CRSR RIGHT}{CTRL WHT}YOU CAN CONTROL YOUR MOVEMENT"
680 PRINT "{2 CRSR RIGHT}BY THE KEYBOARD OR THE JOYSTICK."
690 PRINT:PRINT"{2 CRSR RIGHT}Q - UP       O - LEFT"
700 PRINT "{2 CRSR RIGHT}A - DOWN     P - RIGHT"
710 PRINT "{2 CRSR RIGHT}SPACEBAR - TRANSPORT VIA TELEPORT"
720 PRINT "{2 CRSR RIGHT}FIRE BUTTON - TRANSPORT VIA TELEPORT"
730 PRINT:PRINT "{CRSR RIGHT}USE APPROPRIATE DIRECTIONS ON JOYSTICK"
740 PRINT "{CRSR RIGHT}AND USE CONTROL PORT #2."
750 PRINT:PRINT "      < HIT ANY KEY TO CONTINUE >"
760 FOR I= 1TO1000:NEXT
770 A=PEEK(197):IF A=64 THEN 770
780 PRINT "{SHIFT CLR}{CTRL WHT}{3 CRSR DOWN}{7 CRSR RIGHT}YOU CAN TELEPORT TO ANOTHER"
790 PRINT "PLACE ON THE BOARD BY PRESSING THE"
800 PRINT "SPACE BAR OR THE FIRE BUTTON.  YOU DO,"
810 PRINT "HOWEVER, ONLY HAVE A CERTAIN NUMBER OF"
820 PRINT "TELEPORTS WHICH ARE INDICATED BY THE"
830 PRINT "NUMBER AT THE TOP RIGHT HAND SIDE OF"
840 PRINT "THE SCREEN.  YOU CAN GAIN ADDITIONAL"
850 PRINT "TELEPORTS BY RUNNING OVER DIAMONDS."
860 PRINT "GETTING SPUNKY WILL ALSO GIVE YOU ANO-"
870 PRINT "OTHER TELEPORT.
880 PRINT "{5 CRSR RIGHT}EVERY TIME ALL TEN ROBOTS ON A"
890 PRINT "LEVEL DIE, YOU GO TO THE NEXT LEVEL."
900 PRINT "SPUNKY DOES NOT APPEAR UNTIL THE FOURTH"
910 PRINT "LEVEL.
920 PRINT:PRINT:PRINT"      < HIT ANY KEY TO CONTINUE >"
930 FOR I=1TO1000:NEXT
940 A=PEEK(197):IF A=64 THEN 940
950 REM BEGIN GAME
960 S=54272:FOR L=S TO S+24:POKE L,0:NEXT
970 GET J$:IF J$<>"" THEN 970
980 INPUT "{CTRL BLK}{SHIFT CLR}{2 CRSR DOWN}{2 CRSR RIGHT}USING JOYSTICKS Y OR N";J$
990 D=1:BA=24:DI=6:LV=1:SC=0
```

Continued

ROBOCHASE

COMMODORE 64

Continued

```
1000 GOSUB 2890
1010 GOSUB 1640
1020 GOSUB 1680
1030 GOSUB 1720
1040 IF LV>3 THEN GOSUB 1800
1050 GOSUB 1770
1060 FOR X=1TO2500:NEXT
1070 GOSUB 1830
1080 GOSUB 1120
1090 GOTO 2100
1100 IF LV>3 THEN GOTO 2630
1110 GOTO 1070
1120 IF X1=0 AND Y1=0 THEN 1560
1130 TP=(MN(1,1)*40)+1153+MN(1,2)
1140 SA=((MN(1,1)+Y1)*40)+1153+MN(1,2)+X
     1
1150 NP=PEEK(SA)
1160 IF NP=75 THEN RETURN
1170 MN(1,1)=MN(1,1)+Y1:MN(1,2)=MN(1,2)+
     X1
1180 IF NP<>32 THEN 1210
1190 POKE TP,32:POKE SA,77:POKE S+SA,1
1200 RETURN
1210 IF NP<>74 THEN 1260
1220 POKE TP,32:POKE SA,77:POKE S+SA,1
1230 HF=67:LF=15:WG=17:SD=150:GOSUB 277
     0
1240 SC=SC+125:D=D+1
1250 PRINT "■HOME■■2CRSRDOWN■";TAB(36);D:
     RETURN
1260 IF NP<>72 THEN 1340
1270 POKE TP,32:POKE SA,77:POKE S+SA,1
1280 FOR I=1TO5:IF CX=1 THEN CX=0:GOTO
     1300
1290 CX=1
1300 HF=4:LF=112:WG=33:SD=150
1310 POKE SA+S,CX:GOSUB 2770::NEXT
1320 PRINT "■SHIFT CLR■■2CRSRDOWN■YOU HA
     VE BEEN ELIMINATED BY A ROBOT":PRIN
     T:PRINT
1330 GOTO 2830
1340 IF NP<>94 THEN 1430
1350 POKE TP,32:POKE SA,77:POKE S+SA,1
1360 FOR I=1TO5:IF CX=0 THEN CX=2:GOTO 1
     380
1370 CX=0
1380 HF=6:LF=71:WG=33:SD=200:GOSUB 2770
1390 POKE S+SA,CX:NEXT
1400 PRINT "■SHIFT CLR■■2CRSRDOWN■YOU HA
     VE BEEN ELECTROCUTED BY THE"
1410 PRINT "MAD SCIENTIST'S IMPENETRABLE
      PERIMETER":PRINT:PRINT
1420 GOTO 2830
1430 IF NP<>76 THEN 1500
1440 POKE TP,32
1450 HF=6:LF=71:WG=33:SD=300
1460 FOR I=1TO5:GOSUB 2770::NEXT
1470 PRINT "■SHIFT CLR■■2CRSRDOWN■YOU HA
     VE STEPPED IN RADIOACTIVE ROBOT"
1480 PRINT "JUNK. BYE BYE."
1490 GOTO 2830
1500 IF NP<>73 THEN 1560
1510 POKE TP,32:POKE SA,77:POKE SA+S,1
1520 SC=SC+100:D=D+1
1530 HF=67:LF=15:WG=17:SD=150
1540 FOR I=1TO5:GOSUB 2770::NEXT
1550 PRINT "■HOME■■2CRSRDOWN■";TAB(36);D:
     RETURN
1560 IF FB=0 THEN RETURN
1570 IF D=0 THEN RETURN
1580 D=D-1:SC=SC-50
1590 SA=(MN(1,1)*40)+1153+MN(1,2)
1600 POKE SA,32
1610 GOSUB 2080::MN(1,1)=RW:MN(1,2)=CL
1620 PRINT "■HOME■■2CRSRDOWN■";TAB(36);D
1630 POKE SA,77:POKE S+SA,1:RETURN
1640 FOR I=1TOBA
1650 GOSUB 2080::IF PEEK(SA)<>32 THEN
     1650
1660 POKE SA,75:POKE 54272+SA,2:NEXT
1670 RETURN
1680 FOR I=1TODI
1690 GOSUB 2080::IF PEEK(SA)<>32 THEN
     1690
1700 POKE SA,74:POKE 54272+SA,7:NEXT
1710 RETURN
1720 FOR I=1TO10
1730 GOSUB 2080::IF PEEK(SA)<>32 THEN
     1730
1740 POKE SA,72:POKE 54272+SA,6
1750 BT(I,1)=RW:BT(I,2)=CL:NEXT
1760 RETURN
1770 GOSUB 2080::IF PEEK(SA)<>32 THEN
     1770
1780 POKE SA,77:POKE 54272+SA,1
1790 MN(1,1)=RW:MN(1,2)=CL:RETURN
1800 GOSUB 2080::IF PEEK(SA)<>32 THEN
     1800
1810 POKE SA,73:POKE 54272+SA,3
1820 SK(1,1)=RW:SK(1,2)=CL:RETURN
1830 X1=0:Y1=0:FB=0
1840 IF J$="N" THEN 1900
1850 XT%=PEEK(56320)AND31
1860 X1=SGN(XT% AND 4) - SGN(XT% AND 8)
1870 Y1=SGN(XT% AND 1) - SGN(XT% AND 2)
1880 FB=1 - SGN(XT% AND 16)
1890 RETURN
1900 KY=PEEK(197):IF KY=64 THEN RETURN
1910 IF KY=62 THEN Y1=-1:RETURN
1920 IF KY=10 THEN Y1=1:RETURN
1930 IF KY=38 THEN X1=-1:RETURN
1940 IF KY=41 THEN X1=1:RETURN
1950 IF KY=60 THEN FB=1:RETURN
1960 RETURN
1970 PRINT "■HOME■■6CRSRDOWN■■2CRSRRIGHT
     ■"
1980 PRINT"■2CRSRRIGHT■■SHIFT N■■SHIFT O
     ■■SHIFT P■■3SHIFT K■■SHIFT K■■SHI
     FT Q■■SHIFT K■■3SHIFT K■■3SHIFT K
     ■■SHIFT K■■SHIFT K■■SHIFT K■■SHI
     FT R■■SHIFT P■■3SHIFT K■■SHIFT K■
     ■2SHIFT R■"
1990 PRINT"■2CRSRRIGHT■■SHIFT S■■SHIFT T
     ■■CMDR A■■SHIFT K■■SHIFT K■■2SHI
     FT K■■SHIFT U■■SHIFT K■■SHIFT K■
     ■2SHIFT K■■3SHIFT K■■SHIFT K■■SH
     IFT V■■SHIFT K■■SHIFT W■■SHIFT K■■
     SHIFT P■■SHIFT K■■SHIFT X■"
2000 PRINT"■2CRSRRIGHT■■SHIFT K■■SHIFT Y
     ■■SHIFT P■■3SHIFT K■■SHIFT K■■SHI
     FT Z■■SHIFT K■■3SHIFT K■■SHIFT K
     ■■SHIFT K■■3SHIFT K■■SHIFT K■■SH
     IFT K■■3SHIFT K■■SHIFT K■■2SHIFT
     Z■"
2010 RETURN
2020 PRINT"■CTRL RED■":RETURN
2030 PRINT"■CTRL PUR■":RETURN
2040 PRINT"■CTRL WHT■":RETURN
2050 PRINT"■CTRL GRN■":RETURN
2060 PRINT"■CTRL BLU■":RETURN
2070 PRINT"■CTRL YEL■":RETURN
2080 CL=INT(RND(0)*20)+1:RW=INT(RND(0)*1
     8)+1
2090 SA=1153+CL+(40*RW):RETURN
2100 FOR I=1TO10:VI(I)=0:NEXT
2110 CT=0:FL=0:RD=0:Y1=0:X1=0
2120 FOR I=1 TO 10
2130 IF BT(I,1)=0 THEN RD=RD+1:GOTO 247
     0
2140 TP=(BT(I,1)*40)+1153+BT(I,2)
2150 IF PEEK(TP)=76 THEN BT(I,1)=0:RD=RD
     +1:GOTO 2470
2160 IF BT(I,1)=MN(1,1) THEN Y1=0:GOTO
     2190
2170 IF BT(I,1)<MN(1,1) THEN Y1=1:GOTO
     2190
2180 Y1=-1
2190 IF BT(I,2)=MN(1,2) THEN X1=0:GOTO
     2220
2200 IF BT(I,2)<MN(1,2) THEN X1=1:GOTO
     2220
2210 X1=-1
2220 NY=ABS(BT(I,1)-MN(1,1)):NX=ABS(BT(I
     ,2)-MN(1,2))
2230 RN=INT(RND(0)*20)+1
2240 IF RN>LV THEN 2260
2250 BT(I,1)=BT(I,1)+Y1:BT(I,2)=BT(I,2)+
     X1:GOTO 2280
2260 IF NY>NX THEN BT(I,1)=BT(I,1)+Y1:GO
     TO 2280
2270 BT(I,2)=BT(I,2)+X1
2280 SA=(BT(I,1)*40)+1153+BT(I,2)
2290 NP=PEEK(SA)
2300 IF NP<>32 AND NP<>73 THEN 2320
2310 POKE TP,32:POKE SA,72:POKE SA+S,6:G
     OTO 2470
2320 IF NP<>76 AND NP<>74 AND NP<>75 THE
     N 2340
2330 POKE TP,32:RD=RD+1:BT(I,1)=0:GOTO
     2470
2340 IF NP<>77 THEN 2410
2350 POKE TP,32
2360 FOR I=1TO10:IF CX=0 THEN CX=1:GOTO
     2380
2370 CX=0
2380 HF=4:LF=112:WG=33:SD=150
2390 POKE S+SA,CX:GOSUB 2770::NEXT
2400 PRINT "■SHIFT CLR■■2CRSRDOWN■■CRSRR
     IGHT■YOU HAVE BEEN ELIMINATED BY A
     ROBOT":PRINT:PRINT:GOTO 2830
2410 IF NP<>72 THEN 2470
2420 POKE TP,32:POKE SA,76:POKE SA+S,0:R
     D=RD+1
2430 FOR L6=0 TO I-1
2440 IF BT(L6,1)=BT(I,1) AND BT(L6,2)=BT
     (I,2) THEN BT(L6,1)=0:RD=RD+1
2450 NEXT L6
2460 BT(I,1)=0
2470 NEXT I
2480 IF RD<10 THEN 1100
2490 LV=LV+1:IF BA=12 THEN 2510
2500 BA=BA-4
2510 IF DI=3 THEN 2530
2520 DI=DI-1
2530 SC=SC+250+(10*LV)
2540 IF D=0 THEN 2560
2550 D=1:FOR X=1TO 5000:NEXT
2560 PRINT "■SHIFT CLR■■2CRSRDOWN■■CRSRR
     IGHT■CONGRATULATIONS, YOU HAVE SUCC
     ESSFULLY"
2570 PRINT "■2CRSRRIGHT■CONQUERED LEVEL"
     ;LV-1
2580 PRINT:PRINT"■2CRSRRIGHT■YOUR SCORE
     IS";SC
2590 PRINT "■2CRSRDOWN■PRESS ANY KEY TO
     CONTINUE"
```

Continued

ROBOCHASE

Continued

COMMODORE 64

```
2600 FOR X=1TO1000:NEXT
2610 Z=PEEK(197):IF Z=64 THEN 2610
2620 GOTO 1000
2630 DS=INT(RND(0)*6)+1:X1=0:Y1=0
2640 ON DS GOTO 2650, 2660, 2670, 2680,
     2690, 2700
2650 X1=1:GOTO 2710
2660 X1=-1:GOTO 2710
2670 Y1=1:GOTO 2710
2680 Y1=-1:GOTO 2710
2690 X1=-1:Y1=-1:GOTO 2710
2700 X1=1:Y1=1
2710 TP=(SK(1,1)*40)+1153+SK(1,2)
2720 SS=((SK(1,1)+Y1)*40)+1153+SK(1,2)+X
     1
2730 NP=PEEK(SS):IF NP<>32 THEN 2630
2740 SK(1,1)=SK(1,1)+Y1:SK(1,2)=SK(1,2)+
     X1
2750 POKE TP,32:POKE SS,73:POKE S+SS,3
2760 GOTO 1110
2770 POKE S+5,S5:POKE S+6,240
2780 POKE S+24,15
2790 POKE S+1,HF:POKE S,LF
2800 POKE S+4,WG
2810 FOR ZX=1TOSD:NEXT
2820 POKE S+4,WG-1:RETURN
2830 PRINT:PRINT "YOUR FINAL SCORE IS";S
     C
2840 PRINT:PRINT "YOUR FINAL LEVEL WAS";
     LV
2850 GET A$:IF A$<>"" THEN 2850
2860 PRINT:INPUT "WOULD YOU LIKE TO PLAY
     AGAIN Y OR N";A$
2870 IF A$="N" THEN END
2880 RUN 380
2890 PRINT "■ SHIFT CLR■ CRSRDOWN■ 15CRSR
     RIGHT■ CTRL BLK■ SHIFT ·■ SHIFT A■
     SHIFT B■ SHIFT A■ SHIFT C■ SHIFT D■
     ■SHIFT E■ SHIFT F■ SHIFT G■":PRINT
     TAB(36);D
2900 PRINT TAB(9)"********************
     "
2910 FOR I=1TO18
2920 PRINT TAB(9)"*"TAB(30)"*":NEXT
2930 PRINT TAB(9)"********************
     "
2940 RETURN
```

HCM

ROBOCHASE

IBM PCjr

```
100 REM ***********
110 REM * ROBOCHASE *
120 REM ***********
130 REM BY GREG VAUGHAN
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.3.1
170 REM IBM PCjr
180 REM CARTRIDGE BASIC
190 CLS:KEY OFF:CLEAR ,,,32768!:SCREEN
    5:NR=10:TP=1:L=1
200 OPTION BASE 1:DIM B(34,18),R(10,2)
210 DEF FNRXO=R(Z,1)*8+16
220 DEF FNRYO=R(Z,2)*8+16
230 DEF FNRXN=(R(Z,1)+MX)*8+16
240 DEF FNRYN=(R(Z,2)+MY)*8+16
250 RANDOMIZE TIMER:KEY 15,CHR$(&H40)+C
    HR$(&H10):KEY 16,CHR$(&H40)+CHR$(&H
    18):KEY 17,CHR$(&H40)+CHR$(&H19):KE
    Y 18,CHR$(&H40)+CHR$(&H1E)
260 BEEP OFF:SOUND ON:PLAY "MB"
270 LOCATE 12,15:PRINT "ROBOCHASE":FOR
    A=1 TO 50
280 IF A*.5=INT(A*.5) THEN LINE (107-A*
    2,83-A)-(187+A*2,99+A),9,B ELSE LIN
    E (107-A*2,83-A)-(187+A*2,99+A),4,B
290 NEXT A:LOCATE 21,4:PRINT "USE UPPER
    CASE ONLY (CAPS LOCK)":PRINT:PRINT
    TAB(7);"PRESS ANY KEY TO CONTINUE"
300 A$=INKEY$:IF A$="" THEN 300 ELSE CL
    S
310 LOCATE 8,1:PRINT "INPUT OPTION:":PR
    INT:PRINT "1. JOYSTICK":PRINT:PRINT
    "2. KEYBOARD"
320 A$=INKEY$:IF A$="" THEN 320 ELSE IF
    A$<"1" OR A$>"2" THEN 320 ELSE OPT
    =VAL(A$)
330 CLS:LOCATE 1,15:PRINT "ROBOCHASE"
340 FOR X=0 TO 11:IF X/2=INT(X/2) THEN
    LINE (23-X*2,23-X)-(297+X*2,168+X),
    11,B ELSE LINE (23-X*2,23-X)-(297+X
    *2,168+X),4,B
350 NEXT X:LINE (23,23)-(1,12),15:LINE
    (297,23)-(319,12),15:LINE (23,168)-
    (1,179),15:LINE (297,168)-(319,179)
    ,15
360 B$="C11R7D7L7U7BF2FRNEDNFLNGUBUP4,1
    1"
370 P$="BR3R2DL2FD3RU2R2U2BD6LUBL4DLBLB
    U5DFERD2"
380 S$="BD2D2FDRDR3URUL4HUR2DRUR2DRU2BL
    2L3ER"
390 D$="BR3F3G3H3E3BD2D2NENH"
400 R$="BDD2R2ND4NU2RND2NU3RND2NU3RND4N
    U2R2D2"
410 J$="BD7EURUERFDFDNRL5E2URDFL2"
420 FOR Z=1 TO 30-3*L
430 X=INT(RND*34)+1:Y=INT(RND*18)+1:IF
    B(X,Y)<>0 THEN 430 ELSE B(X,Y)=1:PR
    ESET (X*8+16,Y*8+16):DRAW "XB$;"
440 NEXT Z
450 X=INT(RND*34)+1:Y=INT(RND*18)+1:IF
    B(X,Y)<>0 THEN 450 ELSE B(X,Y)=6:PX
    =X:PY=Y:PRESET (X*8+16,Y*8+16):DRAW
    "C14;XP$;"
460 FOR Z=1 TO (10-L)*(L<11)*(-1)+1
470 X=INT(RND*34)+1:Y=INT(RND*18)+1:IF
    B(X,Y)<>0 THEN 470 ELSE B(X,Y)=3:PR
    ESET (X*8+16,Y*8+16):DRAW "C15;XD$;
    "
480 NEXT Z:FOR Z=1 TO 10
490 X=INT(RND*34)+1:Y=INT(RND*18)+1:IF
    B(X,Y)<>0 OR ABS(PX-X)<6 OR ABS(PY-
    Y)<6 THEN 490 ELSE B(X,Y)=2:R(Z,1)=
    X:R(Z,2)=Y:PRESET (X*8+16,Y*8+16):D
    RAW "C13;XR$;"
500 NEXT Z
510 IF L<4 THEN 530
520 X=INT(RND*34)+1:Y=INT(RND*18)+1:IF
    B(X,Y)<>0 THEN 520 ELSE SX=X:SY=Y:P
    RESET (X*8+16,Y*8+16):DRAW "C2;XS$;
    "
530 LOCATE 24,3:PRINT "SCORE:";SC;TAB(3
    0);"LEVEL:";L;:LOCATE 25,10:PRINT "
    TELEPORT CHARGES:";TP;
540 ON STRIG(0) GOSUB 1170:STRIG(0) OFF
550 ON KEY(15) GOSUB 1200
560 ON KEY(16) GOSUB 1210
570 ON KEY(17) GOSUB 1220
580 ON KEY(18) GOSUB 1230
590 KEY 19,CHR$(&H40)+CHR$(&H39)
600 ON KEY(19) GOSUB 1170
610 NG=1
620 LOCATE 24,9:PRINT SC;:LOCATE 25,28:
    PRINT TP;:IF OPT=1 THEN 660
630 MX=0:MY=0:KEY(15) ON:KEY(16) ON:KEY
    (17) ON:KEY(18) ON:KEY(19) ON
640 KEY(15) STOP:KEY(16) STOP:KEY(17) S
    TOP:KEY(18) STOP:KEY(19) STOP
650 IF F=1 THEN F=0:GOTO 720 ELSE GOTO
    750
660 MX=0:MY=0:AX=STICK(0):AY=STICK(1)
670 STRIG(0) ON:STRIG(0) STOP:IF AY>50
    AND AY<76 AND AX>50 AND AX<76 THEN
    750
680 IF AY<51 THEN MY=MY-1
690 IF AY>77 THEN MY=MY+1
700 IF AX<51 THEN MX=MX-1
710 IF AX>77 THEN MX=MX+1
720 IF PX+MX>34 OR PX+MX<1 OR PY+MY>18
    OR PY+MY<1 THEN 1070
730 C=B(PX+MX,PY+MY):IF C<>0 THEN 1060
740 PRESET (PX*8+16,PY*8+16):DRAW "C0;X
    P$;":B(PX,PY)=0:PX=PX+MX:PY=PY+MY:P
    RESET (PX*8+16,PY*8+16):DRAW "C14;X
    P$;":B(PX,PY)=6:SOUND 770,1,8
750 NG=0:IF L<4 OR SP=1 THEN 840
760 MX=INT(RND*5)-2:MY=INT(RND*5)-2:IF
    SX+MX<1 OR SX+MX>34 OR SY+MY<1 OR S
    Y+MY>18 THEN 760
770 X=B(SX+MX,SY+MY):ON X+1 GOSUB 790,8
    00,810,800,800,800,830
780 GOTO 840
790 PRESET (SX*8+16,SY*8+16):DRAW "C0;X
    S$;":B(SX,SY)=0:SX=SX+MX:SY=SY+MY:P
    RESET (SX*8+16,SY*8+16):DRAW "C2;XS
    $;":B(SX,SY)=4
800 RETURN
810 FOR X=1 TO 10:IF R(X,1)=SX AND R(X,
    2)=SY THEN RB(X)=1:SOUND 220,10:X=1
    0
820 NEXT X:RETURN
830 SC=SC+100:PRESET (SX*8+16,SY*8+16):
    DRAW "C0;XS$;":PLAY "L48 O4 CDEF#GA
    B>CDEF#GAB":SP=1:B(SX,SY)=0:TP=TP+1
    :RETURN
840 FOR Z=1 TO 10:IF RB(Z)=1 THEN 900 E
    LSE MX=0:MY=0:R=INT(RND*(10-L))+1:I
    F R<1 THEN R=1
850 ON R GOSUB 910,910,920,920,920,920,
    930,930,930,930
860 IF MX=0 AND MY=0 THEN GOSUB 910
870 IF B(R(Z,1)+MX,R(Z,2)+MY)<>0 THEN G
    OSUB 940:IF NR<1 THEN 1160 ELSE GOT
    O 900
880 PRESET (R(Z,1)*8+16,R(Z,2)*8+16):DR
    AW "C0;XR$;":B(R(Z,1),R(Z,2))=0:R(Z
    ,1)=R(Z,1)+MX:R(Z,2)=R(Z,2)+MY:PRES
    ET (R(Z,1)*8+16,R(Z,2)*8+16):DRAW "
    C13;XR$;":B(R(Z,1),R(Z,2))=2
890 SOUND 880,1,8
900 NEXT Z:GOTO 620
910 GOSUB 920:GOSUB 930:RETURN
920 MX=SGN(PX-R(Z,1)):RETURN
930 MY=SGN(PY-R(Z,2)):RETURN
```

Continued

PROGRAM LISTING

ROBOCHASE

IBM PCjr

Continued

```
940 ON B(R(Z,1)+MX,R(Z,2)+MY) GOTO 950,
    960,960,990,960,1000
950 PRESET (R(Z,1)*8+16,R(Z,2)*8+16):DR
    AW "C0;XR$;":NOISE 6,15,10:RB(Z)=1:
    NR=NR-1:SC=SC+25:B(R(Z,1),R(Z,2))=0
    :RETURN
960 PRESET (FNRXO,FNRYO):DRAW "C0;XR$;"
    :NOISE 6,15,5:B(R(Z,1),R(Z,2))=0:PR
    ESET (FNRXN,FNRYN):DRAW "C0;XR$;":P
    RESET (FNRXN,FNRYN):DRAW "C6;XJ$;":
    RB(Z)=1:NR=NR-1:SC=SC+25:B(R(Z,1)+M
    X,R(Z,2)+MY)=5
970 FOR Z1=1 TO 10:IF R(Z1,1)=R(Z,1)+MX
     AND R(Z1,2)=R(Z,2)+MY AND RB(Z1)=0
     THEN RB(Z1)=1:NR=NR-1:Z1=10:SC=SC+
    25
980 NEXT Z1:RETURN
990 PRESET (FNRXO,FNRYO):DRAW "C0;XR$;"
    :RB(Z)=1:NR=NR-1:SC=SC+25:B(R(Z,1),
    R(Z,2))=0:SOUND 220,5:RETURN
1000 PRESET (FNRXO,FNRYO):DRAW "C0;XR$;"
    :PRESET (FNRXN,FNRYN):DRAW "C0;XP$;
    ":PRESET (FNRXN,FNRYN):DRAW "C13;XR
    $;"
1010 PLAY "O6 L12 BAGF#EDC< L20 BAGF#EDC
    < L28 BAGF#EDC< L36 BAGF#EDC< L44 B
    AGF#EDC< L52 BAGF#EDC< L60 BAGF#EDC
    "
1020 CLS:LOCATE 12,1:PRINT "YOU HAVE BEE
    N CAPTURED AND KILLED BY A ROBOT.
    YOU HAVEN'T GOT A CHANCE.":FOR Z=1
    TO 2000:NEXT Z:GOTO 1030
1030 CLS:LOCATE 12,1:PRINT "YOUR SCORE I
    S:";SC+(TP*L):LOCATE 20,1:PRINT "WO
    ULD YOU LIKE TO PLAY AGAIN (Y/N)?"
1040 A$=INKEY$:IF A$<>"Y" AND A$<>"N" AN
    D A$<>"y" AND A$<>"n" THEN 1040
1050 IF A$="Y" OR A$="y" THEN RUN ELSE E
    ND
1060 ON C GOTO 1070,1090,1110,1130,1150,
    620
1070 PRESET (PX*8+16,PY*8+16):DRAW "C0;X
    P$;":PLAY "O5 L48 ACDEFGECFGAEBCADE
    GFACBDCC#DEFF#GAB>CC#DEFF#GAB":CLS:
    LOCATE 12,1:PRINT "YOU HAVE COLLIDE
    D WITH A BARRIER AND HAVE DIED F
    ROM RADIOACTIVE POISONING."
1080 FOR D=1 TO 2000:NEXT D:GOTO 1030
1090 PRESET (PX*8+16,PY*8+16):DRAW "C0;X
    P$;":GOTO 1010
1100 CLS:IF C=2 THEN LOCATE 12,1:PRINT "
    YOU HAVE BEEN CAPTURED AND KILLED B
    Y A ROBOT. YOU HAVN'T GOT A CHANCE
    .":FOR Z=1 TO 2000:NEXT Z:GOTO 1030
1110 PRESET ((PX+MX)*8+16,(PY+MY)*8+16):
    DRAW "C0;XD$;":PRESET (PX*8+16,PY*8
    +16):DRAW "C0;XP$;":B(PX,PY)=0:PX=P
    X+MX:PY=PY+MY:PRESET (PX*8+16,PY*8+
    16):DRAW "C14;XP$;":SC=SC+75:TP=TP+
    1
1120 FOR Z=440 TO 6000 STEP 200:SOUND Z,
    1,15:NEXT Z:LOCATE 25,27:PRINT TP;:
    GOTO 840
1130 PRESET (PX*8+16,PY*8+16):DRAW "C0;X
    P$;":PRESET (SX*8+16,SY*8+16):DRAW
    "C0;XS$;":PRESET (SX*8+16,SY*8+16):
    DRAW "C13;XP$;":B(PX,PY)=0:PX=SX:PY
    =SY:B(PX,PY)=6:SP=1:SC=SC+100:TP=TP
    +1
1140 FOR Z=1 TO 30:SOUND INT(RND*4000)+2
    000,1,15:NEXT Z:GOTO 840
1150 PRESET (PX*8+16,PY*8+16):DRAW "C0;X
    P$;":FOR Z=440 TO 110 STEP -20:SOUN
    D Z,1,15:NEXT Z:CLS:LOCATE 12,1:PRI
    NT "YOU HAVE RUN INTO A JUNK PILE."
    THE RADIATION HAS KILLED YOU."
    :FOR Z=1 TO 2000:NEXT Z:GOTO 1030
1160 PLAY "O3 L36 CDEF#GAB>CDEF#GAB>CDEF
    #GAB>CDEF#GAB":L=L+1:NR=10:FOR A=1
    TO 34:FOR C=1 TO 18:B(A,C)=0:NEXT C
    :NEXT A:FOR A=1 TO 10:RB(A)=0:NEXT
    A:SP=0:SC=SC+125:GOTO 330
1170 IF TP<1 THEN RETURN ELSE FOR Z=14 T
    O 0 STEP -.5:PRESET (PX*8+16,PY*8+1
    6):DRAW "C=Z;XP$;":SOUND INT(RND*50
    00+1000),1,15:B(PX,PY)=0:NEXT Z
1180 PX=INT(RND*34+1):PY=INT(RND*18+1):I
    F B(PX,PY)<>0 THEN 1180
1190 FOR Z=1 TO 14:PRESET (PX*8+16,PY*8+
    16):DRAW "C=Z;XP$;":SOUND INT(RND*1
    0000+2000),1,15:NEXT Z:B(PX,PY)=6:T
    P=TP-1:RETURN
1200 IF NG=0 THEN MY=-1:F=1:RETURN ELSE
    RETURN
1210 IF NG=0 THEN MX=-1:F=1:RETURN ELSE
    RETURN
1220 IF NG=0 THEN MX=1:F=1:RETURN ELSE R
    ETURN
1230 IF NG=0 THEN MY=1:F=1:RETURN ELSE R
    ETURN
```

HCM

SNAP-CALC

APPLE II Family

```
100 REM  **********
110 REM  *SNAP-CALC*
120 REM  **********
130 REM
140 REM   BY GARY STRAUSS AND
150 REM   THE HCM STAFF
160 REM   HOME COMPUTER MAGAZINE
170 REM   VERSION 4.3.1
180 REM   APPLE II SERIES APPLESOFT
190 REM
200 REM
210 REM
220 ONERR GOTO 5500: REM  SET UP ERRO
    R HANDLER
230 GOSUB 390
240 GOSUB 5260
250 IF KF = 0 OR A$ < > CHR$ (13) THE
    N 240
260 HOME : GOSUB 700: GOSUB 1140
270 GOSUB 5260: IF KF = 0 THEN 270
280 IF A$ = ESC$ THEN GOSUB 5370: GOTO
     260
290 X = ASC (A$): ON (KY(7) = X) + 2 *
    (KY(8) = X) + 3 * (KY(9) = X) + 4
    * (KY(10) = X) GOSUB 1250,1350,1450
    ,1570
300 ON (KY(11) = X) + 2 * (KY(12) = X)
    + 3 * (KY(13) = X) + 4 * (KY(14) =
    X) GOSUB 1700,1790,1870,1990
310 ON (KY(3) = X) + 2 * (KY(4) = X) GO
    SUB 2990,3620
320 IF KY(1) = X OR KY(2) = X THEN ON
    (KY(1) = X) + 2 * (KY(2) = X) GOSUB
     2070,2550: GOSUB 1140: GOTO 270
330 IF KY(5) = X THEN GOSUB 4820: GOTO
    270
340 IF X = KY(6) THEN GOSUB 4040: GOTO
    260
350 IF FN C(C) < = 0 THEN GOSUB 3540
    : GOTO 270
360 IF (A$ > = "0" AND A$ < = "9") OR
    A$ = "." THEN GOSUB 3440: GOTO 27
    0
370 GOTO 270
380 END
390 REM  INITIALIZATION AND TITLE SCREE
    N
400 REM  (REF:2)
410 REM  CONTROL KEYS
420 DIM KY(14)
430 KY(1) = 12: REM  CTRL L - LOAD DATA
440 KY(2) = 19: REM  CTRL S - SAVE DATA
450 KY(3) = 5: REM  CTRL E - ENTRY CLEA
    R
460 KY(4) = 26: REM  CTRL Z - CALCULAT
    E MODEL
470 KY(5) = 16: REM  CTRL P - PRINT REP
    ORT
480 KY(6) = 6: REM  CTRL F - LOGIC ENTR
    Y
490 KY(7) = 9: REM  CTRL I - CURSOR UP
500 KY(8) = 10: REM  CTRL J - CURSOR LE
    FT
510 KY(9) = 11: REM  CTRL K - CURSOR RI
    GHT
520 KY(10) = 13: REM  CTRL M - CURSOR D
    OWN
530 KY(11) = 25: REM  CTRL Y - PAGE UP
540 KY(12) = 7: REM  CTRL G - PAGE LEFT
550 KY(13) = 8: REM  CTRL H - PAGE RIGH
    T
560 KY(14) = 2: REM  CTRL B - PAGE DOWN
570 ESC$ = CHR$ (27)
580 A = 60:B = 30:TC = 13:LC = 12:CL =
    0:RW = 1
590 DIM D$(A),E$(A),F$(130),J(A,B),K(10
    0),N$(10),LG$(10)
600 FOR IT = 1 TO 7: READ LG$(IT): NEXT
610 FOR IT = 1 TO 5: READ N$(IT): NEXT
620 DEF FN R(R) = RW + R - 1: DEF FN
    C(C) = CL + C - 1
630 R = 1:C = 2
640 NAM$ = ""
650 DIM CP(3):CP(1) = 1
660 REM  TITLE SEGMENT
670 HOME : VTAB 12: HTAB 15: PRINT "SNA
    P-CALC"
680 VTAB 23: HTAB 8: PRINT "PRESS RETUR
    N TO CONTINUE"
690 RETURN
700 REM  DISPLAY SPREAD SHEET
710 VTAB 1: HTAB 1: PRINT NAM$
720 VTAB 3: HTAB 4: INVERSE : FOR IT =
    1 TO 36: PRINT "#";: NEXT
730 FOR IT = 4 TO 23: VTAB IT: HTAB 4
740 PRINT "#";: NEXT : NORMAL
750 IF CL < > 0 THEN 780
760 VTAB 2: HTAB 5: PRINT "ROW NAME"
770 GOTO 790
```

Continued

```
780 VTAB 2: HTAB 5: PRINT "                    ";CL;"
    ";
790 VTAB 2: HTAB 22: PRINT CL + 1; TAB(
    34);CL + 2;" ";
800 FOR IT = 1 TO 10:RS = "  " + STR$ (
    INT (RW + IT - 1))
810 VTAB IT * 2 + 3: HTAB 4 - LEN (R$)
820 PRINT R$;: NEXT
830 IF CL > 0 THEN 930
840 FOR IT = 1 TO 10: VTAB IT * 2 + 3:
    HTAB 5
850 INS = DS(IT + RW - 1): IF LEN (INS
    ) = 0 THEN PRINT IT + RW - 1;"
    ";: GOTO 870
860 PRINT LEFTS ((DS(IT + RW - 1) + "
    "),10)
870 NEXT
880 FOR IT = 1 TO 10: VTAB IT * 2 + 3:
    HTAB 17
890 X = J(RW + IT - 1,CL + 1): GOSUB 99
    0: PRINT XS;
900 VTAB IT * 2 + 3: HTAB 29
910 X = J(RW + IT - 1,CL + 2): GOSUB 99
    0: PRINT XS;
920 NEXT : RETURN
930 FOR IT = 1 TO 10: VTAB IT * 2 + 3:
    HTAB 5
940 X = J(RW + IT - 1,CL): GOSUB 990: P
    RINT XS;
950 HTAB 17:X = J(RW + IT - 1,CL + 1):
    GOSUB 990: PRINT XS;
960 HTAB 29:X = J(RW + IT - 1,CL + 2):
    GOSUB 990: PRINT XS;
970 NEXT
980 RETURN
990 REM     NUMBER FORMAT
1000 XZ = SGN (X): X = INT ( ABS (X) *
    100 + .5):XS = STRS (X)
1010 ON (4 * (X = 0) + 3 * (X > 0) + (X
    > 100)) - 2 GOTO 1030,1020,1040,1050
1020 XS = "0.00": GOTO 1060
1030 XS = "0.0" + XS: GOTO 1060
1040 XS = "0." + XS: GOTO 1060
1050 XS = LEFTS (XS, LEN (XS) - 2) + "."
     + RIGHTS (XS,2)
1060 IF XZ < 0 THEN XS = "-" + XS
1070 XS = RIGHTS ((" " + XS),9)
1080 RETURN
1090 REM     CELL CONTENTS
1100 IF CL + CC - 1 = 0 THEN GOSUB 1200:
     GOTO 1120
1110 X = J(RW + R - 1,CL + C - 1): GOSUB
     990
1120 NORMAL : VTAB R * 2 + 3: HTAB (C -
     1) * 12 + 5
1130 PRINT XS;: RETURN
1140 IF CL + CC - 1 = 0 THEN GOSUB 1200:
     GOTO 1160
1150 X = J(RW + R - 1,CL + C - 1): GOSUB
     990
1160 INVERSE : VTAB R * 2 + 3: HTAB (C -
     1) * 12 + 5
1170 PRINT XS;: NORMAL : RETURN
1180 NORMAL : VTAB R * 2 + 2: HTAB (C -
     1) * 12 + 5: PRINT " ";: RET
     URN
1190 INVERSE : VTAB R * 2 + 2: HTAB (C -
     1) * 12 + 5: PRINT " ";: NO
     RMAL : RETURN
1200 INS = DS(RW + R - 1): IF LEN (INS)
     > 0 THEN 1230
1210 XS = LEFTS (" " + STRS (RW + R - 1) +
     "),10)
1220 RETURN
1230 XS = LEFTS ((DS(RW + R - 1) + "
     "),10)
1240 RETURN
1250 REM     CURSOR UP
1260 CP(1) = 1
1270 GOSUB 1090
1280 IF R = 1 THEN 1310
1290 R = R - 1: GOTO 1330
1300 :
1310 RW = RW - 1: IF RW < 1 THEN RW = 1:
     GOTO 1330
1320 GOSUB 750
1330 GOSUB 1140
1340 RETURN
1350 REM     CURSOR LEFT
1360 CP(1) = 1
1370 GOSUB 1090
1380 IF CL = 0 AND C = 1 THEN GOSUB 114
     0: RETURN
1390 IF C = 1 THEN 1410
1400 C = C - 1: GOTO 1430
1410 CL = CL - 1: IF CL < = 0 THEN CL =
     0
1420 GOSUB 750
1430 GOSUB 1140
1440 RETURN
1450 REM     CURSOR RIGHT
1460 CP(1) = 1
1470 GOSUB 1090
1480 IF FN C(C) = LC AND TC = 0 THEN R
     ETURN
1490 IF FN C(C) < > LC THEN 1510
1500 CL = TC - 2:C = 2: GOSUB 750: GOTO
     1550
1510 IF C = 3 THEN CL = CL + 1
1520 IF CL > B - 2 THEN CL = B - 2
1530 IF C < 3 THEN C = C + 1: GOTO 1550
1540 GOSUB 750
1550 GOSUB 1140
1560 RETURN
1570 REM     CURSOR DOWN
1580 CP(1) = 1
1590 GOSUB 1090
1600 IF R < > 10 THEN R = R + 1: GOTO 1
     640
1610 RW = RW + 1
1620 IF RW > A - 9 THEN RW = A - 9: GOTO
     1640
1630 GOSUB 750
1640 GOSUB 1140
1650 IF FN C(C) = 0 THEN RETURN
1660 IF ES( FN R(R)) = "" THEN RETURN
1670 IF ( FN R(R) = A) THEN RETURN
1680 GOSUB 1090: GOTO 1600
1690 RETURN
1700 REM     PAGE UP
1710 CP(1) = 1
1720 GOSUB 1090: IF FN R(R) < = 1 THEN
     GOSUB 1140: RETURN
1730 IF R = 1 THEN 1760
1740 R = R - 5: IF R < 1 THEN R = 1
1750 GOSUB 1140: RETURN
1760 RW = RW - 5: IF RW < 1 THEN RW = 1
1770 GOSUB 750: GOSUB 1140
1780 RETURN
1790 REM     PAGE LEFT
1800 CP(1) = 1
1810 GOSUB 1090
1820 IF C = 1 THEN 1840
1830 C = 1: GOSUB 1140: RETURN
1840 CL = CL - 3: IF CL < = 0 THEN CL =
     0
1850 GOSUB 750: GOSUB 1140
1860 RETURN
1870 REM     PAGE RIGHT
1880 CP(1) = 1
1890 GOSUB 1090
1900 IF FN C(C) = LC AND TC = 0 THEN R
     ETURN
1910 IF FN C(C) = LC THEN GOSUB 1140:
     RETURN
1920 IF FN C(C) < LC - 2 THEN 1940
1930 CL = TC - 2:C = 2: GOTO 1960
1940 IF C < 3 THEN C = 3: GOSUB 1140: RE
     TURN
1950 CL = CL + 3: IF CL > B - 2 THEN CL
     = B - 2
1960 GOSUB 750
1970 GOSUB 1140
1980 RETURN
1990 REM     PAGE DOWN
2000 CP(1) = 1: IF R = 10 THEN 2030
2010 GOSUB 1090: IF R < 6 THEN R = R + 5
     : GOSUB 1140: GOTO 2050
2020 R = 10: GOSUB 1140: GOTO 2050
2030 RW = RW + 5: IF RW > A - 9 THEN RW
     = A - 9: GOSUB 750: GOSUB 1140: RET
     URN
2040 GOSUB 750: GOSUB 1140
2050 IF ES( FN R(R)) > "" THEN GOTO 200
     0
2060 RETURN
2070 REM     LOAD FROM DISK
2080 CP(1) = 1
2090 HOME : VTAB 1: HTAB 1
2100 PRINT "ENTER YOUR CHOICE:": PRINT "
     "
2110 PRINT "1) LOAD DATA": PRINT "2) LOA
     D LOGIC"
2120 GOSUB 5260
2130 IF KF = 1 AND AS = CHRS (13) THEN
     HOME : GOSUB 700: RETURN
2140 IF KF = 0 OR (AS < "1" OR AS > "2")
     THEN 2120
2150 ON VAL (AS) GOTO 2160,2250
2160 VTAB 10: HTAB 1: INPUT "ENTER FILE
     NAME: ";NS
2170 IF LEN (NS) = 0 THEN HOME : GOSUB
     700: RETURN
2180 NS = NS + ".HCD"
2190 PRINT CHRS (4);"OPEN"NS: PRINT CH
     RS (4);"READ"NS
2200 FOR IT = 1 TO A: FOR JI = 1 TO B: I
     NPUT INS
2210 J(IT,JI) = VAL (INS): NEXT : NEXT
2220 PRINT CHRS (4);"CLOSE"NS
2230 HOME : GOSUB 700
2240 RETURN
2250 IF NAMS < > "" THEN 2290
2260 VTAB 10: HTAB 1
2270 INPUT "ENTER LOGIC NAME: ";NAMS
2280 IF LEN (NAMS) = 0 THEN HOME : GOS
     UB 700: RETURN
2290 NS = LEFTS (NAMS,26) + ".HCL"
2300 PRINT "": PRINT "WORKING . . . "
2310 PRINT CHRS (4);"OPEN"NS: PRINT CH
     RS (4);"READ"NS
2320 INPUT N1S
```

Continued

PROGRAM LISTING

```
2330 INPUT IN$:T1 = VAL (IN$): INPUT IN$:L1 = VAL (IN$)
2340 INPUT IN$:A1 = VAL (IN$): INPUT IN$:B1 = VAL (IN$)
2350 IF A1 < = A AND B1 < = B THEN 2410
2360 PRINT CHR$ (4);"CLOSE"N$
2370 VTAB 12: HTAB 1: PRINT "THE LOGIC IS TOO LARGE. CHANGE LINE 1200."
2380 HTAB 1: PRINT "A = ";A1;" AND B = ";B1
2390 HOME : GOSUB 700
2400 RETURN
2410 IF (A1 < A) THEN A = A1
2420 IF B1 < B THEN B = B1
2430 FOR IT = 1 TO A: INPUT D$(IT): INPUT X$:X = VAL (X$)
2440 IF "" = ASC (D$(IT)) = 127 THEN D$(IT) = "": IF X = 0 THEN 2490
2450 E$(IT) = "": IF X = 0 THEN 2490
2460 FOR JI = 1 TO X: INPUT IN$
2470 E$(IT) = E$(IT) + CHR$ ( VAL (IN$))
2480 NEXT
2490 NEXT
2500 FOR IT = 1 TO 100: INPUT IN$:K(IT) = VAL (IN$): NEXT
2510 PRINT CHR$ (4);"CLOSE"N$
2520 TC = T1:LC = L1
2530 HOME : GOSUB 700
2540 RETURN
2550 REM SAVE TO DISK
2560 CP(1) = 1
2570 HOME : VTAB 1: HTAB 1
2580 PRINT "ENTER YOUR CHOICE:": PRINT "1) SAVE DATA": PRINT "2) SAVE LOGIC"
2600 GOSUB 5260
2610 IF KF = 1 AND A$ = CHR$ (13) THEN HOME : GOSUB 700: RETURN
2620 IF KF = 0 OR (A$ < "1" OR A$ > "2") THEN 2600
2630 ON VAL (A$) GOTO 2640,2750
2640 VTAB 10: HTAB 1: INPUT "ENTER FILE NAME: ";N$
2650 IF LEN (N$) = 0 THEN HOME : GOSUB 700: RETURN
2660 N$ = N$ + ".HCD"
2670 PRINT "": PRINT "WORKING . . . "
2680 PRINT CHR$ (4);"OPEN";N$
2690 PRINT CHR$ (4);"WRITE";N$
2700 FOR IT = 1 TO A: FOR JI = 1 TO B: IN$ = STR$ (J(IT,JI))
2710 PRINT IN$: NEXT : NEXT
2720 PRINT CHR$ (4);"CLOSE"N$
2730 HOME : GOSUB 700
2740 RETURN
2750 IF NAM$ < > "" THEN 2780
2760 VTAB 10: HTAB 1
2770 INPUT "ENTER LOGIC NAME: ";NAM$
2780 N$ = LEFT$ (NAM$,26) + ".HCL"
2790 PRINT "": PRINT "WORKING . . . "
2800 PRINT CHR$ (4);"OPEN"N$: PRINT CHR$ (4);"WRITE"N$
2810 PRINT NAM$
2820 IN$ = STR$ (TC): PRINT IN$
2830 IN$ = STR$ (LC): PRINT IN$:IN$ = STR$ (A): PRINT IN$:IN$ = STR$ (B): PRINT IN$
2840 FOR IT = 1 TO A
2850 IF D$(IT) = "" THEN PRINT CHR$ (127): GOTO 2870
2860 PRINT D$(IT)
2870 X = LEN (E$(IT)): PRINT STR$ (X)
2880 IF X = 0 THEN 2930
2890 FOR JI = 1 TO X
2900 Y$ = STR$ ( ASC ( MID$ (E$(IT),JI,1)))
2910 PRINT Y$
2920 NEXT
2930 NEXT
2940 FOR IT = 1 TO 100:IN$ = STR$ (K(IT))
2950 PRINT IN$: NEXT
2960 PRINT CHR$ (4)"CLOSE"N$
2970 HOME : GOSUB 700
2980 RETURN
2990 REM ENTRY CLEAR
3000 CP(1) = 1
3010 IF FN C(C) < = 0 THEN 3050
3020 J( FN R(R), FN C(C)) = 0
3030 GOSUB 1090: GOSUB 1140
3040 RETURN
3050 D$( FN R(R)) = ""
3060 GOSUB 1090: GOSUB 1140
3070 RETURN
3080 REM NEW COMMAND
3090 PRINT "ENTER:"
3100 PRINT "1) DATA"
3110 PRINT "2) LOGIC"
3120 PRINT "3) ALL - BOTH LOGIC AND DATA": PRINT
3130 PRINT "4) ABORT NEW COMMAND": PRINT "": PRINT
3140 GOSUB 5260: IF KF = 0 OR (A$ < "1" OR A$ > "4") THEN 3140
3150 PRINT ">>>>> ";: INVERSE : PRINT A$: NORMAL
3160 ON VAL (A$) GOTO 3180,3200,3220
3170 RETURN
3180 FOR IT = 1 TO A: FOR JI = 1 TO B: J(IT,JI) = 0: NEXT : NEXT : RETURN
3200 FOR IT = 1 TO A: E$(IT) = "": D$(IT) = "": NEXT
3210 RETURN
3220 FOR IT = 1 TO A: E$(IT) = "": D$(IT) = "": FOR JI = 1 TO B: J(IT,JI) = 0: NEXT : NEXT
3240 RETURN
3250 REM LIST
3260 PRINT "LOGIC NAME IS ";NAM$
3270 PRINT "TOTAL COLUMN IS ";TC
3280 PRINT "LAST COLUMN IS ";LC
3290 FOR IT = 1 TO A
3300 IF D$(IT) > "" THEN PRINT IT;" IS ";D$(IT)
3310 IF LEN (E$(IT)) > 0 THEN GOSUB 3340: PRINT STR$ (IT);" =";I$
3320 NEXT
3330 RETURN
3340 I$ = "": IF E$(IT) = "" THEN RETURN
3350 FOR JI = 1 TO LEN (E$(IT))
3360 U = ASC ( MID$ (E$(IT),JI,1)): IF U = 255 THEN RETURN
3370 I$ = I$ + " "
3380 IF U > 200 THEN I$ = I$ + N$(U - 200)
3390 IF U > 100 AND U < 200 THEN I$ = I$ + STR$ (K(U - 100)) + ")"
3400 IF U < 100 THEN I$ = I$ + STR$ (U)
3410 NEXT
3420 RETURN
3430 REM NUMBER ENTRY
3440 IF CP(1) = 1 THEN CP(1) = 0:CP(2) = 0:CP(3) = 0:TI$ = "": IF A$ = "." AND CP(2) > = 1 THEN
3450 RETURN
3460 IF A$ = "." THEN CP(2) = 1: GOSUB 1090
3470 IF CP(3) = 5 AND CP(2) = 0 THEN CP(2) = 1:TI$ = TI$ + ".": IF CP(2) = 4 THEN RETURN
3480 TI$ = TI$ + A$: J( FN R(R), FN C(C)) = VAL (TI$)
3500 IF CP(2) > = 1 THEN CP(2) = CP(2) + 1: GOTO 3520
3510 CP(3) = CP(3) + 1
3520 GOSUB 1140
3530 RETURN
3540 REM ROW NAME ENTRY
3550 IF ASC (A$) < 32 OR A$ = "," OR A$ = ";" OR A$ = ":" THEN RETURN
3560 IF CP(1) = 1 THEN CP(1) = 0:TI$ = ""
3570 IF LEN (TI$) = 10 OR A$ < " " THEN RETURN
3580 TI$ = TI$ + A$
3590 D$( FN R(R)) = TI$
3600 GOSUB 1140
3610 RETURN
3620 REM CALCULATE MODEL
3630 CP(1) = 1
3640 HOME : VTAB 13: HTAB 13
3650 PRINT "CALCULATING . . . "
3660 FOR Z = 1 TO A: IF E$(Z) = "" THEN 3950
3670 FOR ZC = 1 TO LC:J(Z,ZC) = 0: NEXT
3680 FOR Z1 = 1 TO LEN (E$(Z)):U = ASC ( MID$ (E$(Z),Z1,1))
3690 IF Z1 < > 1 THEN 3760
3700 IF U < 101 THEN U1 = U:LG = 0: GOTO 3810
3710 IF U < 201 THEN U1 = U:LG = 0: GOTO 3820
3720 IF U < > 205 THEN 3760
3730 Z1 = Z1 + 1:U1 = ASC ( MID$ (E$(Z),Z1,1))
3740 FOR ZC = 2 TO LC:J(Z,ZC) = J(Z,ZC) + J(U1,ZC - 1): NEXT
3750 GOTO 3930
3760 LG = 0:Z1 = Z1 + 1
3770 U1 = ASC ( MID$ (E$(Z),Z1,1))
3780 IF U1 < 101 THEN ON U - 200 GOTO 3810,3830,3850,3890
3790 IF U1 < 201 THEN ON U - 200 GOTO 3820,3840,3870,3910
3800 IF U1 = 205 THEN Z1 = Z1 + 1:LG = LG + U1: GOTO 3770
3810 FOR ZC = 1 + LG TO LC:J(Z,ZC) = J(Z,ZC) + J(U1,ZC - LG): NEXT : GOTO 3930
3820 FOR ZC = 1 + LG TO LC:J(Z,ZC) = J(Z,ZC) + K(U1 - 100): NEXT : GOTO 3930
3830 FOR ZC = 1 + LG TO LC:J(Z,ZC) = J(Z,ZC) - J(U1,ZC - LG): NEXT : GOTO 3930
3840 FOR ZC = 1 + LG TO LC:J(Z,ZC) = J(Z,ZC) - K(U1 - 100): NEXT : GOTO 3930
```

Continued

```
3850 FOR ZC = 1 + LG TO LC:J(Z,ZC) = J(Z
     ,ZC) * J(U1,ZC - LG)
3860 NEXT : GOTO 3930
3870 FOR ZC = 1 + LG TO LC:J(Z,ZC) = J(Z
     ,ZC) * K(U1 - 100)
3880 NEXT : GOTO 3930
3890 FOR ZC = 1 + LG TO LC: IF J(U1,ZC -
     LG) = 0 THEN J(Z,ZC) = J(Z,ZC)
     / J(U1,ZC - LG)
3900 NEXT : GOTO 3930
3910 FOR ZC = 1 + LG TO LC: IF K(U1 - 10
     0) < 0 THEN J(Z,ZC) = J(Z,ZC) /
     K(U1 - 100) > 0 THEN J(Z,ZC) = J(Z,ZC)
3920 NEXT : GOTO 3930
3930 IF ABS (J(Z,ZC)) > 99999.99 THEN J
     (Z,ZC) = SGN (J(Z,ZC)) * 99999.99
3940 NEXT Z1
3950 NEXT Z
3960 IF TC = 0 THEN RETURN
3970 FOR Z = 1 TO A:J(Z,TC) = 0: NEXT
3980 FOR Z = 1 TO A: FOR Z1 = 1 TO LC
3990 J(Z,TC) = J(Z,TC) + J(Z,Z1): NEXT
4000 IF ABS (J(Z,TC)) > 99999.99 THEN J
     (Z,TC) = SGN (J(Z,TC)) * 99999.99
4010 NEXT
4020 HOME : GOSUB 700: GOSUB 1140
4030 RETURN
4040 REM LOGIC ENTRY
4050 CP(1) = 1
4060 HOME : VTAB 1: HTAB 11: PRINT "***
     LOGIC MODE ***";
4070 VTAB 2
4080 PRINT "": HTAB 1: INPUT ">";L$:L =
     LEN (L$): P = 0
4090 X = 0
4100 IF L = 0 THEN PRINT "": GOTO 4060
4110 FOR IT = 1 TO L:Y$ = MID$ (L$,IT,1
     )
4120 IF ASC (Y$) < 32 OR Y$ = "," OR Y$
     = ";" OR Y$ = ":" THEN GOSUB 4740
     : GOTO 4080
4130 NEXT
4140 FOR IT = 1 TO 7
4150 IF LEFT$ (L$, LEN (LG$(IT))) = LG$
     (IT) THEN X = IT
4160 NEXT : IF X = 0 THEN 4190
4170 ON X GOSUB 4490,4520,3080,3250,4770
     ,4580,4600
4180 GOTO 4080
4190 FOR IT = 1 TO 130:F$(IT) = "": NEXT

4200 FOR IT = 1 TO 130
4210 FOR JI = P + 1 TO LEN (L$):JP = JI
4220 IF MID$ (L$,JP,1) = " " THEN F$(IT
     ) = MID$ (L$,P + 1,JP - P - 1):P =
     JP: GOTO 4250
4230 F$(IT) = RIGHT$ (L$,( LEN (L$) - P
     ))
4240 NEXT JI
4250 IF JP = LEN (L$) THEN F = IT: GOTO
     4270
4260 NEXT IT: PRINT "!!!! LOGIC SENTENCE
     MAY BE TRUNCATED !!!!": PRINT
4270 IF F$(1) = "MOVE" THEN GOSUB 4700:
     GOTO 4080
4280 LN = VAL (F$(1))
4290 CP(1) = 1
4300 IF (LN < 1 OR (LN > A)) THEN GOSUB
     4670: GOTO 4080
4310 IF F$(2) = "=" THEN 4360
4320 IF F$(2) < > "IS" THEN GOSUB 4670
     : GOTO 4080
4330 IF F$(3) = "NULL" THEN D$(LN) = "":
     E$(LN) = "": GOTO 4080
4340 D$(LN) = LEFT$ (F$(3) + F$(4) + F$
     (5) + F$(6) + F$(7),10)
4350 GOTO 4080
4360 E$(LN) = ""
4370 FOR N = 3 TO F
4380 IF F$(N) = "LAG" THEN U = 205: GOTO
     4450
4390 IF F$(N) = "+" THEN U = 201: GOTO 4
     450
4400 IF F$(N) = "-" THEN U = 202: GOTO 4
     450
4410 IF F$(N) = "*" THEN U = 203: GOTO 4
     450
4420 IF F$(N) = "/" THEN U = 204: GOTO 4
     450
4430 IF LEFT$ (F$(N),1) = "(" THEN AB =
     AB + 1:K(AB) = VAL ( MID$ (F$(N),
     2, LEN (F$(N)) - 2)):E$(LN) = E$(LN
     ) + CHR$ (100 + AB): GOTO 4460
4440 U = VAL (F$(N)): IF (U < 1 OR U >
     AB) THEN GOSUB 4670: GOTO 4080
4450 E$(LN) = E$(LN) + CHR$ (U)
4460 NEXT : GOTO 4080
4470 RETURN
4480 REM MISCELLANEOUS LOGIC SUBROUTINES

4490 IF LEN (L$) < 14 THEN RETURN
4500 NAM$ = RIGHT$ (L$, LEN (L$) - 14)
4510 RETURN
4520 TC$ = MID$ (L$,17, LEN (L$) - 16)
4530 IF TC$ = "OFF" THEN TC = 0: GOTO 45
     70
4540 T1 = VAL (TC$): IF T1 > B THEN PR
     INT "TOTAL COLUMN IS OUT RANGE": PR
     INT "": RETURN
4550 TC = T1
4560 IF TC > 0 THEN LC = TC - 1
4570 RETURN
4580 POP
4590 RETURN
4600 L1 = VAL ( RIGHT$ (L$, LEN (L$)) -
     15)
4610 IF L1 > B THEN PRINT "LAST COLUMN
     IS OUT OF RANGE": PRINT "": RETURN

4620 LC = L1
4630 IF (LC > TC AND TC > 0) THEN LC = T
     C - 1
4640 IF TC = 0 AND LC > B THEN LC = B
4650 IF TC > 0 AND LC < TC - 1 THEN TC =
     LC + 1
4660 RETURN
4670 PRINT ""
4680 PRINT "ERROR ** CAN'T UNDERSTAND YO
     UR ENTRY": PRINT ""
4690 RETURN
4700 IF VAL (F$(2)) > A OR VAL (F$(2))
     < 1 OR VAL (F$(4)) > A OR VAL (F
     $(4)) < 1 THEN GOSUB 4670: RETURN
4710 FOR IT = 1 TO A
4720 J(IT, VAL (F$(4))) = J(IT, VAL (F$(
     2)))
4730 NEXT : RETURN
4740 PRINT ""
4750 PRINT "ERROR ** ILLEGAL CHARACTERS
     ENTERED": PRINT ""
4760 RETURN
4770 REM PRINT OUT LOGIC
4780 GOSUB 5310
4790 GOSUB 3250
4800 PR# 0
4810 RETURN
4820 REM PRINT REPORT
4830 CP(1) = 1
4840 HOME : VTAB 1: HTAB 1
4850 PRINT "ENTER REPORT TITLE"
4860 INPUT "TITLE: ";T$
4870 PRINT "": PRINT "ENTER TODAY'S DAT
     E:
4880 INPUT "DATE: ";D$: PRINT ""
4890 PRINT "HOW MANY ROWS?"
4900 INPUT "ROWS: ";RS
4910 GOSUB 5310
4920 PRINT "SNAP-CALC SPREAD SHEET"
4930 PRINT ""
4940 PRINT T$;"
                ";D$
4950 GOSUB 5230
4960 IF TC = 0 THEN PC = LC: GOTO 4980
4970 PC = TC
4980 S3 = 0
4990 FOR S1 = 1 TO PC STEP 6:S3 = S3 + 1
     :S2 = S1
5000 PRINT " PAGE "S3
5010 GOSUB 5230
5020 PS$ = "ROW :                    "
5030 FOR IT = S2 TO ((PC < S2 + 5) * PC)
     + ((PC > = S2 + 5) * (S2 + 5))
5040 IF IT > S2 THEN 5060
5050 PS$ = PS$ + RIGHT$ ("                "
     STR$ (IT)),8): GOTO 5070
5060 PS$ = PS$ + RIGHT$ ("                "
     STR$ (IT)),10)
5070 NEXT
5080 PRINT PS$: GOSUB 5230
5090 PRINT ""
5100 FOR IT = 1 TO RS: IF LEFT$ (D$(IT)
     ,1) = "*" THEN 5180
5110 PS$ = RIGHT$ ("                " + STR$ (IT)),
     2) + ":  "
5120 PS$ = PS$ + LEFT$ ((D$(IT) + "
     "),11)
5130 FOR JI = S2 TO ((PC < S2 + 5) * PC)
     + ((PC > = S2 + 5) * (S2 + 5))
5140 X = J(IT,JI): GOSUB 990
5150 PS$ = PS$ + "  " + X$
5160 NEXT
5170 PRINT PS$
5180 NEXT
5190 PRINT CHR$ (12)
5200 NEXT
5210 PR# 0: HOME
5220 GOSUB 700: GOSUB 1140: RETURN
5230 FOR ZI = 1 TO 79: PRINT "-";: NEXT
5240 PRINT "": RETURN
5250 RETURN
5260 REM GETKEY
5270 KF = PEEK (-16384) > 127
5280 ON KF + 1 GOTO 5300,5290
5290 POKE -16368,0: A$ = CHR$ ( PEEK (
     -16384))
5300 RETURN
5310 REM SET PRINT CHANNEL
5320 PR# 1: RETURN
5330 REM IN LINE 41020, SET THE
5340 REM NUMBER ACCORDING THE
5350 REM SLOT NUMBER WHERE YOUR
5360 REM PRINTER IS INSTALLED.
```

Continued

PROGRAM LISTING

SNAP-CALC

Continued

APPLE II Family

```
5370 REM    END PROGRAM?
5380 HOME : VTAB 10: HTAB 2:
5390 PRINT "ARE YOU SURE YOU WISH TO END
     THE"
5400 HTAB 2: PRINT "PROGRAM AND ERASE AL
     L DATA?": PRINT "(Y/N): ";A$
5410 HTAB 3: INPUT "(Y/N): ";A$
5420 IF A$ <> "Y" THEN 5440
5430 END
5440 RETURN
5450 REM    DATA
5460 DATA "LOGIC NAME IS","TOTAL COLUMN
     "
5470 DATA "NEW","LIST","PRINT"
5480 DATA "END","LAST COLUMN IS"
5490 DATA "+","-","*","/","LAG"
5500 REM    ERROR HANDLER
5510 REM  (REF: 30)
5520 XX = PEEK (218) + PEEK (219) * 25
     6: POKE 216,0
5530 PR# 0: HOME : VTAB 10: HTAB 1
5540 IF PEEK (222) <> 15 THEN GOSUB
     5650: GOTO 5580
5550 PRINT
5560 PRINT "ERROR ON LINE: ";XX; CHR$ (1
     3)
5570 PRINT "ERROR NUMBER: " PEEK (222);
     CHR$ (13)
5580 PRINT "IF YOU WISH TO CONTINUE," CH
     R$ (13)
5590 PRINT "PRESS <RETURN>, OTHERWISE PR
     ESS" CHR$ (13)
5600 PRINT "ANY OTHER KEY: ";
5610 GET IN$: IF IN$ = "" THEN 5610
5620 POKE 216,0: ONERR GOTO 5500
5630 IF IN$ = CHR$ (13) THEN 260
5640 END
5650 PRINT "DOS ERROR ON LINE:";XX; CHR$
     (13)
5660 PRINT "DOS ERROR NUMBER: " PEEK (22
     2); CHR$ (13)
5670 RETURN
```

HCM

SNAP-CALC

COMMODORE 64

```
100 REM ***********
110 REM * SNAP-CALC *
120 REM ***********
130 REM BY GARY STRAUSS
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.3.1
170 REM C-64 BASIC
180 CL=36:RL=20:DIM D$(CL,RL),RN$(RL),F
    $(80),K(100),E$(80)
190 TC=CL:LC=CL-1:C=0:R=1:SC=0:SR=1:SX=
    100:SY=75
200 POKE 53280,00:POKE 53281,11:PRINT"■
    CMDR GRN■"
210 GOSUB 4400:::::GOSUB 3290
220 REM MAIN CONTROL LOOP
230 GOSUB 3260
240 IF K=13 OR K=17 OR K=29 OR K=145 OR
    K=157 THEN GOSUB 3520
250 IF SC>LC THEN 270
260 IF (K=134 AND SC<>0) OR K=46 OR (K
    >48 AND K<58 AND SC<>0) THEN GOSUB
    2770
270 IF (K=134 AND SC=0) OR (K>31 AND K<
    128 AND SC=0) THEN M=1: GOSUB 261
    0
280 IF K=139 THEN GOSUB 880
290 IF K=138 THEN GOSUB 540
300 IF K=135 THEN GOSUB 3760
310 IF K=133 OR K=137 THEN GOSUB 1710
320 IF K=140 THEN GOSUB 500
330 IF K<>201 THEN 370
340 XY=1:FOR ZZ=1 TO 5
350 K=145:GOSUB 3520
360 NEXTZZ:XY=0:IF VD=1 THEN GOSUB 32
    90
370 IF K<>205 THEN 410
380 XY=1:FOR ZZ=1 TO 5
390 K=17:GOSUB 3520
400 NEXTZZ:XY=0:IF VD=1 THEN GOSUB 32
    90
410 IF K<>202 THEN 450
420 XY=1:FOR ZZ=1 TO 3
430 K=157:GOSUB 3520
440 NEXTZZ:XY=0:IF VD=1 THEN GOSUB 32
    90
450 IF K<>204 THEN 490
460 XY=1:FOR ZZ=1 TO 3
470 K=29:GOSUB 3520
480 NEXTZZ:XY=0:IF VD=1 THEN GOSUB 32
    90
490 GOTO 220
500 GOSUB 1410:::::PRINT"■SHIFT CLR■":
    PRINT"DO YOU REALLY WANT TO EXIT TH
    E PROGRAM?"
510 PRINT:PRINT"(Y/N)"
520 GOSUB 3260:::::IF K<>89 THEN GOSUB
    3290:::::RETURN
530 PRINT"■SHIFT CLR■":END
540 REM CALCULATE LOGIC MODEL
550 GOSUB 1410:::::PRINT"■SHIFT CLR■":
    H$="CALCULATE MODE":GOSUB 3140:::
    :PRINT:PRINT
560 PRINT"CALCULATING..."
570 FOR I=1 TO LC:FOR N=1 TO RL:IF E$(N
    )="" THEN 770
580 D$(I,N)=""
590 FOR M=1 TO LEN(E$(N)):U=ASC(MID$(E$
    (N),M,1))
600 IF M<LEN(E$(N)) THEN VL=ASC(MID$(E$
    (N),M+1,1))
610 GOSUB 850:::::W=VL
620 IF U=205 AND M=1 THEN VL=ASC(MID$(E
    $(N),M+1,1)):D$(I,N)=D$(I-1,VL)
630 IF U=205 AND M=1 THEN M=M+1:GOTO
    720
640 IF W=205 THEN M=M+1:IF I>1 THEN VL=
    ASC(MID$(E$(N),M+1,1)):W=VAL(D$(I-1
    ,VL))
650 IF W=205 AND I=1 THEN W=0
660 IF U=201 THEN D$(I,N)=STR$(VAL(D$(I
    ,N))+W):M=M+1:GOTO 720
670 IF U=202 THEN D$(I,N)=STR$(VAL(D$(I
    ,N))-W):M=M+1:GOTO 720
680 IF U=203 THEN D$(I,N)=STR$(INT(VAL(
    D$(I,N))*100*W)/100):M=M+1:GOTO 7
    20
690 IF U<>204 OR VAL(D$(I,N))=0 OR W=0
    THEN 710
700 D$(I,N)=STR$(INT(VAL(D$(I,N))*100/W
    )/100):M=M+1:GOTO 720
710 VL=U:GOSUB 850:::::D$(I,N)=STR$(VL
    )
720 IF VAL(D$(I,N))<100000.00 THEN 74
    0
730 D$(I,N)="99999.99"
740 IF LEFT$(D$(I,N),1)=" " THEN D$(I,N
    )=RIGHT$(D$(I,N),LEN(D$(I,N))-1)
750 IF VAL(D$(I,N))<1 AND VAL(D$(I,N))>
    0 THEN D$(I,N)="0"+D$(I,N)
760 NEXTM
770 NEXTN:NEXTI
780 FOR N=1 TO RL:D$(TC,N)="":IF TC=0 T
    HEN 840
790 FOR I=1 TO TC-1:D$(TC,N)=STR$(VAL(D
    $(TC,N))+VAL(D$(I,N))):NEXTI
800 D$(TC,N)=RIGHT$(D$(TC,N),LEN(D$(TC,
    N))-1)
810 IF VAL(D$(TC,N))<1 AND VAL(D$(TC,N)
    )>0 THEN D$(TC,N)="0"+D$(TC,N)
820 IF VAL(D$(TC,N))<100000.00 THEN 8
    40
830 D$(TC,N)="99999.99"
840 NEXTN:GOSUB 3290:::::RETURN
850 IF VL>100 AND VL<201 THEN VL=K(VL-1
    00):GOTO 870
860 IF VL<RL THEN VL=VAL(D$(I,VL))
870 RETURN
880 REM LOGIC MODE
890 GOSUB 1410:::::PRINT"■SHIFT CLR■"
900 H$="LOGIC MODE":GOSUB 3140
910 X=0:Y=38:GOSUB 4610
920 GOSUB 3160:::::PRINT:PRINT:GOSUB
    1430
930 IF F$(1)<>"TOTAL" THEN 1010
940 IF F$(4)="OFF" THEN TC=0:GOTO 920
950 MAX=CL:TEST$=F$(4):GOSUB 4360
960 IF ERR=1 THEN 1220
970 GOSUB 4360:::::IF ERR=1 THEN 122
    0:::::TC=VAL(F$(4))
980 TC=VAL(F$(4))
990 IF LC>TC THEN LC=TC-1
1000 GOTO 920
1010 IF F$(1)<>"LAST" THEN 1050
1020 TEST$=F$(4):MAX=TC-1:IF TC=0 THEN M
     AX=CL
1030 GOSUB 4360:::::IF ERR=1 THEN 123
     0
1040 LC=VAL(F$(N)):GOTO 920
1050 IF F$(1)="END" THEN GOSUB 3290:::
     :RETURN
1060 IF F$(1)<>"MOVE" THEN 1110
1070 MAX=LC:TEST$=F$(2):GOSUB 4360
1080 IF ERR=1 THEN 1230
1090 TEST$=F$(4):GOSUB 4360:::::IF ERR=
     1 THEN 1230
1100 FOR I=1 TO RL:D$(VAL(F$(4)),I)=D$(V
     AL(F$(2)),I):NEXTI:GOTO 920
1110 IF F$(1)="LOGIC" THEN LGCN$=F$(4):G
     OTO 920
1120 IF F$(1)="LIST" THEN GOSUB 1540:::
     :GOTO 920
1130 IF F$(1)="PRINT" THEN GOSUB 4620:
     :::GOSUB 1540:::::GOSUB 4640:::::
     GOTO 920
1140 IF F$(1)="NEW" THEN GOSUB 4220:::
     :GOTO 920
1150 IF F$(2)<>"IS" THEN 1210
1160 MAX=RL:TEST$=F$(1):GOSUB 4360
1170 IF ERR=1 THEN GOTO 1230
```

Continued

Continued

COMMODORE 64

```
1180 FOR I=4 TO 80:IF F$(I)="" THEN 1200
1190 F$(3)=F$(3)+" "+F$(I):NEXT
1200 RN$(VAL(F$(1)))=MID$(F$(3),1,10):GO TO 920
1210 IF F$(2)="=" THEN 1240
1220 PRINT"{RVSON}***{RVSOFF} COMMAND NOT RECOGNIZED {RVSON}***{RVSOFF}":PRINT:PRINT:GOTO 920
1230 PRINT"{RVSON}***{RVSOFF} ILLEGAL ROW OR COLUMN {RVSON}***{RVSOFF}":PRINT:PRINT:GOTO 920
1240 MAX=RL:TEST$=F$(1):GOSUB 4360
1250 IF ERR=1 THEN GOTO 1220
1260 AA=VAL(F$(1)):E$(AA)=""
1270 FOR N=3 TO 40:IF F$(N)="" THEN 92
     0
1280 IF F$(N)="LAG" THEN U=205:GOTO 13
     80
1290 IF F$(N)="+" THEN U=201:GOTO 1380
1300 IF F$(N)="-" THEN U=202:GOTO 1380
1310 IF F$(N)="*" THEN U=203:GOTO 1380
1320 IF F$(N)="/" THEN U=204:GOTO 1380
1330 IF MID$(F$(N),1,1)<>"(" THEN 1360
1340 GOSUB 1400:::::AB=AB+1:K(AB)=O
1350 E$(AA)=E$(AA)+CHR$(100+AB):GOTO 1
     390
1360 U=VAL(F$(N))
1370 IF U>200 AND F$(N+1)="" THEN E$(AA)
     ="":GOTO 1220
1380 E$(AA)=E$(AA)+CHR$(U)
1390 NEXTN:GOTO 920
1400 O=VAL(MID$(F$(N),2,LEN(F$(N))-2)):R
     ETURN
1410 REM DELETE SPRITE
1420 POKE V+21,0:RETURN
1430 REM STORE LOGIC STRING TO F$()
1440 FOR I=1 TO 80:F$(I)="":NEXTI:AD=1
1450 FOR N=1 TO 80:FOR I=AD TO 80
1460 IF I>LEN(L$) THEN AE=0:GOTO 1490
1470 IF MID$(L$,I,1)=" " THEN AE=I:GOTO
     1490
1480 NEXTI
1490 IF AE=0 THEN AE=LEN(L$)+1
1500 F$(N)=MID$(L$,AD,AE-AD):AD=AE+1
1510 IF AE>LEN(L$) THEN 1530
1520 NEXTN
1530 RETURN
1540 REM REBUILD ENTRY STRING TO PRINT R
     EPORT
1550 PRINT"LOGIC NAME IS "L$:PRINT"TO
     TAL COLUMN IS "TC
1560 PRINT"LAST COLUMN IS "LC
1570 FOR N=1 TO RL:IF RN$(N)<>"" THEN PR
     INTN" IS "RN$(N)
1580 IF E$(N)="" THEN 1700
1590 I$=STR$(N)+" =":FOR M=1 TO LEN(E$(
     N)):U=ASC(MID$(E$(N),M,1))
1600 IF U=201 THEN I$=I$+" +":GOTO 16
     80
1610 IF U=202 THEN I$=I$+" -":GOTO 16
     80
1620 IF U=203 THEN I$=I$+" *":GOTO 16
     80
1630 IF U=204 THEN I$=I$+" /":GOTO 16
     80
1640 IF U=205 THEN I$=I$+"LAG":GOTO 1
     680
1650 IF U<100 THEN 1670
1660 I$=I$+"("+RIGHT$(STR$(K(U-100)),LEN
     (STR$(K(U-100)))-1)+")":GOTO 1680
1670 I$=I$+RIGHT$(STR$(U),LEN(STR$(U))-1
     )
1680 NEXTM
1690 PRINTI$
1700 NEXTN:RETURN
1710 REM FILE HANDLING
1720 GOSUB 1410:::::IF K=133 THEN H$="L
     OAD FILE":GOTO 2160
1730 H$="SAVE FILE":PRINT"{SHIFT CLR}":G
     OSUB 3140:::::PRINT
1740 PRINT"1 - SAVE DATA":PRINT"2 - SAVE
      LOGIC":PRINT"3 - SAVE BOTH":PRINT
1750 PRINT"INPUT A NUMBER 1-3 :>{CMDR v}
     {SHIFT CRSRLEFT}";
1760 GET Q2$:IF VAL(Q2$)<1 OR VAL(Q$)>3
     THEN 1760
1770 PRINTQ2$:PRINT:ON VAL(Q2$) GOSUB
     1800,,1950,,1790
1780 GOSUB 3290:::::RETURN
1790 GOSUB 1950:::::GOSUB 1810:::::RET
     URN
1800 GOSUB 2580
1810 PRINT:PRINT"NAME OF DATA FILE :";
1820 GOSUB 3160:::::DF$=L$
1830 IF K=84 THEN 1850
1840 OPEN1,8,8,"@0:"+DF$+".D"+",S,W":GOS
     UB 1860:::::CLOSE1:RETURN
1850 OPEN1,1,1,DF$+".D":GOSUB 1860:::::
     CLOSE1:RETURN
1860 FOR I=1 TO RL
1870 T$=RN$(I):IF T$="" THEN T$="{CTRL R
     VSOFF}"
1880 PRINT#1,T$
1890 NEXTI
1900 FOR I=1 TO RL:FOR J=1 TO CL
1910 T$=D$(J,I):IF T$="" THEN T$="{CTRL
     RVSOFF}"
1920 PRINT#1,T$
1930 NEXTJ:NEXTI
1940 RETURN
1950 GOSUB 2580:::::PRINT:PRINT:IF LGCN
     $<>"" THEN 1970
1960 PRINT"NAME OF LOGIC FILE :";:GOSUB
     3160:::::LGCN$=L$
1970 IF K=84 THEN 1990
1980 OPEN 1,8,8,"@0:"+LGCN$+".L"+",S,W":
     GOSUB 2000:::::CLOSE1:RETURN
1990 OPEN 1,1,1,LGCN$+".L":GOSUB 2000:
     :::CLOSE1:RETURN
2000 PRINT#1,TC
2010 PRINT#1,LC
2020 PRINT#1,AB
2030 FOR I=1 TO AB
2040 PRINT#1,K(AB)
2050 NEXTI
2060 FOR I=1 TO RL
2070 T$=RN$(I)
2080 IF T$="" THEN T$="{CTRL RVSOFF}"
2090 PRINT#1,T$
2100 NEXTI
2110 FOR I=1 TO 80
2120 T$=E$(I):IF T$="" THEN T$="{CTRL RV
     SOFF}"
2130 PRINT#1,T$
2140 NEXTI
2150 RETURN
2160 PRINT"{SHIFT CLR}":GOSUB 3140:::::
     PRINT:PRINT"1 - LOAD DATA":PRINT"2
     - LOAD LOGIC"
2170 PRINT"3 - LOAD BOTH":PRINT
2180 PRINT"INPUT A NUMBER 1-3 :>{CMDR v}
     {SHIFT CRSRLEFT}";
2190 GET Q2$:IF VAL(Q2$)<1 OR VAL(Q2$)>3
     THEN 2190
2200 PRINTVAL(Q2$):ON VAL(Q2$) GOSUB 2
     230,,2380,,2220
2210 GOSUB 3290:::::RETURN
2220 GOSUB 2380:::::GOSUB 2240:::::RET
     URN
2230 GOSUB 2580
2240 PRINT:PRINT"NAME OF DATA FILE :";
2250 GOSUB 3160:::::DF$=L$
2260 IF K=84 THEN 2280
2270 OPEN 1,8,8,"0:"+DF$+".D"+",S,R":GOS
     UB 2290:::::CLOSE1:RETURN
2280 OPEN 1,1,0,DF$+".D":GOSUB 2290:::
     :CLOSE1:RETURN
2290 FOR I=1 TO RL
2300 INPUT#1,RN$(I)
2310 IF RN$(I)="{CTRL RVSOFF}" THEN RN$(
     I)=""
2320 NEXTI
2330 FOR I=1 TO RL:FOR J=1 TO CL
2340 IF D$(J,I)="{CTRL RVSOFF}" THEN D$(
     J,I)=""
2350 INPUT#1,D$(J,I)
2360 NEXTJ:NEXTI
2370 RETURN
2380 GOSUB 2580:::::PRINT:PRINT"NAME OF
      LOGIC FILE :";
2390 GOSUB 3160:::::LGCN$=L$
2400 IF K=84 THEN 2420
2410 OPEN 1,8,8,"0:"+LGCN$+".L"+",S,R":G
     OSUB 2430:::::CLOSE1:RETURN
2420 OPEN 1,1,0,LGCN$+".L":GOSUB 2430:
     :::CLOSE1:RETURN
2430 INPUT#1,TC
2440 INPUT#1,LC
2450 INPUT#1,AB
2460 FOR I=1 TO AB
2470 INPUT#1,K(I)
2480 NEXTI
2490 FOR I=1 TO RL
2500 INPUT#1,RN$(I)
2510 IF RN$(I)="{CTRL RVSOFF}" THEN RN$(
     I)=""
2520 NEXTI
2530 FOR I=1 TO 80
2540 INPUT#1,E$(I)
2550 IF E$(I)="{CTRL RVSOFF}" THEN E$(I)
     =""
2560 NEXTI
2570 RETURN
2580 PRINT:PRINT"TAPE OR DISK?":PRINT"(T
     /D)"
2590 GOSUB 3260:::::IF K<>84 AND K<>68
     THEN 2590
2600 RETURN
2610 REM BUILD ROW NAMES
2620 RN$(SR)="":X=5:Y=(SR-R)*2+4:GOSUB
     4610:::::PRINT"
2630 I=1
2640 IF K=134 THEN RN$(SR)="":GOSUB 32
     90
2650 IF K=134 THEN 2760
2660 IF K>132 AND K<141 AND K<>136 OR K=
     201 OR K=202 OR K=204 OR K=205 THEN
     2760
2670 IF K=13 OR K=17 OR K=29 OR K=145 OR
      K=157 THEN GOSUB 3520:::::GOTO
     2760
2680 IF K<>20 OR I=1 THEN 2720
2690 X=3+I:GOSUB 4610:::::PRINT"
2700 RN$(SR)=LEFT$(RN$(SR),LEN(RN$(SR))-
     1)
2710 I=I-1:GOSUB 3260:::::GOTO 2640
```

Continued

PROGRAM LISTING

PROGRAM LISTING

```
2720 IF I=11 THEN GOSUB 3260:::GOTO 2640
2730 IF K<32 OR K>127 THEN GOSUB 3260:::GOTO 2640
2740 I=I+1:RN$(SR)=RN$(SR)+CHR$(K)
2750 X=5:GOSUB 4610::::PRINTRN$(SR):GOSUB 3260:::::GOTO 2640
2760 RETURN
2770 REM DATA ENTRY FOR CELLS
2780 I=2:D$(SC,SR)="  ":GOSUB 4320:::::PRINT"  0.00":I=0
2790 IF K=134 THEN D$(SC,SR)="":GOSUB 3290
2800 IF K=134 THEN 3130
2810 IF K>132 AND K<141 AND K<>136 OR K=201 OR K=202 OR K=204 OR K=205 THEN 3130
2820 IF K=13 OR K=17 OR K=29 OR K=145 OR K=157 THEN GOSUB 3520:::GOTO 3130
2830 D$(SC,SR)=CHR$(K):IF K=46 THEN D$(SC,SR)="0"+D$(SC,SR):GOTO 2990
2840 GOSUB 4320:::::PRINTVAL(D$(SC,SR))
2850 GOSUB 3260
2860 IF K=134 THEN D$(SC,SR)="":GOSUB 3290
2870 IF K=134 THEN 3130
2880 IF K>132 AND K<141 AND K<>136 OR K=201 OR K=202 OR K=204 OR K=205 THEN 3130
2890 IF K=13 OR K=17 OR K=29 OR K=145 OR K=157 THEN GOSUB 3520:::GOTO 3130
2900 IF LEN(D$(SC,SR))=5 AND K<>46 THEN 2850
2910 IF K<>46 AND (K<48 OR K>57) THEN GOTO 2850
2920 D$(SC,SR)=D$(SC,SR)+CHR$(K)
2930 IF K=46 THEN 2990
2940 IF VAL(D$(SC,SR))<=99999.99 THEN 2970
2950 D$(SC,SR)=RIGHT$(D$(SC,SR),LEN(D$(SC,SR))-1)
2960 GOTO 2850
2970 GOSUB 4320:::::PRINTVAL(D$(SC,SR))
2980 GOTO 2850
2990 POKE 198,0:FOR I=1 TO 2
3000 IF K=134 THEN D$(SC,SR)="":GOSUB 3290
3010 IF K=134 THEN 3130
3020 IF K>132 AND K<141 AND K<>136 THEN 3130
3030 IF K=13 OR K=17 OR K=29 OR K=145 OR K=157 THEN GOSUB 3520:::GOTO 3130
3040 IF K<48 OR K>57 THEN GOSUB 3260:::GOTO 3000
3050 D$(SC,SR)=D$(SC,SR)+CHR$(K):GOSUB 4320:::::PRINTD$(SC,SR)
3060 GOSUB 3260
3070 NEXT
3080 POKE 198,0:IF K=134 THEN D$(SC,SR)="":GOSUB 3290
3090 IF K=134 THEN 3130
3100 IF K=13 OR K=17 OR K=29 OR K=145 OR K=157 THEN GOSUB 3520:::GOTO 3130
3110 GOSUB 3260:::::IF K>132 AND K<141 AND K<>136 THEN 3130
3120 GOTO 3080
3130 RETURN
3140 REM DISPLAY MODE ON SCREEN
3150 PRINT"■HOME■"TAB(INT((40-LEN(H$))/2))H$:RETURN
3160 REM INPUT FOR LOGIC ENTRY STRING
3170 L$="":PRINT">■CMDR v■■SHIFT CRSRLEFT■";
3180 GET Q$:IF Q$="" THEN 3180
3190 IF ASC(Q$)=13 AND L$<>"" THEN PRINT"  ":GOTO 3250
3200 IF ASC(Q$)=20 AND L$<>"" THEN PRINT"  ■2SHIFT CRSRLEFT■■CMDR v■■SHIFT CRSRLEFT■";:L$=LEFT$(L$,LEN(L$)-1)
3210 IF ASC(Q$)=32 AND L$<>"" THEN 3240
3220 IF ASC(Q$)<32 OR ASC(Q$)>127 THEN 3180
3230 IF LEN(L$)=80 THEN 3180
3240 PRINTQ$"■CMDR v■■SHIFT CRSRLEFT■";:L$=L$+Q$:GOTO 3180
3250 RETURN
3260 REM KEY SCAN ROUTINE
3270 GET KEY$:IF KEY$="" THEN 3270
3280 K=ASC(KEY$):POKE 198,0:RETURN
3290 REM CLR AND SET SCREEN
3300 PRINT"■SHIFT CLR■":H$="SNAP-CALC":GOSUB 3140:::PRINT:PRINT"ROW ";::POKE V+21,1
3310 IF C<>0 THEN GOTO 3350
3320 PRINT"  NAME 1
  2":X=0
3330 FOR I=1 TO 10:X=0:Y=(I+1)*2:GOSUB 4610:::::
3340 PRINTR+I-1:X=3:GOSUB 4610:::::PRINT")  "RN$(R+I-1):NEXT:GOTO 3390
3350 FOR I=1 TO 3:X=I*11:Y=2:GOSUB 4610
3360 PRINTC+I-1:NEXT
3370 FOR I=1 TO 10:X=0:Y=(I+1)*2:GOSUB 4610
3380 PRINTR+I-1:X=3:GOSUB 4610:::::PRINT")":NEXTI
3390 M=1:IF C=0 THEN M=2
3400 FOR I=M TO 3
3410 FOR J=1 TO 10
3420 X=I*11:Y=J*2+2:GOSUB 4610
3430 PRINT"0.00":T=VAL(D$(C+I-1,R+J-1))):TEMP=ABS(INT(T)):Z=1:X=X+1
3440 IF TEMP<10 THEN 3470
3450 TEMP=ABS(INT(TEMP/10)):Z=Z+1:IFZ=5 THEN 3470
3460 GOTO 3440
3470 X=X-Z:IF VAL(D$(C+I-1,R+J-1))<0 THEN X=X-1
3480 GOSUB 4610
3490 PRINTD$(C+I-1,R+J-1)
3500 NEXT:NEXT
3510 RETURN
3520 REM MOVE CURSOR
3530 VD=0:IF K<>17 AND K<>13 THEN 3580
3540 IF SR=RL THEN VD=1:GOTO 3750
3550 IF SY=219 THEN R=R+1:SR=SR+1:VD=1:IF XY<>1 THEN GOSUB 3290
3560 IF SY=219 THEN 3750
3570 SY=SY+16:SR=SR+1:GOTO 3740
3580 IF K<>29 THEN 3640
3590 IF SC=TC THEN VD=1:GOTO 3750
3600 IF SX=21 THEN SC=SC+1:C=C+1:VD=1:IF XY<>1 THEN GOSUB 3290
3610 IF SX=21 THEN 3750
3620 SX=SX+88:IF SX=276 THEN SX=21:POKE V+16,1
3630 SC=SC+1:GOTO 3740
3640 IF K<>145 THEN 3690
3650 IF SR=1 THEN VD=1:GOTO 3750
3660 IF SY=75 THEN R=R-1:SR=SR-1:VD=1:IF XY<>1 THEN GOSUB 3290
3670 IF SY=75 THEN 3750
3680 SY=SY-16:SR=SR-1:GOTO 3740
3690 IF SC=0 THEN VD=1:GOTO 3750
3700 IF SX=100 THEN C=C-1:SC=SC-1:VD=1:IF XY<>1 THEN GOSUB 3290
3710 IF SX=100 THEN 3750
3720 SX=SX-88:IF SX=-67 THEN SX=188:POKE V+16,0
3730 SC=SC-1
3740 POKE V,SX:POKE V+1,SY
3750 RETURN
3760 REM PRINT REPORT
3770 GOSUB 1410
3780 PO=1:PG=1:PRINT"■SHIFT CLR■":H$="PRINT REPORT":GOSUB 3140:::::PRINT:PRINT
3790 PRINT"TITLE OF REPORT:  ";::GOSUB 3160:::PRINT:TL$=L$
3800 PRINT"DATE:  ";::GOSUB 3160:::::PRINT:DT$=L$
3810 PRINT"LAST ROW TO BE PRINTED:  ";::GOSUB 3160:::::PRINT
3820 IF VAL(L$)>RL THEN L$=STR$(RL)
3830 IF VAL(L$)=0 THEN RETURN
3840 PRL=VAL(L$):PCL=TC:IF TC=0 THEN PCL=LC
3850 L=PCL:IF L>6 THEN L=6
3860 GOSUB 4620
3870 PRINTTL$"                    "DT$:PRINT"PAGE 1":PRINT
3880 PRINT"ROW :  ";::FOR I=1 TO 6:PRINT"  ";I;::NEXT:PRINT:PRINT
3890 FOR I=1 TO PRL
3900 PRINTI")  ";::PRN$=RN$(I)
3910 IF LEFT$(PRN$,1)="*" THEN PRN$=""
3920 IF LEN(PRN$)=10 THEN 3940
3930 FOR M=1 TO 10-LEN(PRN$):PRN$=PRN$+" ":NEXT
3940 PRINTPRN$"  ";
3950 FOR J=PO TO L:PD$=D$(J,I)
3960 IF LEN(PD$)=8 THEN 4060
3970 IF PD$="" THEN PD$="  0.00":GOTO 4060
3980 FOR M=1 TO LEN(PD$)
3990 IF MID$(PD$,M,1)="." THEN 4010
4000 NEXT:PD$=PD$+".00":IF LEN(PS)=8 THEN 4060
4010 IF LEN(PD$)-M=1 AND M<>1 THEN PD$=PD$+"0"
4020 IF M=LEN(PD$) THEN PD$=PD$+"00"
4030 IF LEN(PD$)-M=1 AND M=1 THEN PD$="0"+PD$
4040 IF LEN(PD$)=8 THEN 4060
4050 FOR N=1 TO 6-M:PD$="  "+PD$:NEXTN
4060 PRINTPD$"  ";
4070 NEXTJ:IF J>PCL THEN PRINT
4080 PRINT
4090 NEXTI
4100 IF J>PCL THEN 4210
4110 FOR M=1 TO 65-(4+(2*PRL)):PRINT:NEXT
4120 PG=PG+1:PRINTTL$"                    "DT$:PRINT"PAGE"PG:PRINT
4130 PRINT"ROW :  ";
4140 FOR M=J TO J+5:IF M>PCL THEN 4170
4150 IF M<10 THEN PRINT"  ";
4160 PRINT"  "M;
```

Continued

SNAP-CALC

Continued

COMMODORE 64

```
4170 NEXTM
4180 PRINT:PRINT:PO=PO+6:IF PO>PCL THEN
     PO=PCL
4190 L=L+6:IF L>PCL THEN L=PCL
4200 GOTO 3890
4210 GOSUB 4640:::::GOSUB 3290:::::RET
     URN
4220 REM CLEAR LOGIC MODEL FROM MEMORY
4230 PRINT"1 - ERASE DATA":PRINT"2 - ERA
     SE LOGIC":PRINT"3 - ERASE BOTH DATA
     AND
4240 PRINT"4 - ABORT NEW COMMAND":PRINT:
     PRINT"INPUT A NUMBER 1-4 : >■CMDR v
     ■■SHIFT CRSRLEFT■";
4250 GETQ1$:IFQ1$=""THEN 4250
4260 IF VAL(Q1$)<1 OR VAL(Q1$)>4 THEN
     4250
4270 PRINTQ1$:ON VAL(Q1$) GOSUB 4290,
     4300,,4310,&3000
4280 PRINT"■SHIFT CLR■":RETURN
4290 FOR I=1 TO TC:FOR J=1 TO RL:RN$(J)=
     "":D$(I,J)="":NEXTJ:NEXTI:RETURN
4300 FOR I=1 TO RL:E$(I)="":NEXT:TC=CL:L
     C=TC-1:AB=0:LGCN$="":RETURN
4310 GOSUB 4290:::::GOSUB 4300:::::RET
     URN
4320 REM CALCULATE POSITION IN MATRIX
4330 X=((SX+(PEEK(V+16)*255)-100)/88+1)*
     11-LEN(D$(SC,SR))
4340 IF I<>0 THEN X=X+I+2
4350 Y=((SY-75)/16+2)*2:GOSUB 4610:::::
     RETURN
4360 REM ERROR CHECK
4370 ERR=0:FOR I=1 TO MAX:IF VAL(TEST$)=
     I THEN 4390
4380 NEXT:ERR=1
4390 RETURN
4400 REM STORE SPRITE DATA
4410 AD=832:V=53248
4420 READ M:READ PC:READ M:READ M
4430 FOR I=0 TO 63:READ M:POKE AD+I,M:NE
     XT
4440 POKE 2040,13:POKE V,SX:POKE V+1,SY
4450 POKE V+39,PC:POKE V+29,1
4460 POKE V+16,0:RETURN
4470 DATA 128,12,0,0,0
4480 DATA 0,0,0,0,0,0
4490 DATA 0,0,0,255,255,255
4500 DATA 255,255,255,255,255,0
4510 DATA 0,0,0,0,0,0
4520 DATA 0,0,0,0,0,0
4530 DATA 0,0,0,0,0,0
4540 DATA 0,0,0,0,0,0
4550 DATA 0,0,0,0,0,0
4560 DATA 0,0,0,0,0,0
4570 DATA 0,0,255,255,255,255
4580 DATA 255,255,255,0,0,0
4590 DATA 0,0,0,0,0,0
4600 DATA 0,0,0
4610 PRINT"■HOME■"LEFT$("■24CRSRDOWN■",Y
     )TAB(X);:RETURN
4620 REM OPEN PRINTER PORT
4630 OPEN4,4:CMD4:RETURN
4640 REM CLOSE PRINTER PORT
4650 PRINT#4:CLOSE4:RETURN
```

HCM

SNAP-CALC

IBM PC & PCjr

```
100 REM ***************
110 REM * SNAP-CALC *
120 REM ***************
130 REM BY GARY STRAUSS AND
140 REM THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.3.1
170 REM IBM PC & PCjr
180 REM CASSETTE OR CARTRIDGE BASIC
190 REM OR BASICA
200 REM
210 CLS:LOCATE 12,15:PRINT "SNAP-CALC":
    OPTION BASE 1:A=60:B=30:TC=13:LC=12
    :CL=0:RW=1:US$="####.##":DIM D$(A)
    ,E$(A),F$(130),J(A,B),K(100):KEY OF
    F
220 RP$=" ## ! \ "+US$+" "+US$+" "+US
    $+" "+US$+" "+US$+" "+US$+" "+U
    S$
230 ON KEY(1) GOSUB 450:ON KEY(2) GOSUB
    570:ON KEY(3) GOSUB 660:ON KEY(5)
    GOSUB 670:ON KEY(6) GOSUB 850:ON KE
    Y(7) GOSUB 920:ON KEY(9) GOSUB 1630
240 ON KEY(11) GOSUB 1340:ON KEY(12) GO
    SUB 1360:ON KEY(13) GOSUB 1380:ON K
    EY(14) GOSUB 1400:ON ERROR GOTO 160
    0
250 LOCATE 24,8:PRINT "PRESS ENTER TO C
    ONTINUE";:GOSUB 1500:RESTORE 1620:R
    EAD N$(1),N$(2),N$(3),N$(4),N$(5):D
    EF FNR=RW+R-1:DEF FNC=CL+C-1:R=1:C=
    1
260 KEY 15,CHR$(&H40)+CHR$(&HC1):ON KEY
    (15) GOSUB 1400
270 KEY(15) ON:CLS:GOSUB 350:IF E$(FNR)
    >"" AND FNR<A THEN GOSUB 430:GOSUB
    1400 ELSE IF E$(FNR)>"" AND FNR=A T
    HEN GOSUB 430:GOSUB 1340 ELSE GOSUB
    430
280 DEF SEG=0:POKE 1050,PEEK(1052):GOSU
    B 1510:IF FNC>0 THEN IF (K$>="0" AN
    D K$<="9") OR K$="." THEN GOSUB 300
    ELSE GOTO 280 ELSE GOSUB 330
290 GOTO 280
300 IF CP=1 THEN TI$="":CP=0 ELSE IF VA
    L(TI$+K$)>99999.99 THEN RETURN
310 TI$=TI$+K$:J(FNR,FNC)=VAL(TI$)
320 LOCATE R*2+3,(C-1)*12+5:PRINT USING
    US$;J(FNR,FNC);:RETURN
330 IF CP=1 THEN TI$="":CP=0 ELSE IF LE
    N(TI$)=10 THEN RETURN
340 TI$=TI$+K$:LOCATE R*2+3,5:PRINT TI$
    +SPACE$(10-LEN(TI$)):D$(FNR)=TI$:RE
    TURN
350 LOCATE 1,1:PRINT NAM$;:LOCATE 3,4:P
    RINT CHR$(201);:FOR Z=1 TO 35:PRINT
    CHR$(205);:NEXT:FOR Z=4 TO 23:LOCA
    TE Z,4:PRINT CHR$(186):NEXT
360 IF CL=0 THEN LOCATE 2,5:PRINT "ROW
    NAME ELSE LOCATE 2,5:PRINT "
    CL;"
370 LOCATE 2,21:PRINT CL+1;TAB(33);CL+2
    :FOR Z=1 TO 10:R$=STR$(INT(RW+Z-1))
    :LOCATE Z*2+3,4-LEN(R$):PRINT R$;:N
    EXT
380 IF CL>0 THEN 420
390 FOR Z=1 TO 10:LOCATE Z*2+3,5:IF LEN
    (D$(Z+RW-1))=0 THEN PRINT Z+RW-1;"
    "; ELSE PRINT D$(Z+RW-1)+SPAC
    E$(10-LEN(D$(RW+Z-1))));
400 NEXT
410 FOR Z=1 TO 10:LOCATE Z*2+3,17:PRINT
    USING US$;J(RW+Z-1,CL+1):LOCATE Z*
    2+3,29:PRINT USING US$;J(RW+Z-1,CL+
    2):NEXT:RETURN
420 FOR Z=1 TO 10:LOCATE Z*2+3,5:PRINT
    USING US$;J(RW+Z-1,CL);:PRINT " ";
    LOCATE Z*2+3,17:PRINT USING US$;J(R
    W+Z-1,CL+1):LOCATE Z*2+3,29:PRINT U
    SING US$;J(RW+Z-1,CL+2);:NEXT:RETUR
    N
430 LOCATE R*2+2,(C-1)*12+5:PRINT STRIN
    G$(8,196);:LOCATE R*2+4,(C-1)*12+5:
    PRINT STRING$(8,196);:RETURN
440 LOCATE R*2+2,(C-1)*12+5:PRINT "
    ";:LOCATE R*2+4,(C-1)*12+5:PRIN
    T "
450 CLS:LOCATE 1,1:PRINT "ENTER YOUR CH
    OICE:":PRINT:PRINT "1) LOAD DATA":P
    RINT "2) LOAD LOGIC"
460 K$=INKEY$:IF K$<"1" OR K$>"2" THEN
    460 ELSE ON VAL(K$) GOTO 470,500
470 LOCATE 10,1:INPUT "ENTER FILE NAME:
    ",N$:N$=LEFT$(N$,8)+".HCD"
480 OPEN N$ FOR INPUT AS #1
490 FOR Z=1 TO A:FOR Z1=1 TO B:INPUT #1
    ,J(Z,Z1):NEXT:NEXT:CLOSE #1:RETURN
    270
500 IF NAM$="" THEN LOCATE 10,1:INPUT "
    ENTER LOGIC NAME:",NAM$:N$=LEFT$(NA
    M$,8)+".HCL"
510 OPEN N$ FOR INPUT AS #1
520 INPUT #1,NAM1$,S1,TC1,LC1,A1,B1:IF
    A1>A OR B1>B THEN LOCATE 12,1:PRINT
    "THE LOGIC IS TOO LARGE. CHANGE LI
    NE 210.A =";A1;" AND B =";B1:CLOSE #
    1:RETURN 270
530 IF A1<A THEN A=A1
540 IF B1<B THEN B=B1
550 FOR Z=1 TO A:INPUT #1,D$(Z),E$(Z):N
    EXT
560 FOR Z=1 TO 100:INPUT #1,K(Z):NEXT:C
    LOSE #1:RETURN 270
570 CLS:LOCATE 1,1:PRINT "ENTER YOUR CH
    OICE:":PRINT:PRINT "1) SAVE DATA":P
    RINT "2) SAVE LOGIC"
580 K$=INKEY$:IF K$<"1" OR K$>"2" THEN
    580 ELSE ON VAL(K$) GOTO 590,620
590 LOCATE 10,1:INPUT "ENTER FILE NAME:
    ",N$:N$=LEFT$(N$,8)+".HCD"
600 OPEN N$ FOR OUTPUT AS #1
610 FOR Z=1 TO A:FOR Z1=1 TO B:WRITE #1
    ,J(Z,Z1):NEXT:NEXT:CLOSE #1:RETURN
    270
620 IF NAM$="" THEN LOCATE 10,1:INPUT "
    ENTER LOGIC NAME:",NAM$:N$=LEFT$(NA
    M$,8)+".HCL"
630 OPEN N$ FOR OUTPUT AS #1
640 FOR Z=1 TO A:WRITE #1,D$(Z),E$(Z):N
    EXT
650 FOR Z=1 TO 100:WRITE #1,K(Z):NEXT:W
    RITE #1,NAM$,S,TC,LC,A,B:CLOSE #1:R
    ETURN 270
```

Continued

PROGRAM LISTING

```
660 T1$="":IF FNC>0 THEN J(FNR,FNC)=0:GOSUB 360:GOSUB 430:RETURN 280 ELSE D$(FNR)="":GOSUB 360:GOSUB 430:RETURN 280
670 LOCATE 25,13:PRINT "CALCULATING.....";:FOR Z=1 TO S:IF E$(Z)="" THEN 830
680 FOR ZC=1 TO LC:J(Z,ZC)=0:NEXT:FOR Z=1 TO LEN(E$(Z)):U=ASC(MID$(E$(Z),Z1,1)):IF Z1=1 THEN U=U-101:IF U<0 THEN U1=U:LG=0:GOTO 720 ELSE IF U<201 THEN U1=U:LG=0:GOTO 730
690 IF U=205 AND Z1=1 THEN Z1=Z1+1:U1=ASC(MID$(E$(Z),Z1,1)):FOR ZC=2 TO LC:J(Z,ZC)=J(Z,ZC)+J(U1,ZC-1):NEXT:GOTO 820
700 LG=0:Z1=Z1+1
710 U1=ASC(MID$(E$(Z),Z1,1)):IF U1<101 THEN ON U-200 GOTO 720,740,760,810 ELSE IF U1<201 THEN ON U GOTO 730,750,780,800 ELSE IF U1=205 THEN Z1=Z1+1:LG=LG+1:GOTO 710
720 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)+J(U1,ZC-LG):NEXT:GOTO 820
730 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)+AB(U1-100):NEXT:GOTO 820
740 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)-J(U1,ZC-LG):NEXT:GOTO 820
750 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)-AB(U1-100):NEXT:GOTO 820
760 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)*J(U1,ZC-LG):IF J(Z,ZC)>99999.99 THEN J(Z,ZC)=99999.99
770 NEXT:GOTO 820
780 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)*AB(U1-100):IF J(Z,ZC)>99999.99 THEN J(Z,ZC)=99999.99
790 NEXT:GOTO 820
800 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)/AB(U1-100):NEXT:GOTO 820
810 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)/J(U1,ZC-LG):NEXT:GOTO 820
820 NEXT Z1
830 NEXT Z
840 IF TC=0 THEN RETURN 270 ELSE FOR Z=1 TO A:J(Z,TC)=0:NEXT:FOR Z=1 TO A:FOR Z1=1 TO LC:J(Z,TC)=J(Z,TC)+J(Z,Z1):NEXT:NEXT:RETURN 270
850 CLS:LOCATE 1,1:PRINT "ENTER REPORT TITLE:";:INPUT "TITLE",T$:PRINT:PRINT "ENTER TODAYS DATE:";:INPUT "DATE",D$:PRINT:PRINT "HOW MANY ROWS:";:INPUT "ROWS",RWS:PG=1:ON ERROR GOTO 910
860 PRINT:PRINT "DO YOU WANT A FORM FEED WITH EACH PAGE (Y/N)?";:INPUT FF$:IF FF$="Y" THEN FF=12 ELSE IF FF$="N" THEN FF=13 ELSE GOTO 860
870 LPRINT CHR$(15);"SNAP-CALC SPREAD SHEET":LPRINT:LPRINT T$;TAB(40);D$:LPRINT STRING$(132,61):LPRINT PG:LPRINT STRING$(132,61):LPRINT
880 LPRINT "ROW";ROW NAME";TAB(22);PG;TAB(32);PG+1;TAB(42);PG+2;TAB(52);PG+3;TAB(62);PG+4;TAB(72);PG+5;TAB(82);PG+6;TAB(92);PG+7;TAB(102);PG+8;TAB(112);PG+9
890 FOR Z=1 TO RWS:LPRINT " ";:LPRINT USING RP$;Z,D$(Z),J(Z,PG),J(Z,PG+1),J(Z,PG+2),J(Z,PG+3),J(Z,PG+4),J(Z,PG+5),J(Z,PG+6),J(Z,PG+7),J(Z,PG+8),J(Z,PG+9)
900 NEXT Z:IF PG<B-9 THEN PG=PG+10:LPRINT CHR$(FF):GOTO 870 ELSE ON ERROR GOTO 1600:RETURN 270
910 IF ERR<>68 THEN RESUME NEXT ELSE RESUME 270
920 GOSUB 1550:CLS:LOCATE 25,11:PRINT "*** LOGIC MODE ***";
930 LOCATE 24,1:INPUT ">",L$:L=LEN(L$):P=0
940 RESTORE 1610:FOR Z=1 TO 7:READ LN$:IF INSTR(L$,LN$)=1 THEN ON Z GOTO 1110,1120,1140,1190,1260,1330,1300
950 NEXT
960 FOR Z=1 TO 130:FS=INSTR(P+1,L$," ")
:IF FS>0 THEN F$(Z)=MID$(L$,P+1,FS-P-1) ELSE F$(Z)=RIGHT$(L$,LEN(L$)-P)
970 IF FS=0 THEN F=Z:GOTO 990
980 P=FS:NEXT:PRINT "!!! LOGIC SENTENCE MAY BE TRUNCATED !!!":PRINT
990 IF F$(1)="MOVE" THEN 1290 ELSE LN=VAL(F$(1)):IF LN<1 OR LN>A THEN 1320 ELSE IF F$(2)="=" THEN 1000 ELSE IF F F$(2)<>"IS" THEN 1320 ELSE IF F$(3)="NULL" THEN D$(LN)="":E$(LN)="":GOTO 930 ELSE D$(LN)=LEFT$(F$(3)+F$(4)+F$(5)+F$(6)+F$(7),10):GOTO 930
1000 E$(LN)="":S=((LN>S)*LN*(-1))+((S>LN)*S*(-1)):IF S=0 THEN S=LN
1010 FOR N=3 TO F
1020 IF F$(N)="LAG" THEN U=205:GOTO 1090
1030 IF F$(N)="+" THEN U=201:GOTO 1090
1040 IF F$(N)="-" THEN U=202:GOTO 1090
1050 IF F$(N)="*" THEN U=203:GOTO 1090
1060 IF F$(N)="/" THEN U=204:GOTO 1090
1070 IF LEFT$(F$(N),1)="(" THEN AB=AB+1:K(AB)=VAL(MID$(F$(N),2,LEN(F$(N))-2)):E$(LN)=E$(LN)+CHR$(100+AB):GOTO 1100
1080 U=VAL(F$(N)):IF U<1 OR U>A THEN 1320
1090 E$(LN)=E$(LN)+CHR$(U)
1100 NEXT:GOTO 930
1110 NAM$=RIGHT$(L$,LEN(L$)-14):GOTO 930
1120 TC$=MID$(L$,17,LEN(L$)-16):IF TC$="OFF" THEN TC=0 ELSE TC=VAL(TC$):IF TC>0 THEN LC=TC-1
1130 GOTO 930
1140 PRINT "ENTER:":PRINT "1) DATA":PRINT "2) LOGIC":PRINT "3) ALL - BOTH LOGIC AND DATA":PRINT "4) ABORT NEW COMMAND":PRINT:PRINT
1150 K1$=INKEY$:IF K1$<"1" OR K1$>"4" THEN 1150 ELSE ON VAL(K1$) GOTO 1160,1170,1180,930
1160 FOR Z=1 TO 40:FOR Z1=1 TO 13:J(Z,Z1)=0:NEXT:NEXT:GOTO 930
1170 FOR Z=1 TO 40:E$(Z)="":NEXT:GOTO 930
1180 FOR Z=1 TO 40:FOR Z1=1 TO 13:J(Z,Z1)=0:NEXT:E$(Z)="":NEXT:GOTO 930
1190 PRINT "LOGIC NAME IS ";NAM$:PRINT "TOTAL COLUMN IS";TC:PRINT "LAST COLUMN IS";LC:FOR Z=1 TO 40
1200 IF D$(Z)>"" THEN PRINT Z;" IS ";D$(Z)
1210 IF E$(Z)>"" THEN GOSUB 1230:PRINT STR$(Z);" = ";I$
1220 NEXT:GOTO 930
1230 I$="":FOR X=1 TO LEN(E$(Z)):U=ASC(MID$(E$(Z),X,1)):IF U=255 THEN RETURN ELSE IF U>200 THEN I$=I$+N$(U-200)
1240 IF U>100 AND U<200 THEN I$=I$+"("+STR$(K(U-100))+")" ELSE IF U<100 THEN I$=I$+RIGHT$(STR$(U),LEN(STR$(U))-1)
1250 NEXT:RETURN
1260 LPRINT "LOGIC NAME IS ";NAM$:LPRINT "TOTAL COLUMN IS";TC:PRINT "LAST COLUMN IS";LC:FOR Z=1 TO 40:IF D$(Z)>"" THEN LPRINT STR$(Z);" IS ";D$(Z)
1270 IF E$(Z)>"" THEN GOSUB 1230:LPRINT STR$(Z);" = ";I$
1280 NEXT:GOTO 930
1290 FOR Z=1 TO A:J(Z,VAL(F$(4)))=J(Z,VAL(F$(2))):NEXT:GOTO 930
1300 LC=VAL(RIGHT$(L$,LEN(L$)-15)):IF (LC>TC AND TC>0) OR LC>B THEN PRINT "LAST COLUMN IS OUT OF RANGE":PRINT:IF TC>0 THEN LC=TC-1 ELSE LC=B
1310 GOTO 930
1320 PRINT:PRINT "ERROR ** CAN'T UNDERSTAND YOUR ENTRY":PRINT:GOTO 930
1330 RETURN 270
1340 CP=1:GOSUB 1540:IF R=1 THEN RW=RW-1:IF RW<1 THEN RW=1:GOSUB 360 ELSE GOSUB 360 ELSE GOSUB 440:R=R-1:GOSUB 430
1350 IF E$(FNR)>"" THEN 1340 ELSE RETURN 1510
1360 CP=1:GOSUB 1540:IF C=1 THEN CL=CL-1:IF CL<0 THEN CL=0:GOSUB 360 ELSE GOSUB 360 ELSE GOSUB 440:C=C-1:GOSUB 430
1370 RETURN 1510
1380 IF FNC=LC AND TC=0 THEN 1390 ELSE IF FNC=LC THEN GOSUB 440:CL=TC-2:C=2:GOSUB 360:GOSUB 430:GOTO 1390 ELSE CP=1:GOSUB 1540:IF C=3 THEN CL=CL+1:IF CL>B-2 THEN CL=B-2:GOSUB 360 ELSE GOSUB 360 ELSE GOSUB 440:C=C+1:GOSUB 430
1390 RETURN 1510
1400 CP=1:GOSUB 1540:IF R=10 THEN RW=RW+1:IF RW>A-9 THEN RW=A-9:GOSUB 360 ELSE GOSUB 360 ELSE GOSUB 440:R=R+1:GOSUB 430
1410 IF E$(FNR)>"" AND FNR<A THEN 1400 ELSE IF E$(FNR)>"" AND FNR=A THEN 1340 ELSE RETURN 1510
1420 CP=1:GOSUB 1540:RW=RW-5:IF RW<1 THEN RW=1
1430 GOSUB 360:IF E$(FNR)>"" THEN 1340 ELSE 1510
1440 CP=1:GOSUB 1540:CL=CL-3:IF CL<0 THEN CL=0
1450 GOSUB 360:GOTO 1510
1460 IF FNC+3>LC AND TC=0 THEN RETURN ELSE GOSUB 1540:IF FNC+3>LC THEN GOSUB 440:CL=TC-2:C=2:GOSUB 430 ELSE CP=1:CL=CL+3:IF CL>B-2 THEN CL=B-2
1470 GOSUB 360:GOTO 1510
1480 CP=1:GOSUB 1540:RW=RW+5:IF RW>A-9 THEN RW=A-9
1490 GOSUB 360:IF E$(FNR)>"" THEN 1400 ELSE 1510
1500 A$=INKEY$:IF A$="" THEN 1500 ELSE RETURN
1510 GOSUB 1530:DEF SEG=0:POKE 1050,PEEK(1052)
```

Continued

SNAP-CALC

Continued

IBM PCjr

```
1520 K$=INKEY$:IF K$="" THEN 1520 ELSE I
     F K$=CHR$(13) THEN GOSUB 1400:GOTO
     1520 ELSE IF LEN(K$)>1 THEN GOSUB 1
     550:GOTO 1560 ELSE GOSUB 1550:RETUR
     N
1530 FOR Z=1 TO 15:KEY(Z) ON:NEXT:RETURN
1540 FOR Z=1 TO 15:KEY(Z) OFF:NEXT:RETUR
     N
1550 FOR Z=1 TO 15:KEY(Z) STOP:NEXT:RETU
     RN
1560 IF RIGHT$(K$,1)=CHR$(71) THEN 1420
1570 IF RIGHT$(K$,1)=CHR$(73) THEN 1440
1580 IF RIGHT$(K$,1)=CHR$(81) THEN 1460
1590 IF RIGHT$(K$,1)=CHR$(79) THEN 1480
     ELSE RETURN

1600 LOCATE 25,1:PRINT "*** ERROR":ERR;"
     IN LINE";ERL;" ***";CHR$(7)::FOR E
     R=1 TO 1000:NEXT:LOCATE 25,1:PRINT
     "                              "
     ;:RESUME NEXT
1610 DATA LOGIC NAME IS,TOTAL COLUMN IS,
     NEW,LIST,PRINT,END,LAST COLUMN IS
     DATA + - * / LAG
1620 GOSUB 1550:CLS:LOCATE 1,1:PRINT CHR
1630 $(7);CHR$(7);" ARE YOU SURE YOU WANT
     TO EXIT THE PROGRAM (Y/N)?";
     :INPUT EX$:IF EX$="Y" THEN END ELSE
     RETURN 270
```

HCM

SNAP-CALC

TI-99/4A

```
100 REM **********
110 REM * SNAP-CALC *
120 REM **********
130 REM BY GARY STRAUSS
140 REM HOME COMPUTER MAGAZINE
150 REM VERSION 4.3.1
160 REM TI EXTENDED BASIC
170 DEF RN=RW+R-1 :: DEF CN=CW+Q-1
180 CR$="FF0000000000000FF00" :: CALL CH
    AR(136,CR$,138,CR$)::CALL MAGNIFY(
    4)::CALL DISPLAY AT(9,10)ERASE ALL:"SNA
    P-CALC" :: ON ERROR 1550
190 OPTION BASE 1 :: A=20 :: F,B=13 :: D
    LC=F-1 :: R,Q,RW,G,H,I=1 :: DIM D$(
    20),E$(20),F$(40),G$(21),J(20,13),K
    (50)
200 DEF L(M)=ASC(SEG$(E$(N),M,G)) :: FOR
    N=1 TO 500 :: NEXT N :: ON WARNING
    NEXT :: DV$="DSK1." :: FN$="DATA" FI
    LE" :: PDV$="RS232.BA=9600.DA=8"
210 GOSUB 1150 :: CALL SCREEN(6)::H$=K$ :: GO
    SUB 1150
220 GOSUB 1210 :: IF ((P>47 AND P<58)OR
    P=46)AND CN>0 THEN GOSUB 1130 :: G
    OTO 220
230 IF P>31 AND P<123 AND CN<1 THEN GOS
    UB 1100 :: GOTO 220
240 IF P=133 AND RW<6 THEN RW=1 :: GOSU
    B 1260 :: AC=0 :: GOTO 220 ELSE IF
    P=133 THEN RW=RW-5 :: GOSUB 1260 ::
    AC=0 :: GOTO 220
250 IF P=152 AND RW+5>A-9 THEN RW=A-9 ::
    GOSUB 1260 :: AC=0 :: GOTO 220 EL
    SE IF P=152 THEN RW=RW+5 :: GOSUB 1
    260 :: AC=0 :: GOTO 220
260 IF P=147 AND CW<3 THEN CW=0 :: GOSU
    B 1260 :: AC=0 :: GOTO 220 ELSE IF
    P=147 THEN CW=CW-3 :: GOSUB 1260 ::
    AC=0 :: GOTO 220
270 IF P=132 AND CW+3>LC-2 THEN CW=LC-1
    :: GOSUB 1260 :: AC=0 :: GOTO 220
    ELSE IF P=132 THEN CW=CW+3 :: GOSUB
    1260 :: AC=0 :: GOTO 220
280 IF P>15 OR P=0 THEN 220 ELSE ON P G
    OSUB 500,220,920,920,220,220,310,13
    80,1350,1310,1330,1400,1310,330,290
    :: IF O=1 THEN 220 ELSE 210
290 CALL CLEAR :: CALL DELSPRITE(ALL)::
    DISPLAY AT(12,1):"HALT (Y/N)" ::
    ACCEPT AT(12,11)VALIDATE("YN")SIZE(
    -1):H$ :: IF H$="Y" THEN STOP
300 O=0 :: RETURN
310 AC=0 :: IF CN>0 THEN J(RN,CN)=0 ::
    DISPLAY AT(R*2+3,(Q-1)*8+6):USING 1
    500:J(RN,CN);ELSE D$(RN)="" :: DISP
    LAY AT(R*2+3,4)SIZE(9):""
320 RETURN
330 H$="CALCULATE MODE" :: GOSUB 1170
340 FOR T=G TO LC :: X$="" :: FOR N=G T
    O S :: DISPLAY AT(3,G):"
                            ROW";
    N;" COLUMN";T :: X=0
350 IF E$(N)="" THEN 440
360 J(N,T)=0 :: FOR M=G TO LEN(E$(N)) ::
    U=L(M) :: IF M<LEN(E$(N))THEN V=L(M
    +G)ELSE V=G
370 GOSUB 480 :: W=V :: IF W=205 THEN X
    =G :: M=M+G :: IF T>G THEN V=L(M+G)
    :: W=J(V,T-G)ELSE W=0
380 IF U=201 THEN J(N,T)=J(N,T)+W :: M=
    M+G :: GOTO 430
390 IF U=202 THEN J(N,T)=J(N,T)-W :: M=
    M+G :: GOTO 430
400 IF U=203 THEN J(N,T)=J(N,T)*W :: M=
    M+G :: GOTO 430
410 IF U=204 AND J(N,T)<>0 AND W<>0 THE
    N J(N,T)=J(N,T)/W :: M=M+G :: GOTO
    430 :: ELSE IF U=204 THEN J(N,T)=0
    :: GOTO 430
420 V=U :: GOSUB 480 :: J(N,T)=V
430 NEXT M
440 X$=X$&CHR$(X)::NEXT N :: NEXT T ::
    DISPLAY AT(3,G):" TOTAL COL
    UMN"
450 FOR N=G TO S :: J(N,F)=0 :: IF AV=G
    THEN 470 ELSE IF ASC(SEG$(X$,N,G))
    =G THEN J(N,F)=J(N,F-G)::GOTO 470

460 FOR T=G TO F-G :: J(N,F)=J(N,F)+J(N
    ,T)::NEXT T
470 NEXT N :: CALL SCREEN(6)::H$=C$ ::
    GOSUB 1170 :: GOSUB 1260 :: RETURN
480 IF V>100 AND V<201 THEN V=K(V-100)E
    LSE IF V<101 THEN V=J(V,T)
490 RETURN
500 GOSUB 710 :: O=2 :: H$="LOGIC MODE"
    :: CALL SCREEN(4)::GOSUB 1150
510 J$=">" :: GOSUB 1180
520 IF F$(G)="TOTAL" THEN IF F$(4)="OFF
    " THEN AV=G :: F=0 :: GOTO 510 ELSE
    F=VAL(F$(4))::LC=F-1 :: AV=0 :: G
    OTO 510
530 IF F$(G)="LAST" THEN IF VAL(F$(4))<
    F OR AV=1 THEN LC=VAL(F$(4))::GOTO
    510 ELSE IF AV=0 THEN LC=F-1 :: GO
    TO 510
540 IF F$(G)="END" THEN RETURN ELSE IF
    F$(G)<>"MOVE" THEN 560
550 FOR N=G TO A :: J(N,VAL(F$(4)))=J(N
    ,VAL(F$(2)))::NEXT N :: GOTO 510
560 IF F$(G)="LOGIC" THEN K$=F$(4)::GO
    TO 510 ELSE IF F$(3)="NULL" THEN D$
    (VAL(F$(G)))="" :: E$(VAL(F$(G)))=
    "" :: GOTO 510
570 IF F$(G)="LIST" THEN GOSUB 760 :: G
    OTO 510 ELSE IF F$(G)="PRINT" THEN
    AW=G :: GOSUB 1600 :: GOSUB 760 ::
    GOSUB 1620 :: AW=0 :: GOTO 510
580 IF F$(G)="NEW" THEN GOSUB 1530 :: G
    OTO 510 ELSE IF F$(2)="IS" THEN Y=V
    AL(F$(G)):: D$(Y)=SEG$(F$(3),1,9)::
    S=MAX(S,Y):: GOTO 510
590 IF F$(2)<>"=" THEN PRINT "COMMAND N
    OT RECOGNIZED" :: GOTO 510 ELSE AA=
    VAL(F$(G)):: E$(AA)="" :: S=MAX(S,A
    A)
600 FOR N=3 TO 40 :: IF F$(N)="" THEN 5
    10
610 IF F$(N)="LAG" THEN U=205 :: GOTO 6
    80
620 IF F$(N)="+" THEN U=201 :: GOTO 680
630 IF F$(N)="-" THEN U=202 :: GOTO 680
640 IF F$(N)="*" THEN U=203 :: GOTO 680
650 IF F$(N)="/" THEN U=204 :: GOTO 680
660 IF SEG$(F$(N),G,G)="(" THEN GOSUB 7
    00 :: AB=AB+G :: K(AB)=Y :: E$(AA)=
    E$(AA)&CHR$(100+AB):: GOTO 690
670 U=VAL(F$(N))
680 E$(AA)=E$(AA)&CHR$(U)
690 NEXT N :: GOTO 510
700 Y=VAL(SEG$(F$(N),2,LEN(F$(N))-2)):
    RETURN
710 CALL DELSPRITE(#G):: RETURN
720 FOR N=1 TO 40 :: F$(N)="" :: NEXT N
    :: AD=G :: FOR N=1 TO 40 :: AE=POS
    (L$,",",AD):: IF AE=0 THEN AE=LEN(L
    $)+G
730 F$(N)=SEG$(L$,AD,AE-AD):: AD=AE+G :
    IF AE>LEN(L$)THEN RETURN
740 NEXT N
750 I$="TO MANY WORDS" :: GOSUB 1200 ::
    RETURN
760 I$="LOGIC NAME IS "&K$ :: GOSUB 120
    0 :: I$="TOTAL COLUMN IS "&STR$(F):
    : GOSUB 1200 :: I$="LAST COLUMN IS
    "&STR$(LC):: GOSUB 1200
770 FOR N=G TO A :: CALL KEY(0,P,AH)::
    IF AH<>0 THEN CALL SOUND(100,440,0)
    :: GOSUB 1230
780 IF LEN(D$(N))>0 THEN I$=STR$(N)&" I
    S "&D$(N):: GOSUB 1200
790 IF E$(N)="" THEN 910 ELSE I$=""
800 I$=STR$(N)&" = "
810 FOR M=G TO LEN(E$(N)):: U=L(M)
820 IF U=201 THEN I$=I$&" + " :: GOTO 8
    90
830 IF U=202 THEN I$=I$&" - " :: GOTO 8
    90
840 IF U=203 THEN I$=I$&" * " :: GOTO 8
    90
850 IF U=204 THEN I$=I$&" / " :: GOTO 8
    90
860 IF U=205 THEN I$=I$&"LAG " :: GOTO
    890
```

Continued

PROGRAM LISTING

SNAP-CALC

TI-99/4A

Continued

```
870 IF U>100 THEN I$=I$&"(("&STR$(K(U-10
    0))&"))" :: GOTO 890
880 I$=I$&STR$(U)
890 NEXT M
900 GOSUB 1200
910 NEXT N :: RETURN
920 IF P=4 THEN TTL$="SAVE MODE" ELSE T
    TL$="LOAD MODE"
930 CALL DELSPRITE(ALL)::DISPLAY AT(1,
    1)ERASE ALL:TTL$:"ENTER YOUR CHOICE
    :":"1) DATA":"2) LOGIC"
940 ACCEPT AT(8,1)VALIDATE("12")SIZE(1)
    :LM :: IF LM=1 THEN 1030
950 DISPLAY AT(10,1):"ENTER DEVICE NAME
    :":DV$ :: ACCEPT AT(11,1)SIZE(-28):
    DV$
960 IF K$="" AND DV$<>"CS1" THEN DISPLA
    Y AT(13,1):"ENTER FILE NAME:":K$ ::
     ACCEPT AT(14,1):K$
970 K$=SEG$(K$,1,10)::IF DV$<>"CS1" TH
    EN DV$=SEG$(DV$,1,4)&"."&K$ ELSE D
    V$=DV$
980 IF P=3 THEN OPEN #G:DV$,INPUT ,INT
    ERNAL,FIXED 192 ELSE OPEN #G:DV$,O
    UTPUT,INTERNAL,FIXED 192
990 IF P=4 THEN 1010
1000 INPUT #G:K$,S,F,LC,A,B ::  FOR N=1 T
     O A :: INPUT #G:D$(N),E$(N):: NEXT
     N :: O=0 :: CLOSE #G :: RETURN
1010 PRINT #G:K$,S,F,LC,A,B :: FOR N=1 T
     O A :: PRINT #G:D$(N),E$(N):: NEXT
     N :: O=0 :: CLOSE #G :: RETURN
1020 IF P=3 THEN 1030
1030 DISPLAY AT(9,1):"ENTER DEVICE NAME:
     ":DV$ :: ACCEPT AT(10,1)SIZE(-28):D
     V$ :: IF DV$="CS1" THEN DV$$=DV$ ::
      GOTO 1050
1040 DISPLAY AT(12,1):"FILE NAME:":FN$ :
     : ACCEPT AT(13,1)SIZE(-28):FN$ :: F
     N$=SEG$(FN$&"        ",1,9)&"*" ::
      DV$$=SEG$(DV$,1,4)&"."&FN$
1050 IF P=3 THEN 1080
1060 OPEN #G:DV$$,OUTPUT,INTERNAL,FIXED
     128 :: FOR N=1 TO A :: PRINT #G:J(N
     ,1));J(N,2));J(N,3));J(N,4));J(N,5));J(N
     ,6));J(N,7)
1070 PRINT #G:J(N,8));J(N,9));J(N,10));J(N,
     11));J(N,12));J(N,13):: NEXT N :: CLO
     SE #G :: O=0 :: RETURN
1080 OPEN #G:DV$$,INPUT,INTERNAL,FIXED
     128 :: FOR N=1 TO A :: INPUT #G:J(N
     ,1),J(N,2),J(N,3),J(N,4),J(N,5),J(N
     ,6),J(N,7)
1090 INPUT #G:J(N,8),J(N,9),J(N,10),J(N,
     11),J(N,12),J(N,13):: NEXT N :: CLO
     SE #G :: O=0 :: RETURN
1100 IF AC=0 THEN AC=1 :: M$=""
1110 IF LEN(M$)<9 THEN M$=M$&CHR$(P)ELSE
     CALL SOUND(50,220,0)
1120 DISPLAY AT(R*2+3,4)SIZE(9):M$ :: D$
     (RN)=M$ :: RETURN
1130 IF AC<>G THEN AC=G :: M$=""0"" ELSE I
     F LEN(M$)=8 OR VAL(M$&CHR$(P))>9999
     .99 THEN CALL SOUND(100,220,0):: RE
     TURN
1140 M$=M$&CHR$(P)::J(RN,CN)=VAL(M$)::
     DISPLAY AT(R*2+3,(Q-1)*8+6)SIZE(7):
     USING 1500:VAL(M$)::RETURN
1150 DISPLAY AT(G,10)ERASE ALL:"SNAP-CAL
     C":H$
1160 IF O=G THEN GOSUB 1260 :: GOSUB 122
     0 :: RETURN ELSE RETURN
1170 DISPLAY AT(2,G):H$ :: RETURN
1180 Z=G
1190 INPUT ">":L$ :: GOSUB 720 :: IF AW=
     G THEN PRINT #5:L$ :: RETURN ELSE R
     ETURN
1200 IF AW=G THEN PRINT #5:I$ :: RETURN
     ELSE PRINT I$ :: RETURN
1210 CALL KEY(0,P,AH)::IF AH=0 THEN 121
     0 ELSE CALL SOUND(10,1000,8):: RETU
     RN
1220 CALL SPRITE(#G,136,2,R*16+15,(Q-1)*
     64+68):: RETURN
1230 CALL KEY(0,P,AH)::IF AH=-G THEN 12
     30
1240 CALL KEY(0,P,AH)::IF AH=0 THEN 124
     0
1250 RETURN
1260 IF CW>0 THEN 1290
1270 DISPLAY AT(3,3):"ROW NAME 1
     2";:: N=R
1280 FOR R=1 TO 10 :: DISPLAY AT(R*2+3,1
     ):USING 1580:RN,D$(RN),J(RN,1),J(RN
     ,2);:: NEXT R :: R=N :: RETURN
1290 DISPLAY AT(3,7):CW;TAB(16);CW+1;TAB
     (24);CW+2;:: N=R
1300 FOR R=1 TO 10 :: DISPLAY AT(R*2+3,1
     ):USING 1590:RN,J(RN,CW),J(RN,CW+1)
     ,J(RN,CW+2);:: NEXT R :: R=N :: RET
     URN
1310 IF RN=A THEN RETURN ELSE R=R+1 :: A
     C=0 :: IF R>10 THEN R=10 :: RW=RW+1
     :: GOSUB 1260
1320 GOSUB 1220 :: RETURN
1330 IF RN=1 THEN RETURN ELSE R=R-1 :: A
     C=0 :: IF R<1 THEN R=1 :: RW=RW-1 :
     : GOSUB 1260
1340 GOSUB 1220 :: RETURN
1350 IF CN=LC AND Q=2 THEN RETURN
1360 IF CN=LC THEN CW=F-2 :: Q=2 :: AC=0
     :: GOSUB 1260 ELSE Q=Q+1 :: AC=0 :
     : IF Q>3 THEN Q=3 :: CW=CW+1 :: GOS
     UB 1260
1370 GOSUB 1220 :: RETURN
1380 IF CN=0 THEN RETURN ELSE Q=Q-1 :: A
     C=0 :: IF Q<1 THEN Q=1 :: CW=CW-1 :
     : GOSUB 1260
1390 GOSUB 1220 :: RETURN
1400 CALL DELSPRITE(ALL)
1410 O=2 :: CALL CLEAR :: INPUT "REPORT
     TITLE:":RT$ :: INPUT "DATE":DT$ ::
     INPUT "NUMBER OF ROWS TO REPORT ON
     ":NR :: IF NR=0 THEN RETURN
1420 CALL CLEAR :: DISPLAY AT(20,1):"ENT
     ER DEVICE NAME FOR REPORT":PDV$ ::
     ACCEPT AT(21,1)SIZE(-28):PDV$ :: OP
     EN #5:PDV$,VARIABLE 132
1430 PG=1 :: PRINT #5:CHR$(15);
1440 PRINT #5:RT$;TAB(40);DT$:RPT$("-",1
     30)::" PAGE ";PG:RPT$("-",130)::"ROW
     ";:: FOR TP=0 TO 13 :: PRINT #5:TAB
     (TP*9+10);(PG-1)*13+TP;
1450 NEXT TP :: PRINT #5:: TAB(5);"|";RP
     T$("-","|",125):TAB(5);"|" :: FOR N=G T
     O MIN(NR,A)
1460 PRINT #5:N;TAB(5);"|";:: FOR M=0 TO
     13 :: IF PG=1 AND M=0 THEN PRINT #
     5,USING 1520:D$(N);ELSE PRINT #5,US
     ING 1510:J(N,(PG-1)*14+M);
1470 IF (PG-1)*14+M>=F THEN 1490
1480 NEXT M
1490 PRINT #5:"|" :: NEXT N :: PG=PG+1 ::
      IF (PG-1)*14<F THEN PRINT #5:CHR$(
     12):: GOTO 1440 ELSE GOSUB 1620 ::
     RETURN
1500 IMAGE "####.##"
1510 IMAGE "####.##"
1520 IMAGE "##########"
1530 CALL DELSPRITE(#G):: FOR N=G TO A :
     : D$(N),E$(N)="" :: NEXT N :: FOR N
     =G TO A :: FOR M=G TO B :: J(N,M)=0
     :: NEXT M :: NEXT N
1540 AV,S=0 :: K$="" :: F=13 :: RETURN
1550 CALL ERR(AR,AS,AX,AU):: I$="ERROR C
     ="&STR$(AR)&" TYP="&STR$(AS)&" SEV
     ="&STR$(AX)&" AT "&STR$(AU):: IF AR
     =109 OR AR=130 THEN AW=0
1560 CALL SOUND(200,110,0)
1570 DISPLAY AT(23,1):I$;:: FOR TD=1 TO
     750 :: NEXT TD :: DISPLAY AT(23,1):
     "" ;:: ON ERROR 1550 :: RETURN NE
     XT
1580 IMAGE "### ########## ####.## ####.##
     "
1590 IMAGE "###      ####.## ####.## ####.##
     "
1600 DISPLAY AT(23,1):"ENTER PRINTER DEV
     ICE NAME":PDV$;:: ACCEPT AT(24,1)S
     IZE(-28):PDV$
1610 OPEN #5:PDV$,DISPLAY,OUTPUT,VARIAB
     LE 132 :: PRINT #5:CHR$(15):: RETUR
     N
1620 CLOSE #5 :: RETURN
```

HCM

SPIDER GRAPHICS

APPLE II Family

```
100 REM  **********************
110 REM  *  SPIDER GRAPHICS   *
120 REM  **********************
130 REM  BY WILLIAM K. BALTHROP
140 REM  HOME COMPUTER MAGAZINE
150 REM  VERSION 4.3.1
160 REM  APPLE II SERIES APPLESOFT
170 REM
180 TEXT
190 HOME : VTAB 12: HTAB 12: PRINT "SPI
    DER GRAPHICS": FOR X = 1 TO 1000: N
    EXT X: HOME
200 PRINT "ENTER YOUR CHOICE:": VTAB 4:
    PRINT "1. ANALOG- JOYSTICK, PADDLE
    S": VTAB 6: PRINT "2. KEYBOARD CURS
    OR KEYS": VTAB 12: PRINT "SELECT OP
    TION";:: GET KB$
```

Continued

SPIDER GRAPHICS

Continued

APPLE II Family

```
210 IF KB$ < "1" OR KB$ > "2" THEN HOM
    E : GOTO 200
220 KB = VAL (KB$) - 1
230 HOME : PRINT "PRESS THE NUMBER NEXT
    TO THE MODE TO SELECT A MODE OR
    FUNCTION.": PRINT : PRINT
    "PRESS RETURN, OR THE FIRE BUTTON T
    O KEEP A LINE, PLOT A POINT, OR DRAW
     A RAY."
240 VTAB 23: PRINT "PRESS RETURN TO CON
    TINUE";
250 GET KB$: IF KB$ = "" THEN 250
260 FOR X = 7676 TO 7693: READ A: POKE
    X,A: NEXT X: POKE 232,252: POKE 233
    ,29
270 PX = 140:PY = 100:MODE = 1:PDX = 14
    0:PDY = 100
280 HGR : HCOLOR= 3: SCALE= 1: IF KB =
    1 THEN XDRAW 1 AT PX,PY
290 COL = 3
300 HOME : VTAB 21: PRINT "1. LINE OFF"
    ;: HTAB 20: PRINT "5. PLOT POINT":
    PRINT "2. LINE ON";: HTAB 20: PRINT
    "6. RETURN TO ORIGIN"
310 PRINT "3. ERASE RAYS";: HTAB 20: PR
    INT "7. CHANGE COLOR": PRINT "4. DR
    AW RAYS";: HTAB 20: PRINT "8. CLEAR
     SCREEN";
320 ON MODE GOTO 550,560,570,580,590
330 IF KB = 1 THEN GOTO 370
340 GOSUB 760: GOSUB 870: IF A$ = "" TH
    EN GOTO 340
350 IF A$ < "1" OR A$ > "8" THEN GOTO
    340
360 GOTO 540
370 GOSUB 870: IF A$ = "" THEN GOTO 37
    0
380 IF A$ > "0" AND A$ < "9" THEN GOTO
     540
390 A = ASC (A$)
400 IF A = 8 AND PX > 0 THEN MX = -1:
    MY = 0: GOTO 460
410 IF (A = 21 OR A = 32) AND PX < 279
    THEN MX = 1:MY = 0: GOTO 460
420 IF A = 10 AND PY < 159 THEN MY = 1:
    MX = 0: GOTO 460
430 IF A = 11 AND PY > 0 THEN MY = -1
    :MX = 0: GOTO 460
440 IF A = 13 THEN ON MODE GOTO 330,51
    0,520,530
450 GOTO 330
460 IF MODE = 3 THEN GOTO 490
470 IF MODE < > 2 THEN GOTO 500
480 PY = PY + MY:PX = PX + MX: HCOLOR=
    0: HPLOT OX,OY TO PX - MX,PY - MY:
    HCOLOR= COL: HPLOT OX,OY TO PX,PY:
    GOTO 330
490 PY = PY + MY:PX = PX + MX: HCOLOR=
    0: HPLOT OX,OY TO PX,PY: DRAW 1 AT
    PX - MX,PY - MY: HCOLOR= 3: DRAW 1
    AT PX,PY: GOTO 330
500 XDRAW 1 AT PX,PY: PY = PY + MY: PX =
    PX + MX: XDRAW 1 AT PX,PY: GOTO 330
510 OX = PX: OY = PY: PRINT CHR$ (7);:
    GOTO 330
520 HPLOT OX,OY TO PX,PY: PX = OX: PY =
    Y: PRINT CHR$ (7);: GOTO 330
530 XDRAW 1 AT PX,PY: GOTO 330
540 M = MODE: ON VAL (A$) GOTO 550,560
    ,570,580,590,600,610,670
550 MODE = 1: VTAB 21: HTAB 4: INVERSE
    : PRINT "LINE OFF";: GOTO 680
560 MODE = 2: OX = PX: OY = PY: VTAB 22:
    HTAB 4: INVERSE : PRINT "LINE ON";:
    GOTO 680
570 MODE = 3: TC = COL: COL = 0: VTAB 23:
    HTAB 4: INVERSE : PRINT "ERASE RAY
    S";: HCOLOR= 0: HPLOT OX,OY TO PX,P
    Y: GOTO 680
580 MODE = 4: OX = PX: OY = PY: VTAB 24:
    HTAB 4: INVERSE : PRINT "DRAW RAYS"
    ;: GOTO 680
590 MODE = 5: VTAB 21: HTAB 23: INVERSE
    : PRINT "PLOT POINT";: GOTO 680
600 PX = OX: PY = OY: GOTO 330
610 HOME : VTAB 21: PRINT "0. BLACK
    1. GREEN 2. VIOLET"
620 PRINT "3. WHITE 4. BLACK 5.
    ORANGE"
630 PRINT "6. BLUE 7. WHITE"
640 PRINT "YOUR CHOICE?";
650 GET A$: IF A$ < "0" OR A$ > "7" THE
    N GOTO 650
660 COL = VAL (A$): PRINT A$; CHR$ (7)
    ;: GOTO 300
670 HGR : GOTO 330
680 NORMAL : IF M < > MODE THEN ON M
    GOTO 700,710,720,730,740
690 GOTO 330
700 VTAB 21: HTAB 4: PRINT "LINE OFF ";
    : GOTO 330
710 VTAB 22: HTAB 4: PRINT "LINE ON ";:
    GOTO 330
720 COL = TC: VTAB 23: HTAB 4: PRINT "E
    RASE RAYS";: GOTO 330
730 VTAB 24: HTAB 4: PRINT "DRAW RAYS "
    ;: GOTO 330
740 VTAB 21: HTAB 23: PRINT "PLOT POINT
    ";: GOTO 330
750 DATA 2,0,6,0,8,0,4,0,18,63,32,100,
    45,21,54,30,7,0
760 PX = PDL (0)
770 FOR X = 1 TO 10:PY = PDL (1) * .6
780 XDRAW 2 AT PX,PY: XDRAW 2 AT PX,PY
790 FB1 = PEEK (-16287):FB2 = PEEK
    (-16286): IF FB1 > 127 OR FB2 > 1
    27 THEN ON MODE GOSUB 820,830,840,
    850,860
800 OPX = PX:OPY = PY
810 RETURN
820 RETURN
830 HCOLOR= COL: HPLOT OX,OY TO PX,PY:O
    X = PX:OY = PY: RETURN
840 HCOLOR= 0: HPLOT OX,OY TO PX,PY:OX
    = PX:OY = PY: RETURN
850 HCOLOR= COL: HPLOT OX,OY TO PX,PY:
    RETURN
860 HCOLOR= COL: HPLOT PX,PY: RETURN
870 REM KEY ROUTINE
880 KEY = PEEK (-16384): IF KEY > 12
    7 THEN KEY = KEY - 128: A$ = CHR$ (
    KEY): POKE -16368,0: RETURN
890 A$ = "": RETURN
```

HCM

SPIDER GRAPHICS

IBM PC & PCjr

```
100 REM ***********************
110 REM * SPIDER GRAPHICS *
120 REM ***********************
130 REM BY WILLIAM K. BALTHROP
140 REM HOME COMPUTER MAGAZINE
150 REM VERSION 4.3.1
160 REM IBM PC OR IBM PCjr
170 REM BASICA OR CARTRIDGE BASIC
180 REM WITH 64K MEMORY EXPANSION
190 CLS:LOCATE 12,12:PRINT "SPIDER GRAP
    HICS":FOR X=1 TO 1000:NEXT X:CLS
200 REM
210 REM INPUT OPTION AND INITIALIZATION
220 REM
230 LOCATE 3,3:PRINT "ENTER YOUR CHOICE
    :":PRINT:PRINT "1. IBM PC":PRINT:PR
    INT "2. IBM PCjr"
240 A$=INKEY$:IF A$="" THEN 240 ELSE IF
     A$="1" THEN MACH=1 ELSE IF A$="2"
    THEN MACH=2 ELSE GOTO 240
250 IF MACH=1 THEN SCREEN 1 ELSE CLEAR
    ,,32768!:SCREEN 5:MACH=2
260 CLS:LOCATE 3,3:PRINT "ENTER YOUR CH
    OICE:":LOCATE 6,5:PRINT "1. ANALOG-
    JOYSTICK OR PADDLES":LOCATE 8,5:PR
    INT "2. KEYBOARD-'ESDX' KEYS":LOCA
    TE 12,5
270 A$=INKEY$:IF A$<"1" OR A$>"2" THEN
    270
280 IP=VAL(A$):KEY OFF:PX=160:PY=84:MOD
    E=2:OX=140:OY=84:CLS:CL=15:IF MACH=
    2 THEN COL=10 ELSE COL=3
290 A$="1":LOCATE 5,1:PRINT "PRESS THE
    NUMBER NEXT TO A MODE":PRINT "TO SE
    LECT A NEW MODE OR FUNCTION.":PRINT
    :PRINT "PRESS ENTER, OR THE FIRE BU
    TTON TO KEEP A LINE, PLOT A POINT,
    OR DRAW A RAY.":LOCATE 24,1:PRINT "
    PRESS ENTER TO CONTINUE";
300 A$=INKEY$:IF A$="" THEN 300
310 DIM C(10)
320 PRESET (1,1):DRAW "C7 BLEFGH":GET (
    0,0)-(2,2),C:CLS:A$="2"
330 REM
340 REM DISPLAY MODES AND FUNCTIONS
350 REM
360 GOSUB 1120:LOCATE 22,1:PRINT "1. li
    ne off";:LOCATE 22,20:PRINT "5. plo
    t point";:LOCATE 23,1:PRINT "2. lin
    e on";:LOCATE 23,20:PRINT "6. retur
    n to origin";:LOCATE 24,1:PRINT "3. erase rays";:
    LOCATE 24,20:PRINT "7. change color
    ";:LOCATE 25,1:PRINT "4. draw rays"
    ;:LOCATE 25,20:PRINT "8. clear scre
    en";
370 GOTO 630
380 REM
390 REM KEYBOARD INPUT ROUTINE
400 REM
410 REM
420 IF IP=1 THEN GOTO 980
430 GOSUB 1110:PUT (PX,PY),C,XOR:PUT (P
    X,PY),C,XOR:IF A$="" THEN GOTO 430
```

Continued

PROGRAM LISTING

SPIDER GRAPHICS

IBM PC & PCjr

Continued

```
440 IF A$>"0" AND A$<"9" THEN ON VAL(A$
    ) GOTO 630,630,630,630,630,720,730,
    850
450 A=ASC(A$):IF A=83 AND PX>0 THEN MX=
    -1:MY=0:GOTO 510
460 IF A=68 AND PX<317 THEN MX=1:MY=0:G
    OTO 510
470 IF A=69 AND PY>0 THEN MX=0:MY=-1:GO
    TO 510
480 IF A=88 AND PY<167 THEN MX=0:MY=1:G
    OTO 510
490 IF A=13 THEN ON MODE GOTO 420,570,5
    70,580,590
500 GOTO 430
510 IF MODE=2 THEN LINE (OX,OY)-(PX,PY)
    ,0:PX=PX+MX:PY=PY+MY:LINE (OX,OY)-(
    PX,PY),COL:GOTO 420
520 IF MODE=3 THEN PX=PX+MX:PY=PY+MY:LI
    NE (OX,OY)-(PX,PY),0:GOTO 420
530 PX=PX+MX:PY=PY+MY:GOTO 420
540 REM
550 REM ENTER ROUTINES FOR KEYBOARD MOD
    E
560 REM
570 OX=PX:OY=PY:PRINT CHR$(7);:GOTO 420
580 LINE (OX,OY)-(PX,PY),COL:PX=OX:PY=O
    Y:PRINT CHR$(7);:GOTO 420
590 PSET (PX+1,PY+1),COL:GOTO 420
600 REM
610 REM CHANGE MODE ROUTINES
620 REM
630 IF MACH=2 THEN M=MODE:COLOR 15,0:ON
    VAL(A$) GOTO 640,650,660,670,680 E
    LSE M=MODE:PSET (319,199),2:ON VAL(
    A$) GOTO 640,650,660,670,680
640 MODE=1:LOCATE 22,4:PRINT "LINE OFF"
    ;:GOTO 890
650 MODE=2:LOCATE 23,4:PRINT "LINE ON";
    :OX=PX:OY=PY:GOTO 890
660 MODE=3:LOCATE 24,4:PRINT "ERASE RAY
    S";:OX=PX:OY=PY:GOTO 890
670 MODE=4:LOCATE 25,4:PRINT "DRAW RAYS
    ";:OX=PX:OY=PY:GOTO 890
680 MODE=5:LOCATE 22,23:PRINT "PLOT POI
    NT";:GOTO 890
690 REM
700 REM FUNCTION ROUTINES
710 REM
720 PX=OX:PY=OY:GOTO 420
730 IF MACH=1 THEN GOTO 810 ELSE COLOR
    15,0:LOCATE 22,1:PRINT "0.BLACK 4.
    RED 8.GRAY C.LT RED"
740 LOCATE 23,1:PRINT "1.BLUE 5.MAG.
    9.LT BLUE D.LT MAG.
750 LOCATE 24,1:PRINT "2.GREEN 6.BROWN
    A.LT GRN. E.YELLOW";
760 LOCATE 25,1:PRINT "3.CYAN 7.WHITE
    B.LT CYAN F.HI WHITE";
770 B$=INKEY$:IF B$="" OR B$<"0" OR B$>
    "F" OR (B$>"9" AND B$<"A") THEN GOT
    O 770
780 IF B$>"9" THEN COL=ASC(B$)-55 ELSE
    COL=VAL(B$):IF COL<4 THEN PAL=0 ELS
    E PAL=1:COL=COL-3
790 LOCATE 22,1:PRINT "                    "
    ;:LOCATE 25,1:PRINT "                    "
    ;:
800 A$=STR$(MODE):GOTO 360
810 LOCATE 22,1:PRINT "PALETTE 0
    PALETTE 1                    1. GREEN
                        4. CYAN          2. RED
                                 5. MAGENTA
    LOCATE 25,1:PRINT "3. BROWN
                    6. WHITE                ";:
820 B$=INKEY$:IF B$="" THEN 820 ELSE IF
    B$<"1" OR B$>"6" THEN GOTO 820
830 COL=VAL(B$):IF COL<4 THEN PAL=0 ELS
    E PAL=1:COL=COL-3
840 A$=STR$(MODE):GOTO 360
850 CLS:GOTO 360
860 REM
870 REM TURN OFF OLD MODE
880 REM
890 IF MACH=2 THEN COLOR 2,0:IF M=MODE
    THEN GOTO 420 ELSE ON M GOTO 900,91
    0,920,930,940 ELSE PSET (319,199),1
    :IF M=MODE THEN GOTO 420 ELSE ON M
    GOTO 900,910,920,930,940
900 LOCATE 22,4:PRINT "line off";:GOTO
    420
910 LOCATE 23,4:PRINT "line on";:GOTO 4
    20
920 LOCATE 24,4:PRINT "erase rays";:GOT
    O 420
930 LOCATE 25,4:PRINT "draw rays";:GOTO
    420
940 LOCATE 22,23:PRINT "plot point";:GO
    TO 420
950 REM
960 REM ANALOG INPUT ROUTINE
970 REM
980 PX=STICK(0)*2.5:PY=STICK(1)*1.3:PUT
    (PX,PY),C,XOR:PUT (PX,PY),C,XOR
990 STRIG ON:FB=STRIG(1):STRIG OFF:IF F
    B=-1 THEN ON MODE GOTO 980,1040,105
    0,1060,1070
1000 GOSUB 1110:IF A$="" OR A$<"1" OR A$
    >"8" THEN GOTO 980 ELSE ON VAL(A$)
    GOTO 630,630,630,630,630,720,730,85
    0
1010 REM
1020 REM FIRE BUTTON ROUTINES FOR ANALOG
     MODE
1030 REM
1040 LINE (OX,OY)-(PX,PY),COL:OX=PX:OY=P
     Y:GOTO 980
1050 LINE (OX,OY)-(PX,PY),0:GOTO 980
1060 LINE (OX,OY)-(PX,PY),COL:PX=OX:PY=O
     Y:GOTO 980
1070 PSET (PX,PY),COL:GOTO 980
1080 REM
1090 REM KEY INPUT SUBROUTINE
1100 REM
1110 A$=INKEY$:RETURN
1120 IF MACH=2 THEN COLOR 2,0:RETURN ELS
     E COLOR 0,PAL:RETURN
```

HCM

WILD KINGDOM

APPLE II Family

```
100 REM  ***********
110 REM  *  WILD KINGDOM  *
120 REM  ***********
130 REM  BY MONTE ULM
140 REM  AND THE HCM STAFF
150 REM  HOME COMPUTER MAGAZINE
160 REM  VERSION 4.3.1
170 REM  APPLE II SERIES APPLESOFT
180 IF PEEK(104)<>64 THEN POKE 1
    04,64:POKE 103,1:POKE 16384,0:PR
    INT CHR$(4);"RUN WILD.KINGDOM"
190 TEXT:HOME:VTAB 12:HTAB 15:PRINT "WI
    LD KINGDOM":VTAB 21:HTAB 4:PRINT "
    (J)OYSTICK OR (K)EYBOARD ?"
    (J/K)";:GET A$:HOME:IF A$="J" THEN J=1
200 RESTORE:FOR K=768 TO 914:READ P:POKE K,P:NEXT K
210 C9=0:MEN=3:Z=1:C1=768:C2=772:SOUND=899:DIM D(20,38):POKE 232,111:POKE 233,3:SCALE=1:ROT=1
220 HGR:HCOLOR=1:FOR X=0 TO 6:HPLOT 0,X TO 279,X:HPLOT 0,X+153 TO 279,X+152:NEXT:FOR X=0 TO 153:HPLOT X,0 TO X,159:HPLOT X+266,0 TO X+266,159:NEXT
230 HCOLOR=3:HPLOT 12,7 TO 266,7 TO 266,152 TO 12,152 TO 12,7
240 FOR A=1 TO 18:FOR B=2 TO 37:C=INT(RND(1)*4)+96:IF B=2 AND C>97 THEN C=C-2
250 IF A=1 AND (C=97 OR C=99) THEN C=C-1
260 D(A,B)=C:VTAB A+1:HTAB B+1:POKE 0,C:CALL C1:NEXT:NEXT
270 MEN=MEN-1:IF MEN=1 THEN 1480
280 P=40:P=70:T=48:U=63
290 CP=0:CS=0:HOME:VTAB 23:HTAB 1:PRINT "MEN="MEN TAB(20)"SCORE="C9
300 T2=48:U2=70:D(6,10)=96:VTAB 7:HTAB 11:POKE 0,96:CALL C2
310 HCOLOR=5:DRAW 1 AT P+1,O+1:DRAW 1 AT U+1,T+1
320 HCOLOR=3:DRAW 1 AT U2+1,T2+1
330 D(1,5)=96:VTAB 2:HTAB 6:POKE 0,96:CALL C2:I=1:H=4:G=107:GOSUB 650
340 GOSUB 350:GOSUB 670:GOSUB 800:GOSUB 960:GOTO 340
350 VTAB H+1:POKE 0,D(I,H):CALL C2:IF J=0 THEN 420
360 P0=PDL(0):P1=PDL(1)
370 IF P0<40 THEN 490
380 IF P0<200 THEN 550
390 IF P1<40 THEN 520
400 IF P1>200 THEN 580
410 GOTO 660
420 K=PEEK(-16384)-128:IF K<0 THEN 660
430 POKE -16368,0
440 IF K=ASC("A") THEN 520
450 IF K=8 THEN 490
460 IF K=21 THEN 550
470 IF K=ASC("Z") THEN 580
480 GOTO 660
490 IF G<105 THEN G=105:GOTO 650
500 IF D(I,H)>97 THEN 650
```

Continued

APPLE II Family

```
510 H = H - 1: GOTO 610
520 IF G < > 104 THEN G = 104: GOTO 650
530 IF D(I,H) = 97 OR D(I,H) = 99 THEN 650
540 I = I - 1: GOTO 630
550 IF G < > 107 THEN G = 107: GOTO 650
560 IF D(I,H + 1) < 98 THEN H = H + 1: GOTO 610
570 GOTO 650
580 IF G < > 106 THEN G = 106: GOTO 650
590 IF D(I + 1,H) = 96 OR D(I + 1,H) = 98 THEN I = I + 1: GOTO 630
600 GOTO 650
610 IF H < 2 THEN H = 2: GOTO 660
620 IF H > 37 THEN H = 37: GOTO 660
630 IF I < 1 THEN I = 1: GOTO 660
640 IF I > 18 THEN I = 18: GOTO 660
650 VTAB I + 1: HTAB H + 1: POKE 0,G + 4: CALL C1: POKE 0,16: CALL SOUND:C9 = C9 + 1: VTAB 23: HTAB 28: PRINT C9: RETURN
660 VTAB I + 1: HTAB H + 1: POKE 0,G + 4: CALL C1: RETURN
670 HCOLOR= 0: DRAW 1 AT P + 1,O + 1:X = O / 8:Y = P / 7:Q = D(X,Y): GOSUB 1020:O = O + R:P = P + S
680 IF A = 1 AND CP = 0 THEN C9 = C9 + 500:CP = 1: POKE 0,105: CALL SOUND: CALL SOUND
690 IF A = 1 THEN 720
700 IF O1 = O AND P1 = P THEN O = O - R:P = P - S: GOSUB 1210:O = O + R:P = P + S
710 O1 = X * 8:P1 = Y * 7: IF O = I * 8 AND P = H * 7 THEN POKE 0,51: CALL SOUND: CALL SOUND: GOSUB 1530: GOTO 270
720 HCOLOR= 5: DRAW 1 AT P + 1,O + 1
730 HCOLOR= 0: DRAW 1 AT U + 1,T + 1:X = T / 8:Y = U / 7:Q = D(X,Y): GOSUB 1020:T = T + R:U = U + S
740 IF A = 1 AND CS = 0 THEN C9 = C9 + 500:CS = 1: POKE 0,105: CALL SOUND: CALL SOUND
750 IF A = 1 THEN 780
760 IF T1 = (T) AND U1 = U THEN T = T - R:U = U - S: GOSUB 1210:T = T + R:U = U + S
770 T1 = X * 8:U1 = Y * 7: IF T = I * 8 AND U = H * 7 THEN POKE 0,51: CALL SOUND: CALL SOUND: GOSUB 1530: GOTO 270
780 HCOLOR= 5: DRAW 1 AT U + 1,T + 1
790 RETURN
800 IF J = 1 AND PEEK ( - 16287) < 128 THEN RETURN
810 IF J = 0 AND K < > 13 THEN RETURN
820 ON G - 103 GOTO 890,920,830,860
830 M = I + 2: IF M > 18 THEN RETURN
840 N = H: IF D(M,N) = 97 OR D(M,N) = 99 THEN D(M,N) = D(M,N) - 1: GOTO 950
850 D(M,N) = D(M,N) + 1: GOTO 950
860 N = H + 2: IF N > 37 THEN RETURN
870 M = I: IF D(M,N) > 97 THEN D(M,N) = D(M,N) - 2: GOTO 950
880 D(M,N) = D(M,N) + 2: GOTO 950
890 M = I - 1: IF M < 2 THEN RETURN
900 N = H: IF D(M,N) = 97 OR D(M,N) = 99 THEN D(M,N) = D(M,N) - 1: GOTO 950
910 D(M,N) = D(M,N) + 1: GOTO 950
920 N = H - 1: IF N < 3 THEN RETURN
930 M = I: IF D(M,N) > 97 THEN D(M,N) = D(M,N) - 2: GOTO 950
940 D(M,N) = D(M,N) + 2
950 VTAB M + 1: HTAB N + 1: POKE 0,D(M,N): CALL C2: RETURN
960 HCOLOR= 0: DRAW 1 AT U2 + 1,T2 + 1:X = T2 / 8:Y = U2 / 7::Q = D(X,Y): GOSUB 1020:T2 = T2 + R:U2 = U2 + S
970 IF A = 1 THEN C9 = C9 + 1000: GOSUB 1530: GOTO 280
980 IF T3 = T2 AND U3 = U2 THEN T2 = T2 - R:U2 = U2 - S: GOSUB 1210:T2 = T2 + R:U2 = U2 + S
990 HCOLOR= 3: DRAW 1 AT U2 + 1,T2 + 1:T3 = X * 8:U3 = Y * 7: IF T2 = I * 8 AND U2 = H * 7 THEN POKE 0,57: CALL SOUND: CALL SOUND: GOSUB 1530: GOTO 270
1000 X = T2 / 8:Y = U2 / 7:S2 = 1:S3 = I:S4 = H: GOSUB 1350
1010 S2 = 3:S3 = T / 8:S4 = U / 7: GOSUB 1350:S2 = 2:S3 = O / 8:S4 = P / 7: GOSUB 1350: RETURN
1020 A = 0:X1 = 0:Y1 = 0:X2 = 0:Y2 = 0:R = 0:S = 0: ON Q - 95 GOTO 1030,1040,1060,1070
1030 IF X > 1 THEN X1 = X - 1
1040 IF Y > 2 THEN Y1 = Y - 1
1050 GOTO 1070
1060 IF X > 1 THEN X1 = X - 1
1070 IF X < 18 AND (D(X + 1,Y) = 96 OR D(X + 1,Y) = 98) THEN X2 = X + 1
1080 IF D(X,Y + 1) < 98 AND Y < 37 THEN Y2 = Y + 1
1090 IF X1 + Y1 + X2 + Y2 = 0 THEN A = 1: RETURN
1100 Z = - Z: IF Z = 1 THEN 1160
1110 IF X1 > 0 AND I < X THEN R = - 8: RETURN
1120 IF X2 > 0 AND I > X THEN R = 8: RETURN
1130 IF Y1 > 0 AND H < Y THEN S = - 7: RETURN
1140 IF Y2 > 0 AND H > Y THEN S = 7: RETURN
1150 GOTO 1200
1160 IF Y2 > 0 AND H > Y THEN S = 7: RETURN
1170 IF Y1 > 0 AND H < Y THEN S = - 7: RETURN
1180 IF X2 > 0 AND I > X THEN R = 8: RETURN
1190 IF X1 > 0 AND I < X THEN R = - 8: RETURN
1200 C0 = INT (4 * RND (1)) + 1: GOTO 1220
1210 C0 = ((R + 3 * S) / 16) + 2.5:C0 = INT (C0 + .5)
1220 Z = - Z: IF Z = 1 THEN 1290
1230 ON C0 GOTO 1250,1260,1270,1240
1240 IF Y1 > 0 THEN S = - 7:R = 0: RETURN
1250 IF X1 > 0 THEN R = - 8:S = 0! RETURN
1260 IF X2 > 0 THEN R = 8:S = 0: RETURN
1270 IF Y2 > 0 THEN S = 7:R = 0: RETURN
1280 GOTO 1240
1290 ON C0 GOTO 1300,1330,1320,1310
1300 IF Y2 > 0 THEN S = 7:R = 0: RETURN
1310 IF X2 > 0 THEN R = 8:S = 0: RETURN
1320 IF X1 > 0 THEN R = - 8:S = 0: RETURN
1330 IF Y1 > 0 THEN S = - 7:R = 0: RETURN
1340 GOTO 1300
1350 IF ABS (X - S3) < 2 AND ABS (Y - S4) < 2 THEN 1380
1360 IF ABS (X - S3) < 3 AND ABS (Y - S4) < 3 AND S2 = 1 THEN 1440
1370 RETURN
1380 IF S3 < X THEN D(S3,S4) = 96: VTAB S3 + 1: HTAB S4 + 1: POKE 0,96: CALL C2: RETURN
1390 IF S4 < Y THEN D(S3,S4) = 96: VTAB S3 + 1: HTAB S4 + 1: POKE 0,96: CALL C2: RETURN
1400 IF S3 > X THEN D(S3 + 1,S4) = 96: IF S3 < 18 THEN VTAB S3 + 2: HTAB S4 + 1: POKE 0,96: CALL C2
1410 IF S3 < 18 THEN RETURN
1420 D(S3,S4 + 1) = 96: IF S4 < 37 THEN VTAB S3 + 1: HTAB S4 + 2: POKE 0,96: CALL C2
1430 RETURN
1440 IF S4 < Y THEN D(S3,S4 + 1) = 96: VTAB S3 + 1: HTAB S4 + 2: POKE 0,96: CALL C2: RETURN
1450 IF S3 < X THEN D(S3 + 1,S4) = 96: VTAB S3 + 2: HTAB S4 + 1: POKE 0,96: CALL C2: RETURN
1460 IF S3 > X THEN D(S3,S4) = 96: VTAB S3 + 1: HTAB S4 + 1: POKE 0,96: CALL C2: RETURN
1470 D(S3,S4) = 96: VTAB S3 + 1: HTAB S4 + 1: POKE 0,96: CALL C2: RETURN
1480 HOME : VTAB 21: HTAB 12: PRINT "G A M E -- O V E R"
1490 HTAB 15: PRINT "SCORE = ";C9: PRINT " PRESS 'RETURN' FOR A NEW GAME."
1500 POKE - 16368,0
1510 IF PEEK ( - 16384) < > 141 THEN 1510
1520 POKE 0,J: CLEAR :J = PEEK (0): GOTO 210
1530 HCOLOR= 5: DRAW 1 AT P + 1,O + 1: DRAW 1 AT U + 1,T + 1: HCOLOR= 3: DRAW 1 AT U2 + 1,T2 + 1: POKE 0,103: CALL SOUND: CALL SOUND
1540 HCOLOR= 0: DRAW 1 AT P + 1,O + 1: DRAW 1 AT U + 1,T + 1: DRAW 1 AT U2 + 1,T2 + 1: VTAB I + 1: HTAB H + 1: POKE 0,D(I,H): CALL C2: RETURN
1550 DATA 169,17,208,2,169,36,141,37,3,169,7,133,6,37,0,10,10,10,170,165,41,105,24,133,41,164,36,24,165,41,105,4,133,41,189,47,3,17,40,145
1560 DATA 40,232,198,6,16,237,96,0,0,0,0,0,0,0,0,127,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,127,1,1,1,1,1,1,1,0
1570 DATA 0,24,60,126,24,24,24,0,0,8,12,126,126,12,8,0,0,24,24,24,126,60,24,0,0,16,48,126,126,48,16,0,4,0,54,46,46,46,44
1580 DATA 44,36,60,63,55,45,53,55,39,63,0,169,0,168,136,208,253,44,48,192,56,229,0,208,244,96,0
```

PROGRAM LISTING

HCM

WILD KINGDOM

COMMODORE 64

PROGRAM LISTING

```
100 REM  *********************
110 REM  *  WILD  KINGDOM  *
120 REM  *********************
130 REM  BY  MONTE  ULM
140 REM  AND  THE  HCM  STAFF
150 REM  HOME  COMPUTER  MAGAZINE
160 REM  VERSION  4.3.1
170 REM  C-64  BASIC
180 REM
190 GOTO  1980
200 SD=50:PT=PT+1:SO=54272
210 POKESO+24,15:POKESO+5,17:POKESO+6,241:POKESO,100
220 POKESO+1,SD:POKESO+4,17:FORD=1TO120:NEXT
230 POKESO+1,0:POKESO,16:RETURN
240 SD=20:GOTO  210
250 FORT=1TO3:SD=20:GOSUB  210:::SD=8:GOSUB  210:::NEXT:RETURN
260 POKESO,8:POKESO+1,8:POKESO+24,15
270 POKESO+5,31:POKESO+6,0
280 POKESO+4,129:FORD=1TO200:NEXT
290 POKESO+4,128:FORD=1TO50:NEXT:POKESO+24,0:POKESO,16:RETURN
300 REM  CALCULATE  ADDRESS
310 X=PEEK(SX+I):Y=PEEK(SY+I):IFI=2ORI=4THENM=1
320 IFI=0THENM=1
330 IFI=6THENM=8
340 XS=PEEK(MS)ANDM:IFXS>0THENX=X+256
350 K=(Y-50)/8:J=(X-24)/8:IFJ<1THENSTOP
360 AD=1024+40*K+J:B=PEEK(AD):BL=PEEK(AD-1):BR=PEEK(AD+1):BU=PEEK(AD-40)
370 BD=PEEK(AD+40):DD=PEEK(AD+80):LL=PEEK(AD-2)
380 RETURN
390 IFJS=0THEN  460
400 JY=PEEK(56320)AND31:IFJY=31THENRETURN
410 IF(JYAND16)=0THENZ$="U":GOTO  1360
420 IF(JY  AND  1)=  0THENZ$="I":GOTO  500
430 IF(JY  AND  2)=0THENZ$="M":GOTO  500
440 IF(JY  AND  4)=0THENZ$="J":GOTO  500
450 IF(JY  AND  8)=0THENZ$="K":GOTO  500
460 FORZL=1TO10:GETZ$:IFZ$=""THENNEXT
470 POKE198,0
480 IFZ$="U"THEN  1360
490 IFZ$<"I"ORZ$>"M"THEN  550
500 IFZL$=Z$THENI=6:GOSUB  300:::GOTO  560
510 ZL$=Z$:IFZ$="I"THENPOKEP,193:GOTO  550
520 IFZ$="K"THENPOKEP,194:GOTO  550
530 IFZ$="J"THENPOKEP,196:GOTO  550
540 POKEP,195
550 RETURN
560 Z=ASC(Z$)-72:ONZGOTO  610,  570,  630,  550,  650
570 IFBL=103ORBL=80ORX<33THENRETURN
580 X=X-8:IFX>255THENX=X-256:POKEMS,PEEK(MS)OR8:GOTO  600
590 IFX<256THENPOKEMS,PEEK(MS)AND247
600 POKESX+6,X:GOSUB  1310:::GOTO  200
610 IFB=99ORB=80ORY<59THENRETURN
620 Y=Y-8:POKESY+6,Y:GOSUB  1310:::GOTO  200
630 IFB=80ORB=103ORX>325THENRETURN
640 X=X+16:GOTO  580
650 IFBD=99ORBD=80ORY>225THENRETURN
660 Y=Y+16:GOTO  620
670 REM  MOVE  TIGERS
680 AR=1
690 GOSUB  300:::XD=0:YD=0
700 XH=PEEK(SX+6):IF(PEEK(MS)AND8)<>0THENXH=XH+256
710 XA=X
720 XB=PEEK(SX):IF(PEEK(MS)AND1)<>0THENXB=XB+256
730 XC=ABS(XB-XA):IFI=0THENXC=ABS(XB-XH)
740 YH=PEEK(SY+6)
750 YC=ABS(PEEK(SY)-Y):IFI=0THENYC=ABS(YH-Y)
760 XC=INT(SQR(XC!2+YC!2)):YC=XC
770 IFAR<3THEN  820
780 GOSUB  920:::IFXD>0THEN  800
790 GOSUB  1120:::IFYD=0THEN  840
800 POKESX+I,X:POKESY+I,Y:IFXD=1ORYD=1THENGOSUB  260
810 RETURN
820 GOSUB  1120:::IFYD>0THEN  800
830 GOSUB  920:::IFXD>0THEN  800
840 IFBL=103ORBL=80ORX<33THEN  860
850 GOSUB  1030:::GOTO  800
860 IFB=99ORB=80ORY=58THEN  880
870 Y=Y-8:YD=1:GOTO  800
880 IFB=80ORB=103ORX>326THEN  900
890 GOSUB  1110:::GOTO  800
900 IFBD=99ORBD=80ORY=226THEN  800
910 Y=Y+8:YD=1:GOTO  800
920 REM  X  CHANGE
930 IFAR=1THEN  970
940 IFAR=2THEN  1070
950 IFAR=3THEN  990
960 XD=0:RETURN
970 IFXH=XATHENXD=0:RETURN
980 IFXH>XATHEN  1070
990 IFXC<17THEN  1010
1000 IFBL=80ORBL=103THENXD=0:RETURN
1010 IFBL=80THENPOKEAD-1,99:GOTO  1030
1020 IFBL=103THENPOKEAD-1,32
1030 IFX<33THEN  960
1040 IFX>328THENX=X-16:GOTO  960
1050 X=X-8:IFX>255THENPOKEMS,PEEK(MS)ORI:X=X-256:IFI=0THENPOKEMS,PEEK(MS)OR1
1060 XD=1:RETURN
1070 IFXC<17THEN  1090
1080 IFB=80ORB=103THENXD=0:RETURN
1090 IFB=80THENPOKEAD,99:GOTO  1110
1100 IFB=103THENPOKEAD,32
1110 X=X+16:GOTO  1030
1120 REM  Y  CHANGE
1130 IFAR=1THEN  1170
1140 IFAR=2THEN  1260
1150 IFAR=3THEN  1190
1160 YD=0:RETURN
1170 IFYH=YTHENDY=0:RETURN
1180 IFYH>YTHEN  1260
1190 IFYC<17THEN  1210
1200 IFB=99ORB=80THENYD=0:RETURN
1210 IFB=99THENPOKEAD,32:GOTO  1230
1220 IFB=80THENPOKEAD,103
1230 IFY<59THEN  1160
1240 IFY>234THENY=226:GOTO  1160
1250 Y=Y-8:YD=1:RETURN
1260 IFYC<17THEN  1280
1270 IFBD=80ORBD=99THENYD=0:RETURN
1280 IFBD=80THENPOKEAD+40,103:GOTO  1300
1290 IFBD=99THENPOKEAD+40,32
1300 Y=Y+16:GOTO  1240
1310 PRINT"■HOME■CTRL RVSOFF■24CRSRDOWN■"SPC(24)"SCORE"PT;:RETURN
1320 REM  CHECK  FOR  CAT  COINCIDENCE  AND  DEATH!
1330 FOR  I  =0TO4STEP2:IFPEEK(SX+I)<>PEEK(SX+6)THEN  1350
1340 IFPEEK(SY+I)=PEEK(SY+6)THEN  1690
1350 NEXT:GOSUB  2460:::RETURN
1360 I=6:GOSUB  300
1370 PK=PEEK(2043):IFPK>193THEN  1440
1380 IFY<72THEN  1640
1390 IFBU=99THENBU=32:GOTO  1430
1400 IFBU=80THENBU=103:GOTO  1430
1410 IFBU=32THENBU=99:GOTO  1430
1420 IFBU=103THENBU=80
1430 POKEAD-40,BU:GOSUB  240:::GOTO  1640
1440 IFPK>194THEN  1510
1450 IFX>312THENRETURN
1460 IFBR=80THENBR=99:GOTO  1500
1470 IFBR=99THENBR=80:GOTO  1500
1480 IFBR=103THENBR=32:GOTO  1500
1490 IFBR=32THENBR=103
1500 POKEAD+1,BR:GOSUB  240:::GOTO  1640
1510 IFPK>195THEN  1580
1520 IFY>218THENRETURN
1530 IFDD=99THENDD=32:GOTO  1570
1540 IFDD=80THENDD=103:GOTO  1570
1550 IFDD=103THENDD=80:GOTO  1570
1560 IFDD=32THENDD=99
1570 POKEAD+80,DD:GOSUB  240:::GOTO  1640
1580 IFX<48THENRETURN
1590 IFLL=80THENLL=99:GOTO  1630
1600 IFLL=99THENLL=80:GOTO  1630
1610 IFLL=103THENLL=32:GOTO  1630
1620 IFLL=32THENLL=103
1630 POKEAD-2,LL:GOSUB  240
1640 RETURN
1650 RETURN
1660 GOSUB  390:::GOSUB  1330:::I=2:GOSUB  670:::GOSUB  390:::GOSUB  1330:::I=4:GOSUB  670:::GOSUB  1330
1670 GOSUB  390:::GOSUB  1330
1680 I=0:GOSUB  670:::GOSUB  1330:::GOTO  1660
1690 REM  DEATH  OF  A  HUNTER
1700 MM=PEEK(MS):IF(MMANDB)=0THEN  1750
1710 IFI=4AND(MMAND12)=12THEN  1750
1720 IFI=2AND(MMAND10)=10THEN  1750
1730 IFI=0AND(MMAND9)=9THEN  1750
1740 GOTO  1350
1750 GOSUB  250
1760 IF(PEEK(SE)AND112)=0THENGOTO  2380
1770 IF(PEEK(SE)AND112)=112THENPOKESE,63:GOTO  1800
1780 IF(PEEK(SE)ANDB)=8THENPOKESE,31:GOTO  1800
1790 POKESE,15
1800 R=PEEK(SE)
1810 REM  RESTART  GAME
1820 POKESE,0:POKESX+6,32:POKESY+6,58:POKE2043,194:POKESX,80:POKESY,106
1830 POKESY+2,74:POKESX+4,48:POKESY+4,12
2:POKESX+2,96
1840 POKEMS,0:GOSUB  2550:::POKESE,R:IFZZ=1  THEN  RETURN
1850 ZZ=1:GOTO  1660
1860 PRINT"■SHIFT CLR■";
1870 PRINT"■CTRL RVSON■                                                    ";
1880 FORI=1TO22:PRINT"■CTRL RVSON■ "SPC(38)"■CTRL RVSON■ ";:NEXT
1890 PRINT"■CTRL RVSON■                                                    ";
```

Continued

WILD KINGDOM

COMMODORE 64

```
1900 GOSUB  1310
1910 FORI=1TO22:FORJ=1TO38:A=INT(RND(1)*7)+1
1920 IFJ=1ANDI=1THENNEXT
1930 AD=1024+I*40+J:AC=55296+I*40+J:ONAGOTO 1940, 1940, 1950, 1960, 1960, 1970, 1970
1940 IFJ<38THENPOKEAD,103:IFI=1ANDJ=1THENRUN
1950 IFJ<38ANDI>1THENPOKEAD,80:GOTO  1970
1960 IFI>1THENPOKEAD,99
1970 POKEAC,9:NEXT:NEXT:RETURN
1980 SE=53269:MS=53264:SX=53248:SY=53249:POKESX+8,40:POKESX+10,48:POKESY+8,242
1990 ZL$="K":P=2043:DIML(6),O(6)
2000 POKESY+10,242:C1=53287:POKEC1,0:POKEC1+1,7:POKEC1+2,7:POKEC1+3,13:POKEC1+4,0
2010 POKESY+12,242:POKESX+12,56:POKE2046,194
2020 POKEC1+5,0:FORI=0TO2:POKE2040+I,192:POKE2043+I,194:NEXT:R=127:POKE53280,9
2030 POKE53281,5:POKE646,9:POKEC1+6,0
2040 PRINT"{SHIFT CLR}{6CRSRDOWN}";
2050 PRINTSPC(5)"******************************"
2060 PRINTSPC(5)"******************************"
2070 PRINTSPC(5)"**"SPC(26)"**"
2080 PRINTSPC(5)"**"SPC(26)"**"
2090 PRINTSPC(5)"**"SPC(9)"W I L D "SPC(9)"**"
2100 PRINTSPC(5)"**"SPC(26)"**"
2110 PRINTSPC(5)"**"SPC(6)"K I N G D O M{CRSRRIGHT}"SPC(6)"**"
2120 PRINTSPC(5)"**"SPC(26)"**"
2130 PRINTSPC(5)"**"SPC(26)"**"
2140 PRINTSPC(5)"******************************"
2150 PRINTSPC(5)"******************************"
2160 FORI=0TO319:READDT:POKE12288+I,DT:NEXT
2170 PRINT"{6CRSRDOWN}"SPC(20)"USE JOYSTICKS?  (Y/N)
2180 GETZ$:IFZ$=""THEN  2180
2190 IFZ$="Y"THENJS=1:GOTO  2450
2200 IFZ$="N"THENJS=0:GOTO  2450
2210 GOTO  2180
2220 REM SPRITE DATA
2230 DATA0,0,0,231,0,0,255,0,0,90,0,0,126,0,0,60,0,0,0,0,0
2240 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2250 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2260 DATA24,00,0,60,0,0,126,0,0,24,0,0,24,0,0,24,0,0,0,0,0
2270 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2280 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2290 DATA0,0,0,12,0,0,14,0,0,255,0,0,14,0,0,12,0,0,0,0,0
2300 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2310 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2320 DATA0,0,0,24,0,0,24,0,0,24,0,0,126,0,0,60,0,0,24,0,0
2330 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2340 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2350 DATA0,0,0,48,0,0,112,0,0,255,0,0,112,0,0,48,0,0,0,0,0
2360 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2370 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2380 POKE53281,11:POKE646,7
2390 PRINT"{SHIFT CLR}{10CRSRDOWN}{5CRSRRIGHT}GAME OVER. "
2400 PRINT:PRINT"{5CRSRRIGHT}DO YOU WISH TO PLAY AGAIN?  (Y/N)"
2410 GOSUB  1310
2420 GETZ$:IFZ$=""THEN  2420
2430 IFZ$="N"THENEND
2440 IFZ$<>"Y"THEN  2420
2450 POKESE,0:R=127:POKE53280,5:POKE53281,5:POKE646,9:GOSUB  1860:::GOTO  1820
2460 FORI=0TO4STEP2:GOSUB  300
2470 IFBL<>103ANDBL<>80ANDBL<>160THEN  2540
2480 IFB<>80ANDB<>99ANDB<>160THEN  2540
2490 IFB<>103ANDB<>80ANDB<>160THEN  2540
2500 IFBD<>99ANDBD<>80ANDBD<>160THEN  2540
2510 IFI=0THENPT=PT+1000:GOTO  2380
2520 IFPEEK(SX+I)=L(I)ANDPEEK(SY+I)=O(I)THEN  2540
2530 PT=PT+500:GOSUB  1310:::L(I)=PEEK(SX+I):O(I)=PEEK(SY+I)
2540 NEXT:RETURN
2550 FORI=0TO4STEP2:GOSUB  300
2560 POKEAD-1,32:POKEAD,32:POKEAD+40,32
2570 NEXT:RETURN
```

HCM

WILD KINGDOM

IBM PC & PCjr

```
100 REM ***************
110 REM * WILD KINGDOM *
120 REM ***************
130 REM BY MONTE ULM
140 REM AND THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.3.1
170 REM IBM PCjr CARTRIDGE BASIC
180 REM AND 64K MEMORY EXPANSION
190 REM IBM PC BASICA W/COLOR MONITOR
200 KEY OFF
210 CLS:SCREEN 1:DIM C1(12),C2(12),D(23,32),HB(10),CAT(3,2),CIT(3):COLOR 1,0:LOCATE 12,14:PRINT "WILD KINGDOM":FOR TD=1 TO 2000:NEXT:CLS:COLOR ,0:DEF FNX(Z)=(Z-1)*8+8:DEF FNY(Z)=(Z-1)*10+6
220 DEF FNCX=(CAT(CN,1)-1)*8+10:DEF FNCY=(CAT(CN,2)-1)*10+9:FOR Z=1 TO 31:D(23,Z)=2:NEXT:FOR Z=1 TO 22:D(Z,32)=1:NEXT
230 DEF FNG=D(INT(HX/8),INT(HY/10)):DEF FNG1=D(INT(HX/8),INT(HY/10)+1):DEF FNG2=D(INT(HX/8)+1,INT(HY/10)):DEF FNJ=D(CAT(CN,1),CAT(CN,2)):DEF FNK=D(CAT(CN,1)+1,CAT(CN,2)):RANDOMIZE TIMER
240 DEF FNL=D(CAT(CN,1),CAT(CN,2)+1):MEN=3:SCORE=0:C$="RUR4DRDL3NUL3F2R2EU":H$="A=ANG;C1NU2LDGR4HLEL;":GET(100,50)-(107,56),HB:DEF FNGX=INT(HX/8):DEF FNGY=INT(HY/10)
250 DEF FNDX=HX-FNCX-2:DEF FNDY=HY-FNCY-2
260 DRAW "BM10,30C2;XC$;BM30,30C3;XC$;":GET(10,29)-(16,33),C1:GET(30,29)-(36,33),C2
270 CLS:FOR Z=0 TO 3:LINE (5-Z,7-Z)-(316+Z,184+Z),2,B:NEXT
280 FOR X=1 TO 22:FOR Y=1 TO 31:R=INT(RND*4):D(X,Y)=R:ON R+1 GOSUB 810,820,830,840:NEXT:NEXT
290 Y=5:X=3:GOSUB 810:HY=51:HX=28:GOSUB 860:CAT(1,1)=10:CAT(1,2)=9:CAT(2,1)=9:CAT(2,2)=10:CAT(3,1)=10:CAT(3,2)=10
300 X=10:Y=10:GOSUB 810:X=9:GOSUB 810:X=10:Y=9:GOSUB 810:CN=1:PUT(FNCY,FNCX),C2,PSET:CN=2:PUT(FNCY,FNCX),C2,PSET:CN=3:PUT(FNCY,FNCX),C1,PSET
310 ON KEY(11) GOSUB 680:ON KEY(12) GOSUB 690:ON KEY(13) GOSUB 700:ON KEY(14) GOSUB 710:KEY 15,CHR$(&H40)+CHR$(&H39):ON KEY(15) GOSUB 720:KEY(15) ON
320 LOCATE 25,1:PRINT "SCORE:";SCORE;TAB(20);"MEN  ";:IF MEN>-1 THEN PRINT STRING$(MEN,16);"    "; ELSE PRINT "  ";
330 GOSUB 400:IF EAT=1 THEN GOTO 370 ELSE SCORE=SCORE+CIT(1)*2+CIT(2)*2:IF CIT(3)=1 THEN 360 ELSE LOCATE 25,7:PRINT SCORE;
340 KEY(11) ON:KEY(12) ON:KEY(13) ON:KEY(14) ON:KEY(11) STOP:KEY(12) STOP:KEY(13) STOP:KEY(14) STOP:FOR CN=1 TO 3:IF FNDX=0 AND FNDY=0 THEN GOTO 370
350 NEXT:GOTO 330
360 SCORE=SCORE+1000:SOUND 440,5:GOTO 270
370 SOUND 110,5:EAT=0:MEN=MEN-1:GOSUB 850:FOR CN=1 TO 3:PUT(FNCY,FNCX),HB,PSET:NEXT:IF MEN>-1 THEN GOTO 290
380 FOR Z=1000 TO 110:SOUND Z,1,15:NEXT:CLS:LOCATE 12,1:PRINT "YOUR FINAL SCORE IS :";SCORE:PRINT:PRINT "WOULD YOU LIKE TO PLAY AGAIN (Y/N)
390 Z$=INKEY$:IF Z$="" THEN 390 ELSE IF Z$="N" THEN END ELSE IF Z$="Y" THEN RUN ELSE GOTO 390
400 FOR CN=1 TO 3:IF FNJ=3 AND FNK>1 AND (FNL=1 OR FNL=3) THEN CIT(CN)=1:GOTO 630 ELSE CIT(CN)=0
```

Continued

WILD KINGDOM

Continued

IBM PC & PCjr

```
410 IF RND*10<3 THEN 500 ELSE RZ=ABS(RZ
    -1):IF RZ=1 THEN GOTO 460
420 IF FNDY>0 AND (FNL=0 OR FNL=2) THEN
     MCY=1:GOTO 600
430 IF FNDY<0 AND (FNJ=0 OR FNJ=2) THEN
     MCY=-1:GOTO 600
440 IF FNDX>0 AND FNK<2 THEN MCX=1:GOTO
    600
450 IF FNDX<0 AND FNJ<2 THEN MCX=-1:GOT
    O 600 ELSE GOTO 500
460 IF FNDX<0 AND FNJ<2 THEN MCX=-1:GOT
    O 600
470 IF FNDX>0 AND FNK<2 THEN MCX=1:GOTO
    600
480 IF FNDY>0 AND (FNL=0 OR FNL=2) THEN
     MCY=1:GOTO 600
490 IF FNDY<0 AND (FNJ=0 OR FNJ=2) THEN
     MCY=-1:GOTO 600
500 R=INT(RND*4)+1
510 RZ=ABS(RZ-1):IF RZ=1 THEN ON R GOTO
    520,530,540,550 ELSE ON R GOTO 560
    ,570,580,590
520 IF FNJ=0 OR FNJ=2 THEN MCY=-1:GOTO
    600
530 IF FNL=0 OR FNL=2 THEN MCY=1:GOTO 6
    00
540 IF FNJ<2 THEN MCX=-1:GOTO 600
550 IF FNK<2 THEN MCX=1:GOTO 600 ELSE G
    OTO 520
560 IF FNK<2 THEN MCX=1:GOTO 600
570 IF FNJ<2 THEN MCX=-1:GOTO 600
580 IF FNL=0 OR FNL=2 THEN MCY=1:GOTO 6
    00
590 IF FNJ=0 OR FNJ=2 THEN MCY=-1:GOTO
    600 ELSE GOTO 560
600 IF CAT(CN,1)+MCX<1 OR CAT(CN,1)+MCX
    >22 OR CAT(CN,2)+MCY<1 OR CAT(CN,2)
    +MCY>31 THEN 630
610 IF CN<3 THEN PUT(FNCY-1,FNCX-1),HB,
    PSET:CAT(CN,1)=CAT(CN,1)+MCX:CAT(CN
    ,2)=CAT(CN,2)+MCY:PUT(FNCY,FNCX),C2
    :GOTO 630
620 PUT (FNCY-1,FNCX-1),HB,PSET:CAT(CN,
    1)=CAT(CN,1)+MCX:CAT(CN,2)=CAT(CN,2
    )+MCY:PUT(FNCY,FNCX),C1
630 IF CN<3 THEN IF ABS(CAT(CN,1)-CAT(3
    ,1))+ABS(CAT(CN,2)-CAT(3,2))<4 THEN
     X=CAT(CN,1)+INT(RND*3)-1:Y=CAT(CN,
    2)+INT(RND*3)-1:GOSUB 810 ELSE IF R
    ND*10<2 THEN GOSUB 660
640 SOUND 660,1:IF FNDX=0 AND FNDY=0 TH
    EN EAT=1
650 MCX=0:MCY=0:NEXT CN:RETURN
660 IF FNDX+FNDY>36 AND RND*10>2 THEN R
    ETURN
670 X=CAT(3,1)+INT(RND*3)-1:Y=CAT(3,2)+
    INT(RND*3)-1:GOSUB 810:RETURN
680 IF ANG<>0 THEN ANG=0:GOSUB 850:GOSU
    B 860:RETURN ELSE IF HX>12 AND FNG<
    >3 AND FNG<>2 THEN GOSUB 850:HX=HX-
    8:GOSUB 860:GOSUB 870:RETURN ELSE R
    ETURN
690 IF ANG<>1 THEN ANG=1:GOSUB 850:GOSU
    B 860:RETURN ELSE IF HY>11 AND FNG<
    >1 AND FNG<>3 THEN GOSUB 850:HY=HY-
    10:GOSUB 860:GOSUB 870:RETURN ELSE
    RETURN
700 IF ANG<>3 THEN ANG=3:GOSUB 850:GOSU
    B 860:RETURN ELSE IF HY<308 AND FNG
    1<>1 AND FNG1<>3 THEN GOSUB 850:HY=
    HY+10:GOSUB 860:GOSUB 870:RETURN EL
    SE RETURN
710 IF ANG<>2 THEN ANG=2:GOSUB 850:GOSU
    B 860:RETURN ELSE IF HX<178 AND FNG
    2<>3 AND FNG2<>2 THEN GOSUB 850:HX=
    HX+8:GOSUB 860:GOSUB 870:RETURN ELS
    E RETURN
720 IF (ANG=0 AND FNGX<3) OR (ANG=1 AND
     FNGY<3) OR (ANG=2 AND FNGX>20) OR
    (ANG=3 AND FNGY>29) THEN RETURN ELS
    E ON ANG+1 GOTO 730,750,770,790
730 GT=D(FNGX-1,FNGY):IF GT=2 OR GT=3 T
    HEN D(FNGX-1,FNGY)=D(FNGX-1,FNGY)-2
     ELSE D(FNGX-1,FNGY)=D(FNGX-1,FNGY)
    +2
740 X=FNGX-1:Y=FNGY:ON D(FNGX-1,FNGY)+1
     GOSUB 810,820,830,840:RETURN
750 GT=D(FNGX,FNGY-1):IF GT=1 OR GT=3 T
    HEN D(FNGX,FNGY-1)=D(FNGX,FNGY-1)-1
     ELSE D(FNGX,FNGY-1)=D(FNGX,FNGY-1)
    +1
760 X=FNGX:Y=FNGY-1:ON D(FNGX,FNGY-1)+1
     GOSUB 810,820,830,840:RETURN
770 GT=D(FNGX+2,FNGY):IF GT=2 OR GT=3 T
    HEN D(FNGX+2,FNGY)=D(FNGX+2,FNGY)-2
     ELSE D(FNGX+2,FNGY)=D(FNGX+2,FNGY)
    +2
780 X=FNGX+2:Y=FNGY:ON D(FNGX+2,FNGY)+1
     GOSUB 810,820,830,840:RETURN
790 GT=D(FNGX,FNGY+2):IF GT=1 OR GT=3 T
    HEN D(FNGX,FNGY+2)=D(FNGX,FNGY+2)-1
     ELSE D(FNGX,FNGY+2)=D(FNGX,FNGY+2)
    +1
800 X=FNGX:Y=FNGY+2:ON D(X,Y)+1 GOSUB 8
    10,820,830,840:RETURN
810 IF X>22 OR X<1 OR Y>31 OR Y<1 THEN
    RETURN ELSE D(X,Y)=0:LINE (FNY(Y),F
    NX(X))-(FNY(Y)+9,FNX(X)),0:LINE (FN
    Y(Y),FNX(X))-(FNY(Y)+1,FNX(X)+7),0,
    B:RETURN
820 IF X>22 OR X<1 OR Y>31 OR Y<1 THEN
    RETURN ELSE IF Y=1 AND X=1 THEN 810
     ELSE IF Y=1 THEN 830 ELSE LINE (FN
    Y(Y),FNX(X))-(FNY(Y)+9,FNX(X)),0:LI
    NE (FNY(Y),FNX(X))-(FNY(Y)+1,FNX(X)
    +7),2,B:D(X,Y)=1:RETURN
830 IF X>22 OR X<1 OR Y>31 OR Y<1 THEN
    RETURN ELSE IF X=1 THEN 820 ELSE LI
    NE (FNY(Y),FNX(X))-(FNY(Y)+9,FNX(X)),2:LI
    NE (FNY(Y),FNX(X)+1)-(FNY(Y)+1,FNX(
    X)+7),0,B:D(X,Y)=2:RETURN
840 IF X>22 OR X<1 OR Y>31 OR Y<1 THEN
    RETURN ELSE IF Y=1 THEN 830 ELSE IF
     X=1 THEN 820 ELSE LINE (FNY(Y),FNX
    (X))-(FNY(Y)+9,FNX(X)),2:LINE (FNY(
    Y),FNX(X))-(FNY(Y)+1,FNX(X)+7),2,B:
    D(X,Y)=3:RETURN
850 PUT(HY-3,HX-3),HB,PSET:RETURN
860 DRAW "BM=HY;,=HX;XH$;":SOUND 1000,2
    :RETURN
870 FOR CN=1 TO 3:IF FNDX=0 AND FNDY=0
    THEN EAT=1
880 NEXT:RETURN
```

HCM

WILD KINGDOM

TI-99/4A

```
100 REM ********************
110 REM * WILD KINGDOM *
120 REM ********************
130 REM BY MONTE ULM
140 REM HOME COMPUTER MAGAZINE
150 REM VERSION 4.3.1
160 REM TI EXTENDED BASIC
170 REM
180 CALL CLEAR :: DISPLAY AT(12,8):"WIL
    D KINGDOM" :: DISPLAY AT(22,1):"USE
     JOYSTICKS (Y/N):" :: ACCEPT AT(22,
    21)VALIDATE("YN")SIZE(1):FR$
190 IF FR$="Y" THEN J=1 :: CALL CLEAR :
    : DISPLAY AT(12,5)BEEP:"PLACE ALPHA
    LOCK UP" ELSE GOTO 210
200 FOR TD=1 TO 500 :: NEXT TD
210 RANDOMIZE :: GOSUB 220 :: GOTO 390
220 C9=0 :: MEN=3 :: Z=1 :: DIM D(23,29
    )
230 CALL CHAR(96,"00",97,"FFFF",98,"C0C
    0C0C0C0C0C0C0FFFFC0C0C0C0C0C0C0")
240 CALL CHAR(100,"FFFFFFFFFFFFFFFF"):::
     CALL COLOR(9,12,3)
250 CALL CHAR(112,"00007E5A7E3C18000"):
    : CALL COLOR(11,7,1)
260 RANDOMIZE :: CALL CLEAR :: CALL SCR
    EEN(7)
270 CALL HCHAR(1,4,100,26):: CALL HCHAR
    (23,4,100,26):: CALL VCHAR(2,4,100,
    21):: CALL VCHAR(2,29,100,21)
280 FOR A=1 TO 21 :: FOR B=4 TO 27 :: C
    =INT(4*RND)+96
290 IF B=4 AND C>97 THEN C=C-2 ELSE IF
    A=1 AND(C=97 OR C=99)THEN C=C-1
300 D(A,B)=C :: CALL HCHAR(A+1,B+1,C)::
     NEXT B :: NEXT A
310 CALL CHAR(104,"0018244281818000000020
    4098984020000000181842241800000804 3
    2320408")
320 MEN=MEN-1 :: IF MEN=-1 THEN 1170
330 O=40 :: P=80 :: T=48 :: U=72
340 CAP,CAS=0 :: CALL HCHAR(24,4,113,2)
    :: CALL HCHAR(24,4,107,MEN):: DISPL
    AY AT(24,8):"SCORE ";C9
350 T2=48 :: U2=80 :: D(6,10)=96 :: CAL
    L HCHAR(7,11,96)
360 CALL SPRITE(#2,112,11,O+1,P+2):: CA
    LL SPRITE(#3,112,11,T+1,U+2)
370 CALL SPRITE(#4,112,2,T2+1,U2+2)
380 CALL HCHAR(2,6,96):: D(1,5)=96 :: I
    =1 :: H=4 :: G=107 :: GOSUB 520 ::
    RETURN
390 GOSUB 400 :: GOSUB 540 :: GOSUB 620
    :: GOSUB 400 :: GOSUB 780 :: GOSUB
     620 :: GOSUB 740 :: GOTO 390
400 IF J=0 THEN CALL JOYST(2,E,F):: J=(
    (E+3*F)/4)+5 :: ON J GOTO 530,480,5
    30,420,530,460,530,440,530
410 CALL KEY(1,K,S):: IF S=0 OR K>5 THE
    N RETURN ELSE ON K+1 GOTO 480,530,4
    20,460,530,440
420 IF G=105 THEN 430 ELSE G=105 :: GOT
    O 520
430 IF D(I,H)>97 THEN 520 ELSE H=H-1 ::
     GOTO 500
440 IF G=104 THEN 450 ELSE G=104 :: GOT
    O 520
```

Continued

TI-99/4A

```
450 IF D(I,H)=97 OR D(I,H)=99 THEN 520
    ELSE I=I-1 :: GOTO 510
460 IF G=107 THEN 470 ELSE G=107 :: GOT
    O 520
470 IF D(I,H+1)<98 THEN H=H+1 :: GOTO 5
    00 :: ELSE GOTO 520
480 IF G=106 THEN 490 ELSE G=106 :: GOT
    O 520
490 IF D(I+1,H)=96 OR D(I+1,H)=98 THEN
    I=I+1 :: GOTO 510 :: ELSE GOTO 520
500 IF H<4 THEN H=4 :: RETURN :: IF H>2
    7 THEN H=27 :: RETURN :: ELSE
    520
510 IF I<1 THEN I=1 :: RETURN :: ELSE I
    F I>21 THEN I=21 :: RETURN
520 CALL SPRITE(#1,G,16,(I*8)+1,(H*8)+2
    ):: CALL SOUND(-50,880,0):: C9=C9+1
    :: DISPLAY AT(24,14):C9
530 RETURN
540 X=O/8 :: Y=P/8 :: Q=D(X,Y):: GOSUB
    800 :: O=O+R :: P=P+S
550 IF A=1 THEN IF CAP=0 THEN C9=C9+500
    :: CAP=1 :: GOTO 580 :: ELSE 580
560 IF O1=O AND P1=P THEN O=O-R :: P=P-
    S :: GOSUB 960 :: O=O+R :: P=P+S
570 CALL LOCATE(#2,O+1,P+2):: CALL SOUN
    D(-50,-6,0):: O1=X*8 :: P1=Y*8 :: I
    F X=T/8 :: I=I*8 AND Y=U/8 :: P=H*8 Q=D(X,Y):: GOSUB
    800 :: T=T+R :: U=U+S
590 IF A=1 THEN IF CAS=0 THEN C9=C9+500
    :: CAS=1 :: RETURN :: ELSE RETURN
600 IF T1=T AND U1=U THEN T=T-R :: U=U-
    S :: GOSUB 960 :: T=T+R :: U=U+S
610 CALL LOCATE(#3,T+1,U+2):: CALL SOUN
    D(-50,-6,0):: T1=X*8 :: U1=Y*8 :: I
    F T=I*8 AND U=H*8 THEN 320 ELSE RET
    URN
620 CALL KEY(2,K,L):: IF L=0 THEN RETUR
    N
630 ON G-103 GOTO 690,710,640,670
640 M=I+2 :: IF M>21 THEN RETURN
650 N=H :: IF D(M,N)=97 OR D(M,N)=99 TH
    EN D(M,N)=D(M,N)-1 :: GOTO 730
660 D(M,N)=D(M,N)+1 :: GOTO 730
670 N=H+2 :: IF N>27 THEN RETURN
680 M=I :: IF D(M,N)>97 THEN D(M,N)=D(M
    ,N)-2 :: GOTO 730 :: ELSE D(M,N)=D(
    M,N)+2 :: GOTO 730
690 M=I-1 :: IF M<2 THEN RETURN
700 N=H :: IF D(M,N)=97 OR D(M,N)=99 TH
    EN D(M,N)=D(M,N)-1 :: GOTO 730 :: E
    LSE D(M,N)=D(M,N)+1 :: GOTO 730
710 N=H-1 :: IF N<5 THEN RETURN
720 M=I :: IF D(M,N)>97 THEN D(M,N)=D(M
    ,N)-2 :: GOTO 730 :: ELSE D(M,N)=D(
    M,N)+2
730 CALL HCHAR(M+1,N+1,D(M,N)):: CALL S
    OUND(-50,-3,0):: RETURN
740 X=T2/8 :: Y=U2/8 :: Q=D(X,Y):: GOSU
    B 800 :: T2=T2+R :: U2=U2+S
750 IF A=1 THEN C9=C9+1000 :: GOTO 330
760 IF T3=T2 AND U3=U2 THEN T2=T2-R ::
    U2=U2-S :: GOSUB 960 :: T2=T2+R ::
    U2=U2+S
770 CALL LOCATE(#4,T2+1,U2+2):: CALL SO
    UND(-50,-5,0):: T3=X*8 :: U3=Y*8 ::
     IF T2=I*8 AND U2=H*8 THEN 320
780 X=T2/8 :: Y=U2/8 :: S2=1 :: S3=I ::
     S4=H :: GOSUB 1080
790 S2=3 :: S3=T/8 :: S4=U/8 :: GOSUB 1
    080 :: S2=2 :: S3=O/8 :: S4=P/8 ::
    GOSUB 1080 :: RETURN
800 A,X1,Y1,X2,Y2,R,S=0 :: ON Q-95 GOTO
    810,820,830,840
810 IF X>1 THEN X1=X-1
820 IF Y>4 THEN Y1=Y-1 :: GOTO 840 :: E
    LSE 840
830 IF X>1 THEN X1=X-1
840 IF X<21 THEN IF D(X+1,Y)=96 OR D(X+
    1,Y)=98 THEN X2=X+1
850 IF D(X,Y+1)<98 AND Y<27 THEN Y2=Y+1
860 IF X1+Y1+X2+Y2=0 THEN A=1 :: RETURN
    :: ELSE Z=Z*-1 :: IF Z=1 THEN 910
    :: ELSE 870
870 IF X1>0 AND I<X THEN R=-8 :: RETURN
880 IF X2>0 AND I>X THEN R=8 :: RETURN
890 IF Y1>0 AND H<Y THEN S=-8 :: RETURN
900 IF Y2>0 AND H>Y THEN S=8 :: RETURN
    :: ELSE 950
910 IF Y2>0 AND H>Y THEN S=8 :: RETURN
920 IF Y1>0 AND H<Y THEN S=-8 :: RETURN
930 IF X2>0 AND I>X THEN R=8 :: RETURN
940 IF X1>0 AND I<X THEN R=-8 :: RETURN
950 C1=INT(4*RND)+1 :: GOTO 970
960 C1=((R+3*S)/16)+2.5
970 Z=Z*-1 :: IF Z=1 THEN 1030
980 ON C1 GOTO 1000,1010,1020,990
990 IF Y1>0 THEN S=-8 :: R=0 :: RETURN
1000 IF X1>0 THEN R=-8 :: S=0 :: RETURN
1010 IF X2>0 THEN R=8 :: S=0 :: RETURN
1020 IF Y2>0 THEN S=8 :: R=0 :: RETURN :
     :: ELSE 990
1030 ON C1 GOTO 1040,1070,1060,1050
1040 IF Y2>0 THEN S=8 :: R=0 :: RETURN
1050 IF X2>0 THEN R=8 :: S=0 :: RETURN
1060 IF X1>0 THEN R=-8 :: S=0 :: RETURN
1070 IF Y1>0 THEN S=-8 :: R=0 :: RETURN
     :: ELSE 1040
1080 CALL LDIST ANCE(#4,#S2,D1):: IF D1=64
     THEN 1090 ELSE IF D1=256 AND S2=1
     THEN 1130 ELSE RETURN
1090 IF S3<X THEN D(S3,S4)=96 :: CALL HC
     HAR(S3+1,S4+1,96):: RETURN
1100 IF S4<Y THEN D(S3,S4)=96 :: CALL HC
     HAR(S3+1,S4+1,96):: RETURN
1110 IF S3>X THEN D(S3+1,S4)=96 :: IF S3
     <21 THEN CALL HCHAR(S3+2,S4+1,96)::
     RETURN :: ELSE RETURN
1120 D(S3,S4+1)=96 :: IF S4<27 THEN CALL
     HCHAR(S3+1,S4+2,96):: RETURN :: ELSE
     RETURN
1130 IF S4<Y THEN D(S3,S4+1)=96 :: CALL
     HCHAR(S3+1,S4+2,96):: RETURN
1140 IF S3<X THEN D(S3+1,S4)=96 :: CALL
     HCHAR(S3+2,S4+1,96):: RETURN
1150 IF S3>X THEN D(S3,S4)=96 :: CALL HC
     HAR(S3+1,S4+1,96):: RETURN
1160 D(S3,S4)=96 :: CALL HCHAR(S3+1,S4+1
     ,96):: RETURN
1170 CALL CLEAR :: CALL DELSPRITE(ALL)::
     PRINT :: "GAME OVER";C9;:: "PRESS ENTER
     FOR A NEW GAME";:: :: ::
1190 CALL KEY(0,K,S):: IF S=0 THEN 1190
     ELSE 220
1200 END
```

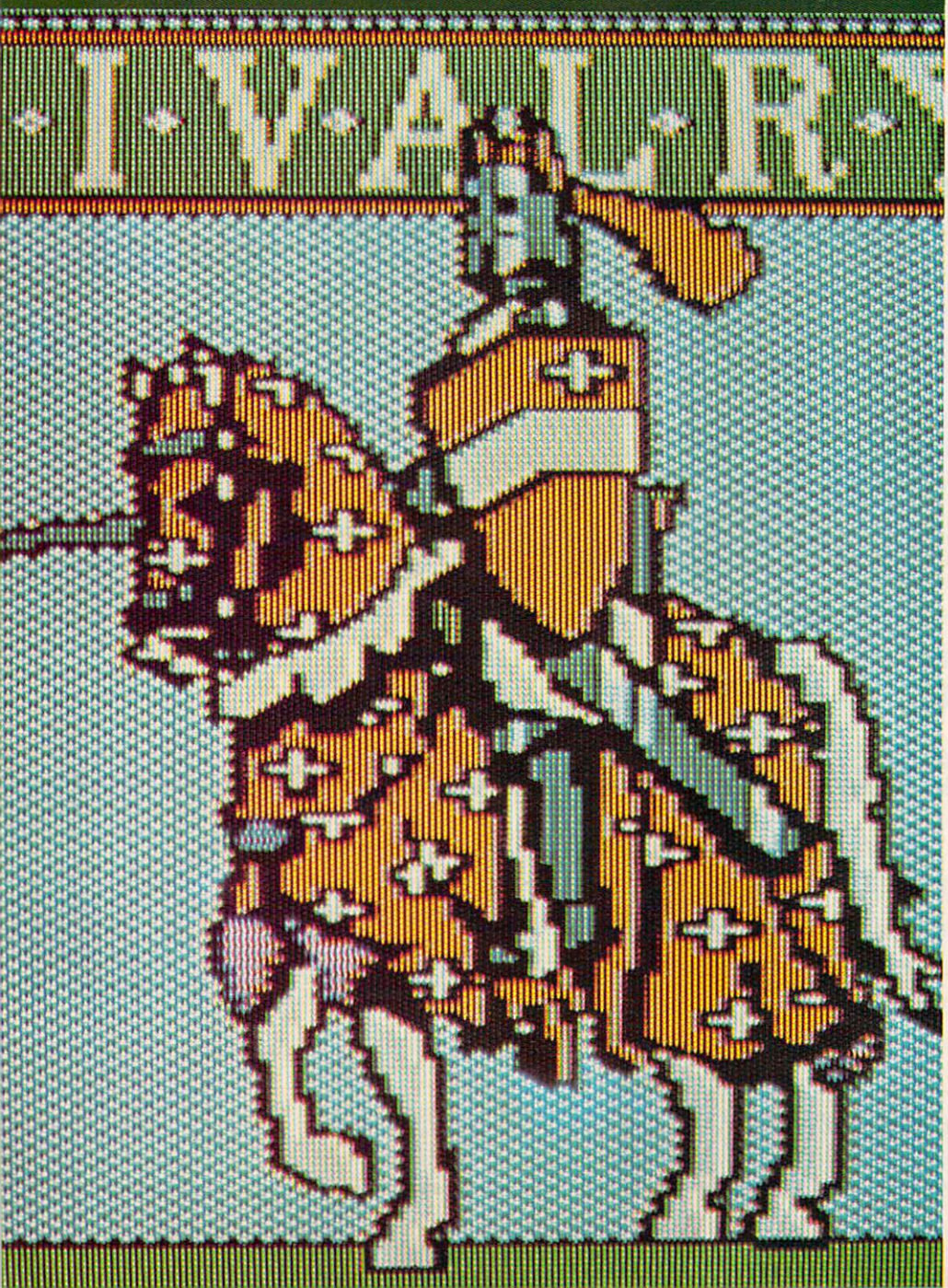
HCM

PROGRAM LISTING

computer gaming T.M.

Name:	Chivalry
Program Type:	Computer-assisted Boardgame
Authors:	Richard Hefter, Janie and Steve Worthington
Machine:	Apple II family
Distributor:	Weekly Reader Family Software Publications 245 Long Hill Road Middletown, CT 06457
Price:	$49.95
System Requirements:	Disk Drive

	Poor	Fair	Good	Excellent
Performance				
Engrossment				
Documentation				



CHIVALRY

A review by Wayne Koberstein

HCM Staff

In the Age of Chivalry, games were serious business—they were *fought* not *played*. Points of honor, of justice, of right and wrong were decided in combat. And over the deadliest contest, a tremendous sense of decorum prevailed: the tradition of Chivalry. In the Age of Computer, we employ games to another purpose: to have fun. Like a modern Connecticut Yankee in King Arthur's Court, many of us prefer our battles on the screen, and our chivalry conveyed with a humorously raised eyebrow.

Weekly Reader (remember them?) Family Software has produced a good rendering of Chivalry as most of us perceive it—mixed with adventure and filtered through a sense of humor. In a very attractive package, with helpful text and nice illustrations, this computer-assisted board game promises to be exciting and fun for the 8 and up crowd; and for the most part, it is. Although really suited to the lower end of the age spectrum, *Chivalry* can provide entertainment to almost anyone, and hours upon hours of enjoyment to the right group. How much you get out of it depends on how you take to this kind of game—a simplified tabletop adventure mastered by your home computer.

Chivalry is most successful at presenting the color and "flavor" of a storybook quest. Action is mostly on the screen, with colorful prompts and short video games. Players keep track of their progress on a matching gameboard. Illustrations on screen and off have the look of some thirties water-colored fairy tale. Even when the game seems to get a little slow, it's still a pleasure to behold. Progress along the board can be fitful, even boring, especially if you're playing alone; but the little arcade-type exercises peppered along the way usually rekindle any lagging interest. These mini-games also sport tasty graphics and some lively animation.

At the Fair Castle you begin a journey fraught with elements of luck, free will, and combat. Your mission: to rescue the King from the clutches of the Black Knight. Well-animated dice throws or wheel spins move

you across the board at a random pace (luck). Occasionally you are offered a choice of doors or paths which can either send you back or push you ahead (free will). And often you enter into on-screen contests that also determine your progress (combat). You may joust, shoot arrows, thread mazes, or (my favorite) fire catapults. More than 20 different contests await you. Joysticks are optional, but using the keyboard may be better to insure positive movement—either up-down or left-right.

Speaking through tapestried text panels, the Gamemaster addresses you as Sir or Lady followed by your previously typed-in name. Beyond these titles, however, the game seems to be non-sexist; Ladys are invited to joust right along with the Knights. After many such battles, you may scale the castle walls to assail the Black Knight. He may defeat you and send you back a few moves, but you can keep coming until he is either broken in combat or gives up in disgust. Do this, and you have freed the King and won his praise.

Chivalry delivers many pleasant surprises and perhaps a few disappointments. Humor plays a prominent role in the adventure and provides much of the fun. Play your darts too well at the Inn and you may upset the patrons: "We did not invite thee in to humiliate us," reads the message," Get thee hence!" And chasing through the Willow Path maze while avoiding the Black Knight's agent can be a crack-up—as well as a challenge to your dexterity with keyboard or paddle. Don't look for much challenge, however, from the Dragon; he apparently does not come out of his lair. I was disappointed to see only his eyes shining in the dark entrance, followed by a message informing me I had been burnt to a crisp. Perhaps a few flames would have been nice?

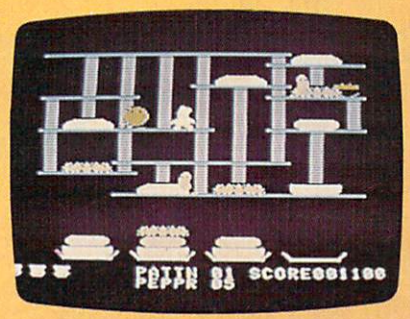
All quibbles aside, *Chivalry* has a lot of class. It is a bright and cheerful evocation of what was sometimes a dark age. Just the artwork in the package itself is engaging. So are the mood-setting madrigals that accompany each stage of the adventure; these simple, spare tunes work well with the Apple's limited sound.

Don't be surprised if you get so caught up in the game on the screen that you forget to even look at the gameboard! You will, however, still need it as a map and a record of your progress. Far from being a distraction, the board provides a common playing field and needed information while saving space on the disk for active game features. It is also a nice piece of artwork in itself, reflecting the above-average screen visuals for the graphically difficult Apple.

Although more expensive than a typical family boardgame, *Chivalry* provides so much more: humor, combat, a computer gamemaster, and a lively show for the money. The Connecticut Yankees at Weekly Reader have done well—adding just enough of the old dark medieval spirit as spice, and transforming Chivalry from decorous battle into a higher order of fun.

HCM

HCM Review



Name:	Burgertime
Program Type:	Arcade game
Machine:	TI-99/4A, IBM PC, Apple IIe
Distributor:	For 99/4A: Texas Instruments P.O.Box 10508 Lubbock, TX 79402 For IBM PC and Apple IIe: Intellivision 5150 Rosecrans Hawthorne, CA 90250
Price:	$29.95 TI-99/4A Cartridge $24.95 IBM PC Disk $24.95 Apple IIe Disk
	Poor Fair Good Excellent
Performance:	
Engrossment:	
Documentation:	

Burgertime

A review by Steve Nelson

HCM Staff

I am a junkfood junkie and proud of it, but I had never realized just how dangerous it was for fast-food restaurant employees to suport my habit until I played *Burgertime*: The game's hero must make my munchies while trying to avoid being munched in the process.

In *Burgertime* you control the hero, Chef Peter Pepper, a tireless maker of hamburgers beset by villainous rogues in his own kitchen. These villains are known by very innocuous names, Mr. Hot Dog, Mr. Pickle, and Mr. Egg—who at first glance, would seem to belong in any kitchen. But watch out—their single-minded intent is to prevent you from receiving your order of hamburgers. In order to do this, they attempt to dispatch Chef Peter.

In the two implementations of *Burgertime* we examined, there were some differences. (We were unable to get the Apple IIe version for review). The TI-99/4A version has six different kitchens (screens); each screen becomes progressively harder. *Burgertime* on the IBM PC is quite a bit more sophisticated. There are nine different levels of play, four starting speeds, and twelve kitchens. You also have a two-player option—something not available on the TI-99/4A.

The game begins in the first kitchen with four unfinished hamburgers. You must negotiate several tiers connected by ladders to reach the various portions of the giant burgers; the meat is on one level, the lettuce on another, and the top buns on still another. Racing around the levels, climbing ladders to get to the layers of the burgers, and avoiding the villains in the process is easier said than done.

A Dash of Spice

The game's creators, however, don't leave you totally defenseless. If you get in a tight spot, you can have Chef Peter throw pepper at the bad guys—stunning them momentarily to allow you to escape. And if you're fast and clever, you can lure Mr. Hot Dog and his cronies underneath a layer of one of the hamburgers and make it fall to the next level, effectively squashing them.

Once you have prepared all the burgers, you advance to the next level and begin again. Each of the different kitchens are progressively harder to negotiate. There are less ladders, and Mr. Hot Dog and his cronies move a little faster in their efforts to stop Chef Peter from completing his order.

As you become more skilled at the game, you can try building up points by squashing and dropping as many of the bad guys as possible, and by collecting bonus points for picking up ice cream cones and cups of coffee which appear periodically throughout the game.

Burgertime's graphics are excellent on both versions I tested. The IBM PC, however, was noticably sharper in detail. In both versions, movement of characters is very smooth—almost as good as a game in a coin arcade. You can play with a joystick or the keyboard. I found that I could control Chef Peter more effectively with the keyboard, but you may prefer using a joystick. The game responds very well to your input, moves along quite fast, and the instructions that come with both versions are excellent.

While the IBM PC's graphics are superior to the 99/4A's, it loses out in the sound effects department. Both versions of the game have a catchy tune that apparently is the *Burgertime* theme but the 99/4A's version lets you hear it repeatedly—suggesting a carnival atmosphere. Also, because of the special sound chip in the 99/4A, the sound effects of the TI version are far richer—adding much more to the excitement of play.

Burgertime is fast-paced, challenging, and fun. Other than adding more sound effects to the IBM PC version, my only suggestion would be to change the program so that Chef Peter has to be careful not to fall off the edges of the tiers (as if he doesn't have enough trouble filling his orders already). If you've ever played Donkey Kong or Mario Brothers, you will enjoy playing *Burgertime* as well.

HCM

Robochase

by Greg Vaughan
and the HCM Staff

While on a routine patrol in Sector 47, you face sudden attack. A tractor beam grabs hold of your spaceship, and pulls you into the lair of a mad scientist. Without ceremony—other than to point out that you didn't heed the "NO SOLICITING" satellite—he teleports you to his asteroid summer home and metes out a death sentence for your intrusion upon his afternoon nap.

His method of execution is bizarre. You are thrown into a room full of deadly robots and electrified barriers. The robots begin to move straight toward you. Your only hope is to maneuver the robots into the electrified barriers and into each other. If surrounded, you can use your teleportation belt to escape—but use it only as a last resort, because with only one charge currently in your possession, it *is* a *last* resort.

The diamond-shaped recharger packs located on the ground are really your only hope, so you'll have to round them up. If you destroy all of the robots in the room, and even manage to accumulate several extra teleport charges, you will be teleported into another room with another ten robots. This time there will be fewer barriers and recharger packs. And to make things worse, half of the charges in your teleport belt will be gone!

The Game

Robochase is a fight to the death in a futuristic prison. You can move up, down, left, or right by means of either the Q, A, O, and P keys or the joystick. If you're using the keys, you can't move diagonally, but if you're using an eight-way joystick, you can. You can teleport by using either the space bar or the fire button on the joystick.

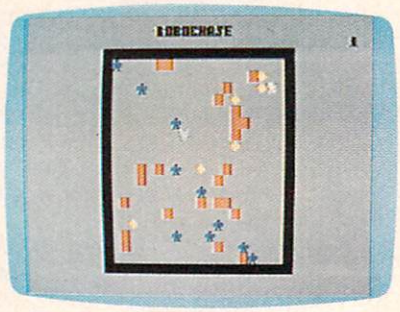
Each level of play opens with ten robots. The robots move directly towards you either horizontally, vertically, or diagonally. (The chance that they will move diagonally increases as you go to higher levels.) If the robots run into barriers, they are destroyed. If they run into each other, both robots are destroyed and a junk pile is formed. The junk pile then acts as a new barrier.

In the upper right-hand corner of the screen, the tally of your teleports is displayed. You start the game with one. To gain additional teleports, you run across the white diamonds on the screen. Every time you go up a level, the number of teleports you have left is divided by two, unless you have only one—in which case, you'll be allowed to keep it.

You can lose in a number of different ways: You can be captured by a robot, run into a junk pile or barrier, or bash into one of the outside walls. If all the robots on a level die, your score is displayed, and you will go on to the next level.

Starting on the fourth level, Spunky the Martian will appear. Spunky moves in a random pattern and will destroy barriers, robots, and teleport rechargers. If you eliminate Spunky, you get a bonus teleport charge. To eliminate Spunky you need to catch him, which isn't easy because Spunky is constantly and randomly moving around.

Scoring for the game is as follows: You get 25 points for each robot killed, 75 points for each teleport charge collected, and 100 points for eliminating Spunky. You also get 125 points for conquering a level, along with 10 points multiplied by a factor equal to that level. You lose 50 points each time you teleport.



When you're moving game characters around on the video screen, it's often convenient to consider character position in terms of *x* and *y* coordinates rather than as an absolute memory address. This is especially helpful when you're trying to establish the positions of two characters relative to one another. Consider an example where one character is at address 1048 and another is at address 1088. If you look at only the difference in their absolute addresses, it might appear that the characters are a long way from each other when in reality, the character at address 1088 is in the character position directly below the character at address 1048.

In the C-64 version of *Robochase*, the positions of each character on the screen are maintained as *x* and *y* coordinates. The *x* coordinate indicates a column on the screen, and the *y* coordinate designates a row position. The room is 18 rows long and 20 columns wide. To ascertain relative positions of characters with respect to each other, all you have to do is compare their *x* and *y* coordinates. If you want to move one character towards another, it is a simple matter to compare the row and column locations of the characters and then either increment or decrement the row or column of the character you want to move in the desired direction. The robots move towards the player in this same manner.

Deriving a Row and Column

In order to derive initial values for the row and the column, you must first decide on the boundaries of your "arena" (how many rows and columns). In *Robochase* the inside of the room is set at 18 rows and 20 columns. These then, will be the maximum numbers of the row and column positions of all the characters. The position in the upper left-hand corner of the room will be the beginning position of an array and will have a row value of 1 and a column value of 1.

We will keep track of each character by storing its row and column position in a two-dimension array. As an example, consider the two-dimension array that stores the position for each of the robots in *Robochase*. It is DIMensioned BT(10,2) in line 190 of the program. There will be 10 robots (dimension one), and each will have a row and column (dimension two), that will indicate its position in the room.

Robochase *for the TI-99/4A appeared in the November 1983 issue of* 99'er Home Computer Magazine.

Absolute Address

When the upper left-hand corner of the "arena" has a row value of 1 and a column value of 1, the absolute address of the corresponding position is 1194. This address is calculated using the following formula:

ADDRESS = (ROW*40) + COLUMN + 1153

The value 1153 is dependent upon the size of your room and where you want to position it on the video screen.

As a character moves down on the screen in the same column, the row value increases by a factor of 1 and the column value remains the same. If, from inside a FOR__NEXT loop, we POKE an address derived from an incrementing row value (leaving the column value the same), the character will move vertically down the screen. If we increment the column value instead of the row value, the character will advance across the screen in a horizontal direction.

ROBOCHASE (C-64)
Explanation of the Program

Line Nos.	
100-170	Header.
180	Initializes game arrays.
190	Sets top of BASIC memory.
200-250	Transfers standard character definitions to RAM.
260-420	Loads game character definitions into RAM.
430-490	Displays title screen.
500-510	Gets input in response to prompt from keyboard.
520-940	Prints game instructions on screen.
950-960	Clears and initializes C-64 sound chip.
970	Clears keyboard buffer.
980	Prompts for use of joysticks.
990	Initializes game variables.
1000-1060	Primary program loop.
1070-1110	Secondary program loop.
1120-1630	Subroutine to move player on screen.
1640-1670	Subroutine to place barriers on screen.
1680-1710	Subroutine to place diamonds on screen.
1720-1760	Subroutine to place robots on screen.
1770-1790	Subroutine to place man (player) on screen.
1800-1820	Subroutine to place Spunky on screen.
1830-1960	Subroutine to get directions from either keyboard or joystick.
1970-2070	Subroutine that enables header screen.
2080-2090	Subroutine that randomly selects positions on game screen for all the character placement subroutines.
2100-2620	Subroutine to move the robots.
2630-2760	Subroutine to move Spunky.
2770-2820	Subroutine for sound.
2830-2880	End of Game routine.
2890-2940	Subroutine to display game screen.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

In translating *Robochase* for the Apple from the TI version, we were faced with several problems: The TI-99/4A has both its own sound chip and the CALL SOUND utility (available in TI-BASIC), whereas all sound on the Apple must be created by "plucking" the Apple's speaker. Each time you PEEK memory location -16336, the speaker moves. If you pluck the speaker fast enough, a musical tone is produced.

The most dynamic sounds on the Apple are created with machine language programs because Applesoft BASIC is too slow to PEEK enough times per second to create a wide variety of tones. We create our sounds for *Robochase* by POKEing in the machine language routine (lines 1530 through 1590) at the beginning of the program and then CALLing the routine when we want to play a note. CALL 866 causes a jump to the machine language program.It is nested in the subroutine of lines 1000 and 1010.

To use this routine, we place a value for the pitch in the variable PT and a value for the duration of the note in LN. These values must be between 0 and 255. The lower the value placed in these variables, the lower the note and the shorter the duration.

Another interesting problem arises from using high resolution (hi-res) graphics. In hi-res mode, the Apple screen is accessed pixel-by-pixel, whereas TI BASIC uses a character-oriented graphics system. To adapt the program to the Apple, we designated all the shapes to be drawn to begin in the upper-left corner. That way all the shapes can be referenced to their positions on the screen from a common starting place.

A two-dimensional array called SL%(,) keeps track of Screen Locations. The first member gives the column or *x* position, the second gives the row or *y* position. This integer array allows us to keep track of the screen "character-by-character" because we defined all the shapes to be approximately the same size as a printed character (8 pixels by 8 pixels). The Draw Object subroutine (lines 950-960) converts these "character" locations into pixel locations and then uses the Applesoft DRAW command to place the objects (robots, diamonds, etc.) on the screen. By setting up the game in this manner, we found it easier to translate all the detail of hi-res graphics with the easy access of character-oriented display techniques.

Also, when the Apple hi-res screen is used with a program of any length or with large arrays, the location where the Applesoft BASIC program is loaded must be changed, or the hi-res screen memory area will be overwritten by the array

ROBOCHASE

variables. To assure that the hi-res screen is protected, we PEEK the start of BASIC location in line 170 (page zero locations 103 and 104). If they are not above the hi-res screen, we POKE the proper values and RUN *Robochase* again. This causes the computer to reload the program at the new start of BASIC location and thus protect the hi-res screen.

ROBOCHASE (Apple II Family)
Explanation of the Program

Line Nos.	
100-160	Program header.
170	Protect hi-res screen.
180-190	Branch to subroutines, POKE shape table, and sound routines.
200-220	Clear screen, display title screen, and find out if joysticks or keyboard is used.
230-260	Initialize for new game.
270-290	Branch to routines to draw first screen.
300-480	Get player's input for move.
490-600	Move person and branch if move makes it necessary.
610-830	Move robots and branch if person is caught.
840-940	Move Spunky if he is in the game.
950-960	Draw object subroutine.
970-990	Read keyboard subroutine.
1000-1010	Make noise subroutine.
1020-1040	Title screen subroutine.
1050-1110	End game and play again option.
1120-1130	Get random number subroutine.
1140-1150	Delete person subroutine.
1160-1520	Subroutine to draw first screen.
1530-1590	Poke speaker machine language subroutine.
1600-1690	Poke shape table subroutine.
1700-1720	Person electrified subroutine.
1730-1780	Teleport person.
1790-1830	Person captured by robot.
1840-1930	Initialize next level subroutine.
1940	Draw barrier subroutine.
1950-1960	Update score, etc. subroutine.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

The rules of play for the PCjr version of *Robochase* are the same as for the other versions of this game. The biggest difference between the PCjr version of *Robochase* and the others is in the way its graphics and sound effects are created.

All of the shapes on the screen are created with the DRAW command. Only the perimeter barrier is created with the LINE command. The shapes for all of these figures are defined in subcommand strings in lines 370 to 410. Each of these strings contain subcommands that instruct the computer to draw the specific shape. These subcommands are extremely easy to learn and use. Simple subcommands like U for UP and L for LEFT make drawing shapes as easy as connecting the dots with a pencil. Other subcommands let you draw diagonally, move without drawing, or draw a line and then return to the spot the line started from. By adding a number after many of these subcommands, you can repeat the subcommand the specified number of times. For example, D8 would draw a line Down 8 pixels. (The pixel is the smallest dot the computer can work with on the screen.)

You can rotate the angle of a shape in two different ways. The A subcommand lets you set an angle of 0, 90, 180, or 270 degrees. This subcommand simplifies the problem of creating four different shapes to move in four directions. For examle: DRAW "A2" will cause all subsequent DRAW statements to draw their shapes at a 180 degree rotation from normal. The TA subcommand allows you more accuracy in selecting the angle of the shape. TA can be set to any value between -360 and +360. This gives the object a full 360-degree accuracy. You should keep in mind, however, that on small objects you may not be able to see the difference in just a few degrees of rotation because of the pixel resolution. In the statement: DRAW "TA 180" the shapes draw

after this statement will be drawn with a rotation of 180 degrees from normal.

The S subcommand lets you set the scale of the shape. The default size of the shape is 4. If you reduce the scale size to 1, you'll make the shape 1/4 of its original size. The maximum scale factor is 255, which will make the shape approximately 64 times larger than its original size. A scale of 8 will make the shape twice its normal size.

You can give your shapes color by placing the C subcommand, followed by the number of the color you've selected, in a string. You can paint the object you draw with color using the P subcommand, which saves you considerable time when you're drawing large objects. Use this subcommand cautiously, however: You must first establish a border to contain the filling, or it may spill out and fill the whole screen. Think of your shape as a bucket that you want to fill with water. If there are any holes in the bucket, the water will spill out.

The PLAY and SOUND commands create all of the sound effects in the program. The SOUND command lets you select a frequency, volume, and duration for a tone to be played. You can also optionally select one of the three voices in the computer to play the note.

The PLAY command is a little more complicated but offers up a whole new world in which to create computer music. Like the DRAW command, it lets you set up a subcommand string which is then executed. With the subcommand string for the PLAY command, you can select a note by its name (CDEFGAB), select the octave in which the note is to be played, adjust the tempo and length of the note, and set the volume.

The music (or simply a series of notes from the PLAY command) can play in either the background or foreground mode. In foreground mode, the program will not continue until the music has finished playing. In background mode, the program will continue executing while the music plays. You can select an *interrupt* to automatically check the music buffer and branch to a routine to fill the buffer when it gets low. If you do this, the continuous play of music in the background becomes transparent to the application program that is running.

ROBOCHASE (IBM PCjr) Explanation of the Program

Line Nos.	
100-180	Program header.
190-260	Initialize the program and define functions.
270-300	Display the title screen.
310-320	Input option to use the joysticks or the keyboard.
330-610	Inritialize a new game, display the playing screen, and set up the key branches.
620	Display the score and number of remaining transporter charges. Branch to appropriate input routine.
630-650	Input routine for the keyboard option.
660-710	Input routine for the joysticks option.
720-740	Move the player and branch to collision routine if a collision is detected.
750-830	Move Spunky and check for captures.
840-930	Move the ten robots.
940-1000	Subroutines to handle robot captures and collisions.
1010-1020	Player is captured by a robot.
1030-1050	Option to play again.
1060-1150	Subroutines to handle collisions between the player and other obstacles.
1160	Move player up to the next level.
1170-1190	Transporter subroutine.
1200-1230	Subroutines for the ON KEY GOSUB command. These subroutines are selected in lines 550 to 580. During the program, branches to these routines are interrupt-driven so that pressing a key at any time will always be detected.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

HCM

SEND $2.00 for our current catalog of TI and 3rd PARTY CATALOG FOR THE 99/4A Home Computer. We include a $5.00 certificate for use with your first order!!

WHAT NOW?

The one to start with is the one to stay with.

Texas Instruments' Home Computer.

32K Memory Expansion Card
RS232 Interface Card

SPECIAL INTRODUCTORY OFFER:
ORDER BOTH CARDS TOGETHER AND DEDUCT $10.00.....

CORCOMP is a former TI-99/4A supplier and its Model Masters Division is the developer of the TI Mini-Writer software and soon to be released action game "Intercept" on module.

TEX-COMP™

TI USERS SUPPLY

TEX-COMP is pround to announce that it has been appointed an authorized dealer by CORCOMP the leading manufacturer of TI compatible peripherals and accessories. As reported in InfoWorld and the Los Angeles Times, CORCOMP is now marketing a fully compatible 32K memory expansion card and RS232 card with a disk controller card to follow soon.

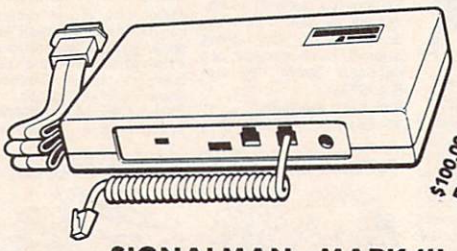
TEX-COMP is now taking orders on the 32K card and RS232 card for immediate delivery from stock. Both cards meet or exceed TI performance and quality specifications and carry a 12 month limited factory warranty.

32K Memory Expansion Card.....$119.95

RS232 Card.....$79.95

Visa and MasterCard
Order Direct
(818) 366-6631

Send Check or Money Order to: **TEX-COMP • P.O.BOX 33084 • GRANADA HILLS, CA. 91344 •**



$100.00 SPECIAL $100.00 FREE Source/TEXNET with purchase of Signalman MK III

Comes complete with cable and cord. This is a unit specially designed for the TI-99/4A and should not be confused with other Anchor/Signalman models that sell for less and/or are not compatible with the TI-99/4A.

SPECIAL: WE HAVE A LIMITED SUPPLY OF TI TERMINAL EMULATOR II WHICH CAN BE ORDERED WITH THE MODEM FOR $24.95. (one per modem please)

ACCESSORY: 9VOLT BATTERY ELIMINATOR FOR MARK III.....$10.95

SIGNALMAN™ MARK III
TI 99/4A COMPATABLE
MODEM

Finally, a low cost, direct connect, high quality and super reliable TI-99/4 and 99/4A compatable modem that comes complete and ready to use — just plug it into a TI RS/232 expansion card.

Mfg. Suggested List Price $139.00

TEX-COMP PRICE
Limited-Time Introductory Offer
$94.95
Including shipping & insurance

SUPER JOYSTICK II

The Only JOYSTICK that can control THE MUNCHMAN!

Available exclusively from Tex-Comp

Rated #1 August 1982 Creative Computing

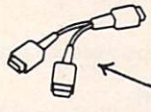
Improved version

The case has been redesigned for easy hand-held operation. The solid steel, short-throw control shaft with an arcade-style ball on the top offers precise, 8-position action. The "soft-touch" fire button is located at the top end of the molded base allowing either right or left-hand play. This improved fire button location is designed to take advantage of the greater dexterity of the index finger resulting in quicker response, time and less fatigue during extended play

SAME ACTION AS THE REAL ARCADE GAMES

•SWITCHABLE GATE-PLATE™. Setting the switchable gate-plate indicator to the 4-way position allows only horizontal or vertical motion dramatically improving response for maze-type games

Helps you make higher levels of Munchman . . . if you're fast enough



NEW PRICE 29.95

Free Shipping*
Second Unit Only $24.95
*Continental U.S.

With FREE Port Expander

TEX-COMP™

TI USERS SUPPLY COMPANY

Send Check or Money Order to:

• P.O.BOX 33084 • GRANADA HILLS, CA. 91344 •

Authorized Dealer

Credit Cards add 3%

[818] 366-6631

MasterCard VISA

TERMS: All prices F.O.B. Los Angeles. For fastest service use cashiers check or money order. Personal checks take at least ten days to clear. Add 3% shipping and handling ($3.00 minimum). East of Mississippi 4½%. (Free shipping on all software orders over $100 and on postpaid specials). Add 6½% S.T. for delivery in Calif. Prices and availability subject to change without notice. We reserve the right to limit quantities. *this ad supercedes all previous ads.*

Cyber-Cipher

by Oris B. Davis
and the HCM Staff

You are sitting quietly in your office at CIA headquarters playing with your newest toy—a Texas Instruments 99/8. Suddenly, your supervisor rushes in and hands you an envelope marked TOP SECRET—EYES ONLY. Locking your door and closing the blinds, you open the envelope.

The note inside informs you that the department's super computer has locked up, and it will become permanently jammed if too many attempts are made to break its access code—which no one can remember. You know the code is made up of four different color choices. Your mission—should you decide to take it—is to decipher the computer's color-coded password, and save the government from wasting 32 million dollars on its latest toy.

The Program

The program begins by asking you to select the level of difficulty. The difficulty levels determine how much help you will get from the computer, and the number of guesses you will be allowed to discover the password. The difficulty levels are explained below.

After entering your difficulty preference, the playing screen is displayed. On the left side of the screen are all of the possible color choices, and the appropriate key to press for each one:Blue = B, Lime = L, Yellow = Y, Red = R, Magenta = M, Green = G. Below this is the low score and a message line. The final code will be displayed on the top right side of the screen.

It is under this display where you will attempt to figure out the password by making guesses and interpreting clues. Make your guesses by pressing four keys to indicate your color selections. As you press each key in the first three levels of difficulty, you will see its corresponding color displayed on the right side of the screen. On levels 1, 3, and 4, you will also get an audible response, the tone of which changes depending on the color.

After you have made your selections, the computer checks your guess. For every color that is correct, but not in the right place, you get a white dot. For every color that is in the right place, you will get a black dot. For example, if the

password is:	and your guess is:
Blue	Blue
Red	Lime
Lime	Red
Blue	Green

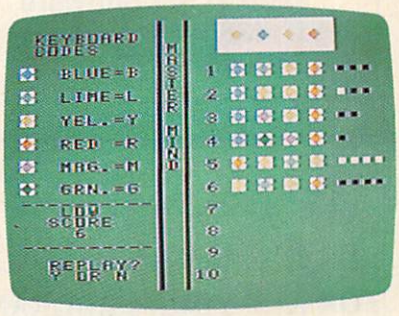
you would receive two white dots, and a black dot. The two white dots represent the center two colors, Lime and Red. The colors are in the password, but you don't have them in the right place yet. Your guess for blue was correct, so you would get one black dot. Green would not garner any dots because it isn't even in the password. Notice that colors can be repeated more than once—it's possible for the password to be all one color.

Levels of Difficulty

Level 1) This is the easiest level. You are given all of the clues at the computer's disposal, which includes displaying all of your guesses on the screen, giving you an audible feedback of the color, and providing you with the black and white dot clues. You are allowed up to ten guesses at the password.

Level 2) At this level you lose the audible feedback from each color and you only have nine guesses to solve the password.

Level 3) It's at this level that you must really put your brain to the test. After you make your selection and your inputs are displayed with the audible tone, your input is erased from the screen, forcing you to remember what your previous choices were. At this level, you only have eight guesses.



Level 4) Don't attempt this level right away. Learn the lower levels first, and become accustomed to the audible tones. On this level, the tones are your only feedback, so if you want to succeed, you must remember all of your selections. Here, you have only seven guesses before the CIA's Super Computer locks up forever.

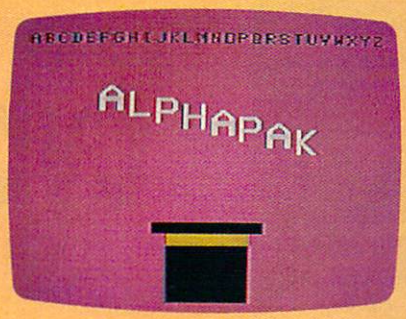
CYBER-CIPHER (TI-99/4A)
Explanation of the Program

Line Nos.	
100-170	Program header.
180-390	Initialize program graphics and variables.
400-540	Display the title screen and input the level of difficulty.
550-1130	Display the playing screen.
1140-1570	Input player's responses.
1580-1890	Check to see if the color is right.
1900-2250	Check to see if both color and location are correct.
2260-2400	Player loses a game routine.
2410-2600	Player wins a game routine.
2610-2640	Display the correct answer.
2650-2760	Option to play again.
2770-2780	Halt the program.
2790-2840	Time delay routines.
2850-2880	Routine to display strings vertically on the screen.
2890-2900	Screen data for display.

For the Key-In listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

HCM

HCM Review



Name: Alpha-Pak
Program Type: Educational Game
Machine: TI-99/4A
Distributor: Smith-Ware
6841 SE Ramona
Portland, OR 97206
Price: $9.95, cassette, diskette
System Requirements:
Disk Drive or Cassette Player, Extended Basic.

	Poor	Fair	Good	Excellent
Performance:			▬	
Engrossment:				▬
Documentation:			▬	

ALPHA-PAK

A review by Wayne Koberstein

HCM Staff

Some of the simplest ideas are also the best. And what could be simpler than teaching a child the alphabet? "Easier said than done," you say, but *Alpha-Pak*, a new game by Smith-Ware, brings some variety to the letter-learning process without making it complicated.

Once the program is loaded, colorful letters fly out of a magician's hat and form the alphabet in a line above it, as a pleasant contrapuntal tune announces the title screen. *Alpha-Pak* contains five short "games," each with a different emphasis. Their titles describe their contents: Letter-Flash, Letter-Choice, ABC Countdown, Snail-Trail, and Alphabet Hop.

The first two look the same on screen. One letter at a time is displayed inside of a colored, lattice-like frame. Each time the letter changes, so does the frame color and—with an attached TI Speech Synthesizer—a voice speaks the letter. Letter-Flash automatically moves through the alphabet as you press the space bar for each succeeding letter. Letter-Choice displays individual letters as their corresponding keys are pressed.

Game 3, ABC Countdown, is an example of *Alpha-Pak*'s satisfying simplicity: By inserting the correct letters into gaps in the alphabet, the child keeps a fuse burning across the bottom of the screen. When the sequence is complete, a rocket climbs to a starlit sky and explodes in colorful fireworks.

In Snail Trail, punching the correct key will hoist a letter out of the snail's path for a little bird to carry away.

Nothing special happens at the end of Alphabet Hop—but it is still engaging. The object of this game is to choose the proper letter in the alphabetical sequence out of a group of four. Little ones should at least enjoy making the frog hop to the lily pad on which sits the correct letter.

A child can easily move between games by pressing 1 and returning to the menu. Once there, another number will call up another game. Of course, someone just learning the alphabet will not be able to read the menu. Perhaps a cartoon symbol representing each menu selection would be helpful—although after playing all the games a few times with an older person, a youngster will probably pick up which number calls which game. If not, there is always the element of surprise . . .

Wise to the ways of impatient-but-energetic youngsters, *Alpha-Pak* keeps the ball rolling as fast as they want it to go. It has the kind of computer responses that pre-schoolers enjoy—press the bar or a key and something interesting happens: a fuse burns on to a waiting rocket; a voice speaks a new letter or says "oh, oh" to mark a mistake; a frog jumps. This is all simple stuff—but it works.

Alpha-Pak's five mini-games do not represent a wide range of difficulty. They do each emphasize different aspects of learning the alphabet—from recognizing specific letters (and their place on the keyboard), to memorizing the proper alphabetic sequence (viewed both as a whole and in isolated groupings). Generally, the first two games introduce the alphabet and prepare the learner for the next three games, which test for memorization. If a mistake is made in this last group of games, the program allows repeated attempts until the right answer is given. If progress continues without error, the voice offers encouragement by saying: "You are doing very well."

With five different fun-spirited games to choose from, it's doubtful that a youngster will get bored with *Alpha-Pak* before learning something.

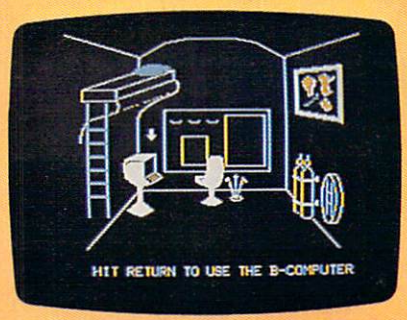
Having enjoyed *Alpha-Pak* myself (even though I do know my alphabet), I can't presume to suggest much in the way of improvements. Usually, in reviewing a game, I run into something that begs changing; but in this case, I am happy with *Alpha-Pak* as it is. Its graphics could be fancier—but that would only distract from the game's purpose, which is not to present an arcade tour-de-force, but to involve children in learning their ABC's.

I strongly recommend using the TI Speech Synthesizer with this program. Humorous and even somewhat human-sounding, the voice greatly enhances enjoyment and involvement in the game.

Alpha-Pak offers no elaborate packaging or extra visual aids. (Instructions consist of two computer print-out pages.) Nor is there anything ostentatious about the price. However, the program ranks with, and even surpasses, more expensive software designed for the same purpose.

HCM

HCM Review



Name: In Search of the Most Amazing Thing
Program Type: Adventure game
Machine: Apple IIe, Commodore 64, IBM PC, PCjr
Distributor: Spinnaker Software
215 First Street
Cambridge, MA 02142
Price: $39.95 Disk
System Requirements:
Joysticks needed on Commodore 64

	Poor	Fair	Good	Excellent
Performance:				
Engrossment:				
Documentation:				

In Search of the Most Amazing Thing

A review
by Steve Nelson
HCM Staff

If you have been searching for a non-violent, educational game that your children can play—stimulating more than just their hand-eye coordination—you will be pleased to note that Spinnaker Software's new adventure game, *In Search of the Most Amazing Thing*, does just that—and more . . .

In playing the game, you pilot the B-Liner (a combination hot-air balloon, and all-terrain vehicle) around the Darksome Mire (a sticky swamp on the planet Porquatz) searching for clues that will lead you to the Most Amazing Thing (a mysterious object that is hidden somewhere in the Darksome Mire).

Whew, did you get all that? Well there's lots more. You're going to have to ask Uncle Smoke a lot of questions—but don't bug him to much, as he's a sleepy old coot, and he has a tendency to clam up if you wake him too fast. Uncle Smoke's advice comes in real handy out in the Darksome Mire, and it's a very good idea to write down what he has to say so you don't forget it.

You begin the game by obtaining chips (money) to buy items for the B-Liner from the Galactic Store. Uncle Smoke will give you objects that he has collected over the years, and plenty of advice to help you trade for chips with the people of Metallica. Metallicans live deep underground and love to barter chips for the strangest things. It takes a few attempts before you learn how to deal with these people—they are cunning traders. So be careful, and listen closely to what Uncle Smoke has to say. Uncle Smoke will also give you advice and helpful hints on how to deal with the various cultures that live in the Mire.

Your computer's keyboard links you to the B-Liner's control panel. You can fly the B-Liner over the Darksome Mire, or you can drive it. Flying is faster; but the skys above the Mire are very windy, and you must learn how to control the B-Liner if you want the winds to carry you to your destination. They can just as easily blow you way off course. Driving is easier, but much slower—and you must worry about getting food and oil. You must also try to avoid the treacherous Mire Crabs that disguise themselves as Night Rocks and attack you when you get too close.

What are Mire Crabs and Night Rocks, and how do you get food and oil? How do you know where to go in the Darksome Mire? Where do you begin looking for the Most Amazing Thing? Who are these people that live in the Mire? Will they help you? These are just some of the problems you'll have to deal with in your quest.

Spinnaker provides excellent instructions—but there is no substitute for getting out there and experiencing the game itself. It takes time to learn how to deal with the natives (some are helpful and some are not), and to fly the B-Liner.

As you travel through the Darksome Mire and trade with different cultures in search of the Most Amazing Thing, it is nice to know that you can stop at any time, save the game on disk, and return to your same location when you begin again.

Once you find the Most Amazing Thing, the computer automatically relocates it in another secret place—so you can continue to play the game as many times as you wish.

In Search of the Most Amazing Thing may seem slow at first, and difficult to understand—but as you play, the clues from Uncle Smoke and other instructions slowly fall into place, and the game becomes quite a challenge. It may take you several tries before you get the hang of it. But try not to get discouraged—it is worth the effort.

Some aspects of the game, however, seem more like unnecessary handicaps than real challenges. For instance, leaving the B-Liner and flying with the jet-pack is very difficult. Movement is just not very responsive. Also, when dickering with the Metallicans, the process is repetitive, and tends to get

Continued on page 118

Wild Kingdom

by Monte Ulm
and the HCM Staff

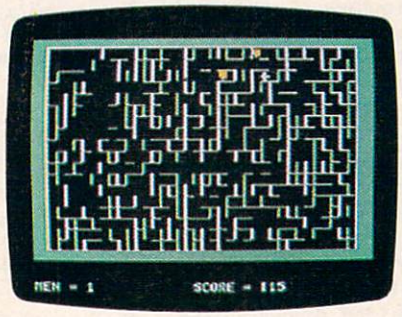
Only the stout of heart need read any farther, for this action game takes you into the most dangerous part of Africa.

You are in charge of an expedition to capture three vicious tigers which have been terrorizing the local villagers. Although you have devised an ingenious system of gates and pens to trap the animals, the only building materials available are the few trees you can find, and the tall grass. If angered, the tigers could easily break through these flimsy barriers.

Speed and cunning are your most valuable resources if you plan to pull off this feat without becoming the main course in the lion's den.

Three tigers are roaming around on the screen—one male, and two females. Your chief goal is to capture the male tiger first, because with this beast in captivity you can control the two females. To do this, you need to keep the male separated from the females, for if the females sense that the male is nearby, they will tear down the walls to get to him. Needless to say, this could be quite disastrous if you happen to be standing nearby when they escape.

The hungry tigers will constantly stalk you, so you must always be on your guard. The male tiger is especially dangerous, for if he senses that you are close to him, he may tear down the walls to get to you. You have to operate very quickly to trap him.



Your one defense is the power to open and close the walls. You have devised an ingenious method which lets you open and close a wall that is *one wall away from you*. You cannot close a wall that is *directly in front of you*. This trick allows you to use yourself as bait, standing on just the other side of a wall, enticing a tiger into a pen with three sides. When the tiger finally enters, you can close the other side—trapping the tiger—and run away.

If you press a key in a direction other than the one you are facing, you simply change directions and do not move—the next time you press a key in that direction you will move.

REQUIRED EQUIPMENT: Extended BASIC
OPTIONAL EQUIPMENT: joysticks

Wild Kingdom was originally written on the TI-99/4A. The game and its rules are the same on all the systems, but the keys used on the keyboard differ slightly.

KEY FUNCTION
E Move hunter up.
S Move hunter left.
D Move hunter right.
X Move hunter down.
Y Open and close gates.

The TI version uses character graphics to create the maze. Only four characters are needed to draw it—one is a blank, the second has a wall on the left side, the third has a wall on the top of the character, and the fourth has a wall on both the left side and the top. By randomly placing these characters on the screen, you create the maze.

Don't worry if it seems that entire portions of the maze are closed off to you. You can open and close various walls of the maze by placing the hunter one wall away from the wall you want to change, pointing him in the wall's direction, and pressing Y, or the fire button on the joystick.

To keep from upsetting the walls as the characters move about, we used sprites for the hunter and the three cats. This means that they can be placed anywhere on the screen without affecting the normal character graphics.

***WILD KINGDOM* (TI-99/4A)**
Explanation of the Program

Line Nos.	
100-170	Program header.
180-190	Display the title screen and input an option to use the joysticks.
200-210	Branch to the setup routine for a new game.
220-380	Set up routine for a new game, and repaint the playing screen.
220	Initialize the game variables, and the array that keeps track of the screen contents.
230-250	Define character graphics and color.
260-270	Clear the screen and draw the border.
280-300	Draw the maze on the screen randomly.
310	Define the four characters that are used for the hunter. Create four arrows to indicate the hunter's direction.
320-380	Place the three tigers and the hunter on the screen.
390	Control loop for the entire program.
400-530	Scan the keyboard and move the hunter.
540-610	Control loop to move the two yellow (female) tigers.
620-730	Routine to open and close the gates to the walls when either [ENTER] or the fire button is pressed.
740-790	Control loop to move the black (male) tiger.
800-1070	Logic to determine the direction of movement for the tigers.
1080	Determine the proximity between each of the yellow cats and the black cat.
1090-1160	Routine to make the walls disappear when the cats break them down.
1170-1190	End of game routine. Restart new game.
1200	Physical end of the program.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

REQUIRED EQUIPMENT: C-64
OPTIONAL EQUIPMENT: joysticks

Wild Kingdom on the Commodore 64 is quite similar to the TI version. Both the C-64 and the TI use character graphics to create the maze, and both use sprites for the hunter and the cats. This was not the case for the Apple and the IBM machines.

The trickiest part of the program is opening and closing gates. If you want to open a closed gate, then position your hunter so he faces the closed gate at a distance of one gate away. Then press either U or the fire button on the joystick.

KEY FUNCTION
I Move hunter up.
J Move hunter left.
K Move hunter right.
M Move hunter down.
U Open and close gates.

The Commodore 64 uses its built-in graphics characters to create the maze. The only shape definition in the program is for the hunter and the three tigers. The hunter and all three tigers are placed on the screen as sprites—the program doesn't need to consider whether they will affect the background characters.

WILD KINGDOM (C-64)
Explanation of the Program

Line Nos.	
100-180	Program header.
190	Branch to initialize the system and display the title screen.
200-290	Sound routines.
300-380	Calculate addresses.
390-450	Joystick routine.
460-490	Key input routine.
500-550	Change hunter's shape if direction has changed.
560-660	Move the hunter.
670-810	Control loop to move the tigers.
820-1300	Routine to figure out where to move the cats.
1310	Subroutine to print the score.
1320-1350	Check to see if the cats have eaten the hunter.
1360-1650	Routines for the hunter opening and closing the gates.
1660-1680	Main control loop for the entire program.
1690-1800	Routine to handle the hunter's death.
1810-1970	Restart the game.
1980-2210	Initialize the program variables, display the title screen, and input the joystick option.
2220-2370	Sprite data.
2380-2440	End of game, option to play again.
2450-2540	Check for capture of the tigers and adjust the points.
2550-2570	Subroutine to calculate the screen address for the sprites.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

OPTIONAL EQUIPMENT: joysticks

The Apple version of *Wild Kingdom* operates just like the other versions—but because of differences in the way the Apple works with graphics, the similarities stop there. This version takes advantage of a useful machine language routine to create and manipulate character graphics. The routine, which is completely contained in data statements, allows you to define your own characters and then display them on the graphics screen. This method is used to create the maze and the hunter.

KEY FUNCTION
A Move hunter up.
← Move hunter left.
→ Move hunter right.
Z Move hunter down.
[RETURN] Change gates.

The data for the shapes is contained in data statements following the machine routine. The cats are created with a standard shape table, because the cats need to be two different colors. The machine routine allows only one color. Also, a short machine routine is included to create sounds using the internal speaker.

Character Graphics

The character graphics routine allows you to create your own graphics characters, and place them on the screen as though they were text characters. These characters are much easier to define than the standard Apple shape tables. Each character is seven pixels wide by eight pixels high, and eight bytes of data define each character. The values placed into the data statements represent the bits turned on in each row of the character. A typical row of pixels can be defined like this:

TOP ROW OF 7 PIXELS IN A CHARACTER

1	1	0	0	0	1	1	bit	value	bits on
							6	64	64
							5	32	32
							4	16	0
							3	8	0
							2	4	0
							1	2	2
							0	1	1

Value of byte in data statement: 99

The character graphics routine is fairly easy to use. This routine requires that the character definitions start at address 815 (HEX 032F). However, you could alter the program at address 803, and 804 (HEX 0323 0324). To find the values to place in these addresses, use the following formula:

X = INT(DTADD / 256)
Y = DTADD - X

Place X in address 804, and place Y in address 803. DTADD is the address where you put the character definition table.

Continued on next page

Wild Kingdom

To access the program, you need to use the HTAB and VTAB commands to place the character, just as you would with normal text. Then you need to POKE into location 0 (zero) the character number you wish to display. The first character in the definition table is number 0 (zero), the second character is number 1 (one), and so on.

In this version of *Wild Kingdom*, the values POKEd into location zero start at 96. The last three bits of the value are not used. This means you could make the value any number from 0 to 255, and only the lowest five bits would be used. However, this technique limits your shape definition to 32 characters. (POKEing 0 to 31 would give you the 32 characters; POKEing 32 to 63 would produce the same 32 characters, and so on.)

You can enter the machine language program in two ways: CALL 768 will execute the program and replace the graphic currently on the screen with the entire character area of 7x8. Or, if you use CALL 772, the program will place the character over the top of the existing graphic without disturbing it. It works like an OR operation, and only turns on the pixels that were specified to be turned on; it doesn't affect the others. In this version of *Wild Kingdom*, these two CALL values are placed in variables C1 and C2.

Sound

This program also contains a sound routine in machine language, allowing you to create your own tones on the internal speaker. If you want to incorporate this routine into your own programs, you will want to experiment with different sounds. To generate a sound, POKE a value between 1 and 255 into location 0:

```
POKE 0,16
```

Then use the CALL command to call a machine language program:

```
CALL 899
```

WILD KINGDOM (Apple)
Explanation of the Program

Line Nos.	
100-170	Program header.
180	Check to see if the program has been loaded into high memory. If it has not, reload program into high memory.
190	Display the title screen and input control options.
200	Read the machine code routine from data statements and poke it into memory.
210-330	Set up variables and screen.
210	Set up variables and the shape tables.
220-230	Clear the screen and draw the borders.
240-260	Draw a random maze.
270-330	Place the tigers and the hunter on the screen.
340	Main control loop for the entire game.
350-660	Move and draw the hunter.
670-790	Control and draw the two orange tigers.
800-950	Open and close gates.
960-1010	Control and draw the white tiger.
1020-1340	Logic routine to control the tigers' movements.
1350-1370	Determine distance between the orange and white tigers. Branch to break down barriers.
1380-1470	Routines to break down the barriers.
1480-1520	End of game routine.
1530-1540	Subroutine to draw and erase shapes when the hunter is caught by the tigers.
1550-1580	Data statements containing the machine language routine for character graphics and sound.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

In this program, the variable SOUND was set to 899 so that when you jump to the sound routine, you can use CALL SOUND.

PC jr

REQUIRED EQUIPMENT: PCjr with Cartridge BASIC or PC with color graphics adapter, RGB monitor, and BASICA.

The IBM version (for both PC and PCjr) of *Wild Kindom* uses the LINE statement in four different subroutines for simulating characters on the screen to create the maze. These subroutines are in lines 810 through 840.

Because the IBM machines do not have sprites, it is necessary to draw and erase each character as it moves through the maze. The PUT command is the fastest way to accomplish this. It would have been fine for placing and erasing cats on the screen and for creating the hunter, if the hunter didn't require four different shapes—four different arrays. For this reason, the DRAW command was selected because it has the capability to rotate shapes. The hunter is simply an arrow shape that points in the direction of travel.

The A parameter of the DRAW command is perfect for shape rotation. It allows you to turn the shape in one of four directions with an input from 0 to 3. The flag used to indicate direction is incorporated into the DRAW command, thus four characters were created with the effort of one.

The arrow keys are used on the IBM machines to move the hunter. the space bar opens and closes the walls.

WILD KINGDOM (IBM PC and PCjr)
Explanation of the Program

Line Nos.	
100-190	Program header.
200-260	Initialize program variables, graphics, and functions.
270-320	Display the playing screen.
330-350	Main control loop for the game.
360	Increase score when black cat is caught.
370	Hunter gets caught.
380-390	End of game routine. Option to play again.
400-650	Subroutine to move the cats through the maze.
660-670	Determine if cat breaks down the walls.
680-710	Control loops for hunter's movement.
720-790	Control opening and closing of gates by the hunter.
800-840	Subroutines to place the different parts of the maze on the screen.
850	Clear the hunter or cats' last position from the screen.
860	Draw the hunter in his new position.
870-880	Check to see if the hunter has been eaten yet.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

HCM

Amazing Thing ... from p. 115

somewhat boring. As for the Mire Crabs, once they attack you, it's nap-time (you'll see what I mean.) There really should be something else you could do to avoid them.

The graphics are simple and nicely drawn, although there are not very many different screens. The sound effects are minimal, except for the Musix program, which allows you to create your own songs to help you in communicating with the various cultures of the Darksome Mire.

The various machine versions are virtually identical with one exception: The C-64 version requires joysticks and is very slow to load—traveling underground to barter with the Metallicans seems to take forever.

Included in the package is a complete novel— about Uncle Smoke's adventures with his nephew Terry as they search the Darksome Mire for the Most Amazing Thing—a nice "extra" to an innovative package.

In Search of the Most Amazing Thing is a nice change from typical video and educational games that require you to blow up a space ship or kill off a few thousand aliens in order to feel like a winner. It's a complicated journey with many surprises. Just make sure you've got all the gadgets you need for the B-Liner, and enough chips—because as Uncle Smoke says, "you're gonna need em."

HCM

SPEEDER

by Robert S. Keller
HCM Staff

Here's a little VIC-20 game, just 32 lines of program code, that easily illustrates some game and graphics principles. You can play it as it is, or flex your programming muscles and make it into a game of your own design. An interval of 20 between-line numbers in the game makes it easy for you to insert additional program lines.

A green *Speeder* ricochets across the screen, using logic and random motion to elude you. Using the keys E, X, D, and S (for up, down, right, and left), you draw a purple barrier that the speeder cannot cross. You win if you can touch the speeder with the end of your line. The program displays the number of moves that you made, tells you if you've set a record for least number of moves, and waits for your keypress to signal it to begin again.

Speeder's Program

(Refer to the numbered sections of *Speeder*'s key-in program listings.)

(1) This section clears the screen and sets the initial value of the variables—thus it's called an *initialization* section.

(2) Reads the keyboard with the GET statement. GET doesn't wait for keyboard input like INPUT does—if nothing is there, it just goes on, which is ideal for the continuous action of games.

(3) Adds to the purple barrier according to section 2's instructions. It uses variables A and B as counters to make sure that you don't go off the screen. The counters work like this: The barrier starts on row 10, so the initial value of B, the "row counter," is 10. Every time the barrier moves up, B decreases by 1, and every time it moves down, it increases by 1. Before each move, the program checks to see if the move will increase B above 23 (the last row) or decrease it below zero (the first row). If either of these cases is true, the move simply isn't made. The column counter, A, works the same way.

The purple barrier (and Speeder in section 7) is POKEd into screen and color memory. Screen memory holds the shape of an object, and color memory holds the color. Commodore's manuals and many third-party books list these memory locations, which begin at 7680 (screen) and 38400 (color). Information headed for the screen is stored here. It is read by a special chip, which sends the content to the television or monitor.

(4) Checks to see if you've caught Speeder. If you have, you go to section 8; if you haven't, to section 5.

(5) Uses counters to keep Speeder from crossing the screen boundaries. X2 and Y2 are the coordinates of its proposed new position, and X1 and Y1 define its old position. If Speeder has reached the edge of the screen, this code randomly chooses either to bounce Speeder off or head back the way it came.

(6) First looks at the screen color one step ahead of Speeder (line 1000). If the color is white, the screen ahead is clear, and execution moves on to section 7, which moves Speeder.

If the color isn't white, the program knows that the way is blocked by the purple barrier, and it randomly chooses to either bounce Speeder off the obstacle or send it back the way it came. Then, instead of going on to actually move Speeder, execution goes back to sections 2-5. The program needs to make sure that the new direction it has chosen doesn't violate the screen border—and if Speeder is trapped between the purple line and the border, then the program needs to read the keyboard and move the purple line.

(7) This section erases the old Speeder by POKEing BL (character 32, a space) into screen memory, and 1, the color white, into color memory. Then Speeder's new position is POKEd in. Screen motion often flows best when you arrange to erase an object's old position just before you plot its new position.

(8) Completes the player's turn by printing the number of moves the player took. Then it tells if a record has been set, and prompts the player for a keypress as a signal to return to section 1.

These variables are used in the program:

R	**Current record for least moves**
Count	**number of moves taken**
A$	**Keyboard input**
XP	**Poke location of Speeder**
XC	**Speeder's shape**
G	**Speeder's color**
C	**Offset to screen color memory**
Delta	**Changes purple line direction**
FLAG	**Makes sure Speeder doesn't escape from the corners of the screen**
P	**Line's color**
AP	**Poke location of purple line**
AC	**Line shape**
A and B	**Purple line vertical and horizontal position**
DX and DY	**Changes speeder's horizontal and vertical position**
X1 and Y1	**Speeder's old position**
X2 and Y2	**Speeder's new position**
BL	**Blanks out old Speeder position**
DP	**Checks to see if Speeder has hit purple line**

Customizing your Speeder

Here are some suggested alterations, arranged in increasing order of difficulty. The framework of the game can be expanded in many directions.

1. Change the game so that the Speeder entirely disappears and the game turns into a drawing program. Jump from the end of section 3 to line 320.
2. Change the color of the line or the Speeder by changing P and G in section 1.
3. How about changing the *shapes* of the Speeder and the purple barrier? (Change the values of XC and AC.)
4. Put a sound routine in section 3 to hear a sound every time the purple barrier is enlarged, section 7 to hear a sound every time Speeder moves, between lines 1020-1040 in section 6 to hear a sound every time Speeder hits the purple barrier, and in lines 820-880 to get a sound every time Speeder hits the screen boundary.
5. Add a title screen, perhaps with a title tune. Place it before section 1.
6. If you change the numbers in lines 580

Continued on next page

Speeder

and 620, you can restrict the purple line so that it can't move out of an area of the screen. If you restrict it to the middle of the screen and leave Speeder free to travel anywhere, the game's player will have to set a crafty trap in order to capture it.

7. How about ending a turn if the player hasn't captured Speeder before it has made a certain number of moves? You could count the moves in line 1140, and if a total was exceeded, you could jump to a new section at the end of the program.

   The number of moves remaining could be continuously printed in the upper left corner of the screen as the game is played. Be sure to include a [HOME] in your print statement, and a semi-colon at the end, or the number of moves remaining will march right down your screen instead of staying up in the corner.
8. What if the purple line couldn't cross itself? You would need to look ahead of the purple line in color memory, just like Speeder does in lines 1000-1060, to make sure no purple lay in your path. To look ahead you will have to determine the direction of the purple line.
9. What if obstacles sprouted in the paths of both the purple line and the Speeder as the game goes on?
10. Put more graphics in section 8!
11. If you want to be a pro, make it possible for the game to store the highest score and the champ's name on disk or on tape. This extension would go in section 8.
12. Add joystick control to the game in section 2. The program could query you and select the appropriate controls in section 1, and then the appropriate section of code could be executed in section 2.

SPEEDER (VIC-20)
Explanation of the Program

Line Nos.	
100-220	Program header.
240-340	Initialization.
360-500	Read keyboard.
520-640	Add purple line.
660-720	Check for game win.
740-900	Check Speeder's screen boundary.
920-1060	Check Speeder's purple line boundary.
1080-1160	Move Speeder.
1180-1340	End game and scoring.

HCM

VIC-20

```
100 REM  **********
120 REM  *  SPEEDER  *
140 REM  **********
160 REM  BY  ROBERT  KELLER
180 REM  HOME  COMPUTER  MAGAZINE
200 REM  VERSION  4.3.1
220 REM  V20  BASIC
240 REM
260 REM  SECTION  ONE
280 REM
300 PRINT"■SHIFT CLR■";:R=2000
320 AP=7888:A=10:B=10:XP=8056:X1=2:Y1=1
    7:C=30720:DX=1:DY=1:RR=-1:RX=-.5:CO
    UNT=0
340 P=6:G=5:BL=32:XC=160:AC=160
360 REM
380 REM  SECTION  TWO
400 REM
420 GETA$:IFA$="E"THENDELTA=-1:GOTO  58
    0
440 IFA$="X"THENDELTA=1:GOTO  580
460 IFA$="D"THENDELTA=1:GOTO  620
480 IFA$="S"THENDELTA=-1:GOTO  620
500 GOTO  800
520 REM
540 REM  SECTION  THREE
560 REM
580 IFB+DELTA>0ANDB+DELTA<24THENAP=AP+D
    ELTA*22:B=B+DELTA:GOTO  640
600 GOTO  720
620 IFA+DELTA>-1ANDA+DELTA<22THENAP=AP+
    DELTA:A=A+DELTA
640 POKEAP,AC:POKEAP+C,P:COUNT=COUNT+1
660 REM
680 REM  SECTION  FOUR
700 REM
720 IFDELTAANDAP=XPTHEN  1240
740 REM
760 REM  SECTION  FIVE
780 REM
800 X2=X1+DX:Y2=Y1+DY
820 IFX2<0THENDX=1:FLAG=-1:IF(INT(RND(0
    )+RX))THENDY=DY*RR
840 IFX2>21THENDX=-1:FLAG=-1:IF(INT(RND
    (0)+RX))THENDY=DY*RR
860 IFY2<0THENDY=1:IF(INT(RND(0)+RX))AN
    DNOTFLAGTHENDX=DX*RR
880 IFY2>22THENDY=-1:IF(INT(RND(0)+RX))
    ANDNOTFLAGTHENDX=DX*RR
900 FLAG=0:X2=X1+DX:Y2=Y1+DY
920 REM
940 REM  SECTION  SIX
960 REM
980 DP=XP+DX+DY*22
1000 IF(PEEK(DP+C)AND15)=1THEN  1140
1020 IF(INT(RND(0)+RX))THENDX=DX*RR
1040 IF(INT(RND(0)+RX))THENDY=DY*RR
1060 GOTO  420
1080 REM
1100 REM  SECTION  SEVEN
1120 REM
1140 POKEXP,BL:POKEXP+C,1:XP=DP
1160 POKEXP,XC:POKEXP+C,G:X1=X2:Y1=Y2:GO
     TO  420
1180 REM
1200 REM  SECTION  EIGHT
1220 REM
1240 IFCOUNT<RTHENPRINT"■HOME■NEW  RECORD
     :";COUNT:R=COUNT:GOTO  1280
1260 PRINT"■HOME■";COUNT;"MOVES"
1280 PRINT"■3CRSRDOWN■  PRESS  ANY  KEY"
     :PRINT"  TO  BEGIN  AGAIN"
1300 FORN=1TO2000:NEXT
1320 GETA$:IFA$=""THEN  1320
1340 PRINT"■SHIFT CLR■";:GOTO  320
```

HCM

THE BOOLEAN BRAIN

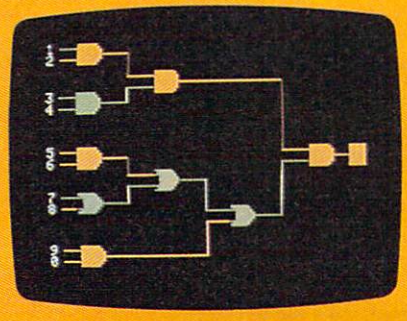
by W.K. Balthrop
HCM Staff

It is late. Another night of blasting aliens draws to a close. You reach to turn the computer off—*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. But 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. Yes, somehow your home computer has actually become your home!

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. Above the door in the center of the screen is one of four letters which indicates the direction you are facing. For example, if you're facing north, the letter N would appear on the wall. To move in any of the four directions simply press either E, W, N or S. You do not need to press [ENTER]. 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.

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.

To open a door, you must satisfy the lock's digital logic. On the left side of the screen are 10 lines—inputs to the five gates. The lines are numbered 1 2 3 4 5 6 7 8 9 0, in the same order as they appear on the keyboard. 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.)

Continued on p.134

HCM Review

Name:	Murder by the Dozen
Program Type:	Deduction Game
Machine:	Apple, Commodore 64, IBM PC and PCjr, Atari
Distributor:	CBS Software 1 Fawcett Place Greenwich, CT. 06836
Price:	$34.95, disk

	Poor	Fair	Good	Excellent
Performance:				
Engrossment:				
Documentation:				

MURDER BY THE DOZEN

A review by Steve Nelson

HCM Staff

While I enjoy saving the world from certain destruction by holding off hordes of various space monsters and demons, I sometimes get "wrist lash" and find myself in the vulnerable position of being unable to use my joystick. Or, even worse, after hours of endless play I find that colorful graphics have been indelibly imprinted on my brain, and even after I turn off the monitor, I see space invaders everywhere I look. Egad! There goes one now! I believe that this phenomenon is not unlike the battle fatigue that soldiers experience after spending too much time in the trenches. Fortunately, there are ways to alleviate these symptoms, and one of the best ways I've found is to play the new game from CBS Software, *Murder By The Dozen*.

The game consists of 12 murders, each with its own sordid scenario. The crime scenes are all somewhere in Micropolis—a city, we're told, populated with friendly people. But in your role as chief homicide detective, you soon discover that some of the people aren't so friendly after all. The investigation begins with a phone call to you at Police Headquarters. You are given the case history and a starting point—it's up to you to determine the motive and name the murderer.

There are 28 locations in the city, each one with seven options that may or may not lead to usable clues. These seven options are displayed on screen while you type in your selection. Choices one through three involve interviewing people, and choices four through six allow you to examine the physical evidence. Use choice seven to change locations and search for more clues.

As you begin your investigation, you will quickly find yourself with several suspects and motives but no closer to solving the case. Now your sleuthing abilities are put to the test.

While the object of this game is to solve the crime, if that was all you had to do you could simply investigate every clue and eventually determine the culprit. Fortunately, that is not the case. To make this game more challenging, its creators have added a time element. From the moment you begin searching for clues, you begin accumulating time on the game clock. At each location you must decide which options will result in usable clues. Each choice leads to several numbers, which must then be looked up in the clue book to determine their meaning. If you interview the wrong person or examine evidence that isn't germane to the case, you have gained nothing and lost valuable time. When you move to a new location, the program adds travel time to the game clock based on the distance between locations. A short trip may add only a few minutes, but traveling across town can take 30 minutes or longer. Keep this in mind when planning your moves.

Once you decide that you have completed your investigation and are certain that you know "who done it," you can check to see if you are correct. If you are right, the game rates your detecting skills based on the time that elapsed during your investigation. *Murder By The Dozen* is designed so that you can play it alone or with up to three other detectives. When more than one player is involved, you are not only racing the clock, but the other detectives as well: The first one to crack the case is the winner.

I found this game to be quite challenging. In fact, I could have been sued for false arrest in the first case I "cracked." I wasn't even close to finding the real murderer.

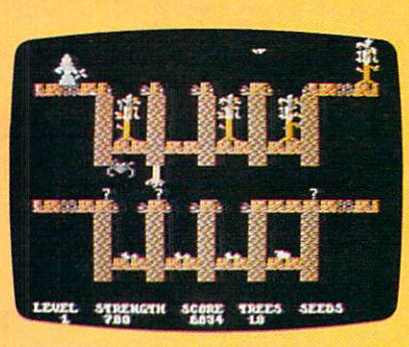
The game package provides you with everything you need to solve the murders. It includes a detective manual, which explains the game's procedures and rules in detail; a book of 700 clues; a set of note pads complete with a map of Micropolis; and, of course, the solutions—cleverly disguised so that you don't inadvertently see the solution to more than one murder at a time.

I played *Murder By The Dozen* on the C-64 and the Apple IIe and the game performed with no problems. On the Commodore, however, there is a noticable time delay in loading and executing the program because the Commodore's disk drive is interfaced through a serial port. This is a minor annoyance though, and once the program is under way, the Commodore handles it fine.

I have only one complaint: there are just 12 cases. Once you solve them all, the game's possibilities are exhausted. Still, I recommend this game to anyone who seeks a change from ordinary computer games and desires one that stimulates the reasoning part of the mind, rather than just the reacting part.

HCM

HCM Review



Name: Necromancer
Program Type: Arcade/Adventure Game
Authors: Bill Williams, Scott and Steve Coleman
Machine: Commodore 64
Distributor: Synapse Software
5221 Central Avenue
Richmond, CA 94804
Price: $34.95, diskette
System Requirements:
Joystick

	Poor	Fair	Good	Excellent
Performance				
Engrossment				
Documentation				

NECROMANCER

A review by Wayne Koberstein

HCM Staff

Necromancer is an evil magician, literally a "mover of the dead," who has overcome Death and now uses it as an ally. His minions include ogres, and zombie spiders. You enter his haven as a druid, a priest of the old white magic, and wield a flashing wisp, which can zap enemies or animate an army of walking trees to crush the spider larvae in their crypts.

Though simple in description, this game is cleverly constructed out of many single elements to form a complicated whole. In the thick of battle, the realm of the dead turns out to be quite lively. Though not a full-blown adventure, Necromancer offers more than simple arcade. To play at all, you will certainly need quick eye-to-hand reflexes; but without a good strategy as well, you may never even get to see the Necromancer, let alone defeat him.

As you progress through the three "Acts" of this dream-play, a status board at the bottom of the screen displays your total points, a separate "strength" figure, and the number of trees that survive various attacks. In Act One you must use the wisp to plant and protect trees on the old site of an enchanted forest, now a barren plain. The more trees you grow, the faster and more viciously they are attacked by spiders' bite and ogres' axe. That's where a quick finger on the joystick comes in handy. However, you must be wise about how you employ this magic wand; some moves may gain you points, but they will cost you strength, which you must retain long enough to protect your trees until they are mature. The remaining Acts Two and Three each have five levels of increasing speed and difficulty. Here you must think and act quickly to survive very long, or more and more of the evil wizard's surprises will sap your strength and scatter your poor druid to the four corners of the screen.

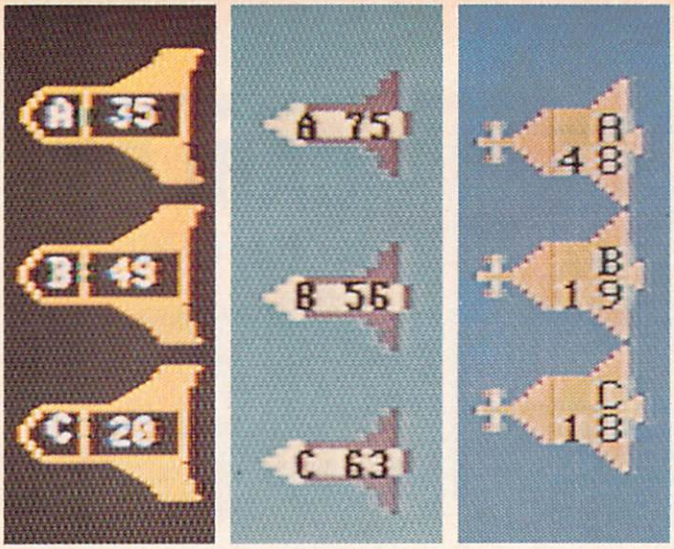
In Act Two, you must descend through the spider crypts—where you and your Ent-trees dodge more spiders and the cruel Hands of Fate—to crush the spiders' larvae. Any that hatch will come back to haunt you in Act Three. Strength and strategy are absolute factors in your continued survival; your point score becomes almost irrelevant. This is where arcade really gives way to adventure.

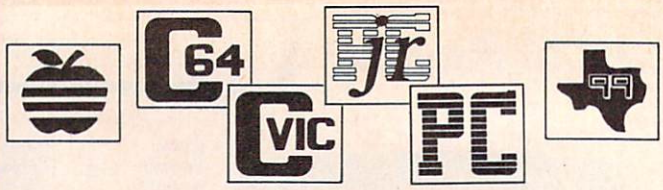
If you have survived this far, Act Three will take you to the Necromancer's lair. Here you must step on all thirteen tombstones in each level, while fending off both the reincarnating wizard and his zombies. Attaining entry to this third act is no small feat, but it's your performance in the preceding acts that determines how well you will do here. If you didn't grow enough trees in Act One to smash most of the spider eggs in Act Two, you won't be able to run or fight quickly enough to get beyond the first levels of Act Three. Like me, you may play the game again and again before you master all of its elements and finally defeat the Necromancer, bringing the enchanted forest back to life.

I enjoyed this game not only for its plot, but for the sheer visual fun of its imagery. Though small and set against a plain dark background, all of the main figures are colorful and well-animated. The trees walk convincingly. Cute little hopping seed pods sidle up to the druid and nudge him on the shoulder when he's too busy fighting to pay attention. Fingers writhe on the Hands of Fate as they grab cruelly from the crypt ceiling. Faces appear in the spider-poisoned trees and scream for the healing wisp.

Compared to the game itself, the instructions are very plain, with just enough information to cover the subject. However, the package art is great. Perhaps someday home computer game visuals may compare well with the artist's brush. The sound effects and music for *Necromancer* quite effectively set the mood and help clarify game action. A lovely little funeral dirge accompanies the opening title, and jazzy syncopated background scores keep pace with the action itself. Even during a close call between druid and spider, the proper sound immediately lets you know who won, without having to look at the bottom screen tally. All of the game's music and noises reflect the Commodore's impressive sound range.

Good animation, colorful visual effects, informative sound, and a strong element of strategy combine to make *Necromancer* a cut above simple arcade games, and a treat for either neophyte or experienced adventure-seekers. You can have it both ways—sitting down for a few minutes of exciting "wisp-em-up," or settling in for a longer quest and a possible look at the reborn enchanted forest. But let the player beware: *Necromancer* has its charms, and not the least of these is the magic by which it will grab, and hold, your attention. HCM





MISSILE MATH

by J.C. Adelmann
and the HCM Staff

Convincing children that multiplication can be fun is about as easy as convincing vegetarians that meat is good for them—they just won't believe you. If this is the case in your household, then give *Missile Math* a try. When math exercises are transformed into computer games, they acquire a whole new appeal to the younger set, making learning a little bit easier. This exercise in multiplication skills is no exception. *Missile Math* is an educational game for the young student who has a basic understanding of multiplication. It provides a simple, yet effective method of testing and developing skills through multiplication of the numbers 1 through 9.

The program begins when you select the number you want to work with. This number will be the primary number in the multiplication problems. For example: If you select 6, then all problems will be composed of 6 times a number from 0 to 9.

You now have the option to view a multiplication table based on the number you selected. If you choose this option, then all of the problems that will be used in the current session will be displayed, with their answers. If you select 6, for instance, then a table starting with 6x0=0 and ending with 6x9=54 will be displayed.

After displaying the multiplication table, the program gets into full swing. Initially, you see three missiles on the right side of the screen. Within each missile is either the letter A, B, or C, and to the right of each letter is a number. One of these three numbers is the correct answer to a multiplication problem showing at the bottom of the screen. Enter your answer choice by pressing either the A, B, or C key.

If you choose a wrong answer, a low tone blurts forth, with the message that the answer is incorrect. The correct answer then displays, and the program starts a new problem.

When you select the right answer, you hear a short musical tune, and the rockets advance toward the left side of the screen. On the PC/PCjr and the Commodore 64 versions of the program, the student is given a little extra incentive to improve his or her score: The program begins by playing only a few notes of a tune. As the score improves, more of the tune is played with every correct response, until finally, the student gets to hear the entire tune.

The score is based on how many problems a student answers correctly from a series of 10. After the user has worked 10 problems, a final screen displays the student's score and percentage of correct answers.

The TI version of *Missile Math* will run in either BASIC or Extended BASIC. Character graphics are used to draw the missiles, and a well-known subroutine is used to display text without scrolling the screen. In most cases, you can take the text to be displayed, separate each character from the string, and convert it to its ASCII symbol. It is then simple to place each character on the screen—one character at a time—with the HCHAR, or VCHAR commands.

It takes time, though, to separate each character from its string of characters and convert them to an ASCII value. We decided speed was of greater concern in this program, so we opted to place the ASCII values in a numeric array, thus eliminating two steps in the process. Now, the program simply indexes into the array to extract a value. This method is much faster, but you pay a price for it—an enormous amount of memory is required to store each character that needs to be displayed.

Remember: a string uses one byte of computer memory to store one character; a numeric array of ASCII values requires nine bytes of memory to store each character. So, if you plan to display very much text, or if your program is pushing the limits of the system's memory capacity, then you may have to settle for the slower method.

Compare the code needed to execute each of these two methods:

METHOD 1 - Numeric array as used in this program

```
200 DIM MS1(22),MS2(12)
220 FOR I=1 TO 22
230 READ MS1(I)
240 NEXT I
1950 FOR I=6 TO 27
1960 CALL HCHAR(22,I,MS1(I-5))
1970 NEXT I
2000 DATA 84,72,69,106,67,79,82,82,69,67,84,106,65,78,83,
     87,69,82,106,73,83,106
```

METHOD 2 - String reconstruction to ASCII format (not used in this program)

```
5000 A$="THE CORRECT ANSWER IS "
6000 FOR I=1 TO LEN(A$)
6010 ASCII=ASC(SEG$(A$,I,1))
6020 CALL HCHAR(22,I+5,ASCII)
6030 NEXT I
```

Method 1 is much faster, but it requires a lot more memory to accomplish the same task.

MISSILE MATH (TI-99/4A)
Explanation of the Program

Line Nos.	
100-180	Program header.
190-270	Reserve space for arrays.
280-310	Input option to display the instructions.
320-450	Display the times table option.
460-480	Get the player's input.
490-960	Subroutine to create a random problem and set up the parameters.
970-1050	Calculate the number right, number wrong, and the percentage right.
1060-1120	Create colored strip for screen graphics.
1130-1440	Determine whether the response was correct or not. Change the numbers in the equation.
1450-1770	Subroutine to create the graphics reward for a correct answer. Move the missiles, and change the numbers displayed in them.
1780-1790	Data statements containing messages to be displayed.
1800-1880	Print instructions on the screen.
1890-1990	Print the correct choice and the answer.
2000-2010	Data statements.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

The Commodore 64 version of *Missile Math* is the only version that takes advantage of sprites. A sprite is a graphics shape that can be placed on the screen with high resolution positioning—it is not restricted to the normal boundaries of character graphics. The computer can place sprites at any pixel position on the screen.

Sprites also have the advantage of large size and flexibility. A sprite's normal size is 23 pixels wide by 20 pixels high. Compare this to a normal character which is 8 pixels wide by 8 pixels high. In addition to their normal large size, sprites can be expanded. Each sprite can be enlarged to either twice its normal width, twice its normal height, or both at the same time. You can then maneuver this large graphics shape around the screen with the highest possible resolution—on the pixel level.

We used three identical sprites in this program to create the three missiles. The subroutine in line 1050 reads the data in lines 1130 through 1150, and places it into memory starting at location 12288. This is enough data for one sprite shape, which is used for all three missile sprites. Line 890 tells the computer where to look in memory for each of the sprites' shapes. (This is done within the FOR_NEXT loop.) By placing the value 192 into locations 2040, 2041, and 2042, you are instructing the computer to get the shape information from the same place, starting at location 12288. The POKEs to locations 53287, 53288, and 53289 set up the sprites' colors.

The sprites are positioned in line 900 by POKEing their vertical coordinates into locations 53249, 53251, and 53253. Their horizontal coordinates are the same, because all three missiles are evenly lined up on the screen at all times. The horizontal coordinates are set in lines 570 to 590. Here, the missiles move toward the left side of the screen every time a problem is correctly answered. The variable XX contains the value of the new horizontal coordinate, while X1, X2, and X3 contain the addresses for the three sprites' horizontal registers—53248, 53250, and 53252.

MISSILE MATH (C-64)
Explanation of the Program

Line Nos.	
100-170	Program header.
180	Branch to initialize program variables, display title screen, and instructions.
190-270	Subroutines to produce sound effects and music.
280-300	Set up variables for display in missiles.
310-320	Clear bottom part of screen.
330-430	Main control loop. Calculate the values for the problem, and display the values inside the missiles. Display the problem at the bottom of the screen.
440	Accept user's response to the problem.
450-470	Display correct answer.
480	Check for correct response.
490	User entered incorrect answer.
500-590	User entered correct answer. Give graphics and music reward.
600-610	Wait for a key to be pressed before continuing to the next problem.
620-660	Initialize main control loop.
670-780	Program begins and restarts with this routine. Enter the number to work with and display the times table.
790-870	End of the 10 problems. Display the score and give the option to do it again.
880-920	Initialize program variables and constants.
930-1040	Display title screen.
1050	Place graphics information in memory for the missile, which is a sprite.
1060-1080	Load music routines into memory.
1090-1120	Input option to display the instructions.
1130-1180	Data containing the information for the sprite graphics and the music routines.
1190-1270	Display instructions.

For the Key-In listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

Text continues on next page

Missile Math

The VIC-20's low memory capacity and lack of sprites may turn most computer users away, but this program is proof that the VIC-20 computer can be just as useful and enlightening as its big brother. This version of *Missile Math* makes perfect use of the VIC-20's built-in graphics character set to create the three missiles. The routine to do this is in lines 810 through 920. As you can see if you scan the code characters in these same lines, you can change colors, print special graphics characters, and even reverse them—all from the same statement.

At first glance, the listing looks very confusing. This is because the special characters of the VIC keyboard are hard to decipher from a printed listing. For this reason, they are translated into short notes about each key, and surrounded by the pointing hands.

Graphics are simple to implement—if you understand how the VIC allows you to place special graphics commands right in the PRINT command. Any graphics character or KEY command that can be entered outside of the program can be used in the PRINT statement.

To perform screen positioning, the cursor control keys can be placed in the PRINT command as well. (See line 1080.) Here, the screen is first cleared and the cursor is returned to the upper left corner of the screen because the [SHIFT CLR] key is imbedded in the PRINT command. The next imbedded command is 4CRSRDOWN, which indicates that you should press the cursor-down key four times. When the program encounters this in the PRINT command, the cursor will automatically move down four lines before the next item in the print string is reached. That next item is the text that you wish to print on the screen.

Thus, with this one PRINT command, you have cleared the screen, taken the cursor to the Home position (upper left corner), moved down four lines, and printed the message MULTIPLICATION GAME.

MISSILE MATH (VIC-20) Explanation of the Program

Line Nos.	
100-170	Program header.
180-220	Display title screen and ask if users wish to view the instructions.
230-240	Initialize program variables and set screen color.
250-290	Input the number to be worked with.
300-360	Ask users if they would like to see the times table, and then display the table.
370-440	Load up the R( ) array with 10 numbers for the problems.
450-550	Set up problem and branch to display it.
560-580	Get user's answer.
590	Check for right answer.
600-640	Wrong answer given.
650-760	End of session. Display results and percentage right. Give option to do it again.
770-970	Display the missiles and the problem.
980-1070	Routines for producing musical tones.
1080-1130	Display instructions.
1140-1150	Data containing information for the musical routine.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

Although the Apple computer is very powerful, it has been around for a long time, which means it must sometimes rely on old technology to perform some tasks considered commonplace with current systems. This is evident in the Apple's ability to create music—machine language is the only feasible method of of producing musical notes on the built-in speaker.

We wanted to use music as part of the reward for answering equations correctly, so it was necessary to include a short, useful, machine-code routine in the program. The entire routine is contained in one line of data in line 250. The subroutine in line 240 loads this routine at location 770.

The routine is easy to access, and makes musical programs a snap. Examples of accessing this routine are in lines 1580 through 1610, and line 1640. The value POKEd into location 768 sets up the duration of the tone—the larger the number, the longer the duration. The frequency is determined by the value you POKE into location 769—the higher the value, the higher-pitched the tone will be. If you use this routine outside of this program, you might want to experiment with your own values to produce desirable effects. To call the routine once the frequency and duration have been set, use the CALL command to location 770.

Have you ever wanted to re-RUN your program while keeping some of the variables intact? When a program is re-RUN, it clears the values out of all of the variables. But, you can save those values to memory before re-RUNing the program, and then re-load the variables after the program begins running again.

In this version of *Missile Math*, we wanted to save the user's name so that you wouldn't have to re-enter it every time you restart the program. In line 510 the user's name is input, and line 520 saves the name to memory. It accomplishes this by first plucking out one character at a time with the MID$ function. That character is then converted to its ASCII value. The ASCII value is poked into an area of memory that is not affected by re-RUNing the program. Later, in line 640, the ASCII values are PEEKed out of memory and converted back to their string equivalent with the CHR$( ) function.

This procedure can be used for a number of your own applications: you could pass values or strings back and forth between programs, or—as in this case—when re-RUNing the program. If used properly, parts of memory could even be used for temporary binary or ASCII files. These files, though limited in size, could be kept off of the disk environment and passed between programs or subroutines.

MISSILE MATH (Apple)
Explanation of the Program

Line Nos.	
100-170	Program header.
180-230	Display title screen.
240	Poke music routine into memory.
250	Data for the music routine.
260	Poke shape table into memory.
270-370	Shape table for the missiles.
380-480	Initialize program graphics and variables.
490-560	Accept the user's name. Save the name in memory so that subsequent RUNing of the program won't cause the program to forget it.
570-580	Accept option to display the instructions.
590-630	Input the number to practice with. Also set up variables for the program.
640-670	Recover the user's name when the program has been re-RUN.
680-770	Input option to display the times table. Display times table if the option was selected.
780-1130	Main control loop. Determine the problem, and display it and the missiles on the screen.
1140-1240	Input answer, and check for correct or incorrect response.
1250-1270	Branch to move the missiles, then return for next problem if less than 10 problems have been worked.
1280-1340	End of session. Option to do it again.
1350-1440	Print instructions.
1450-1560	Subroutines to display the missiles and answers.
1570-1620	Routine to play a tune when the right answer is selected.
1630-1640	Routine to play a tune when the wrong answer is selected.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

Text continues on next page

Missile Math

PC jr

This version of *Missile Math* will run on either the IBM PC or PCjr. On the PC, a color interface adapter and color monitor are required.

The screen graphics consist of three rockets which move from right to left every time an answer is correct. These three rockets are surprisingly fast to draw, considering their size. You can draw the rocket the conventional way back on the title screen, and save it in an array with the GET statement. In the main sequence of the program, all that's needed to place the rocket back on the screen is a PUT statement supplied with the screen coordinates.

The game's music works well on the IBM PC, because it is all generated through the internal speaker. If you have a PCjr and would like to output the music to an external speaker, such as the one in your television set, you can add this line:

```
195 BEEP OFF:SOUND ON
```

The IBM PC and PCjr generate music with a very easy-to-use command: PLAY. The PLAY command lets you program music using the actual musical notes familiar to most people. (See line 560.) Here, we assign a string of music commands to the string variable A$, and tell the PLAY command to execute this string twice. A number of commands are available for creating the music—they can be placed in a string variable as we did here, or played directly with the PLAY command. Take a look at the first part of line 560:

```
A$="L16T60O2F#...
```

This part of the string set the length of the notes to sixteenth notes (L16). The tempo was set to 60 beats per minute with the T60 command, and the octave was set to 2 with the O2 command. Finally, the note F# (# for sharp) was entered. As you can see, this makes writing music as easy as thinking up the notes.

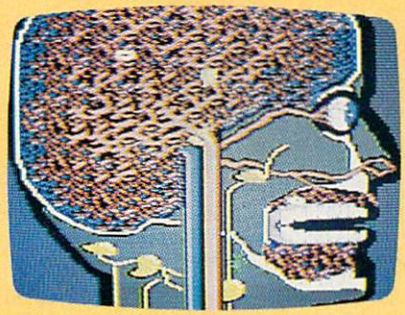
MISSILE MATH (IBM PC and IBM PCjr) Explanation of the Program

Line Nos.	
100-190	Program header.
200-240	Clear the screen. Assign the music reward commands to P1$, P2$, P3$, and P4$.
250-260	Draw the missile on the screen, and store its shape in the ROCKET( ) array.
270-280	Display the title screen, and prompt for instructions option.
290-300	Prompt for the number to practice with.
310-340	Prompt for the times table option. Display the table if it's selected.
350-390	Find next problem.
400-420	Display the letters A, B, and C, along with three values in the missiles.
430-440	Display the new problem and accept the answer.
450	Check for the right answer.
460-470	User entered wrong answer.
480-540	User entered right answer. Play a tune and advance the score.
550-560	User got all ten answers correct. Give special reward.
570-580	End of session message and score. Option to play again.
590	Display the instructions.
600	Routine to display the missiles at their new location.
610	Key scan routine.

For the Key-In listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

HCM

HCM Review



Name:	Microsurgeon
Program type:	Surgical Simulation
Machine:	TI-99/4A, IBM PCjr
Distribution:	Imagic Co. 981 University Los Gatos, CA. 95030
Price:	$39.95 IBM cartridge $29.95 TI cartridge

	Poor	Fair	Good	Excellent
Performance:				
Engrossment:				
Documentation:				

MICROSURGEON

A review by Steve Nelson

HCM Staff

A A few years ago I saw a movie about a team of doctors and scientists who were reduced in size along with a submarine-like ship and injected into a patient's bloodstream in order to perform microsurgery. At the time, the movie was considered pure science fiction, but technology has changed the face of medicine in many ways. Although scientists still are unable to reduce people down to the size of a speck of protoplasm, the concept of microsurgery is no longer just a figment of some screenwriter's fertile imagination.

Specialists in the field of medicine are performing microsurgery every day; but why should they get all the gorey—I mean, the glory—just because they went to college for ten years? Thanks to Imagic's new game, *Microsurgeon*, you too can perform delicate microsurgery and pilot a robot probe through the body of your patient.

The object of the game is to cure your patient of one of eight different conditions—each at a separate location in the body. You begin the game by selecting one of three levels of play (student, intern, surgeon). You are offered a choice of victims . . . er . . . patients to operate on, and are given some power units, the number of which increases or decreases during the game depending on your operating skills. Power units allow you to move your probe and dispense medication inside the body of your patient. You maneuver the probe in the body through the bloodstream and lymph system to discover what type of condition must be treated in each of the eight locations. Once you determine the condition, you cure it by shooting the disease with either ultrasonic rays, antibiotics, or aspirin. Each medication treats different conditions so you must be sure you have the correct diagnosis and cure.

Be forewarned, however, that while you are curing the brain, the lungs may go terminal. You must *monitor* all areas of the body to be sure you don't spend too much time in one place and neglect another. In the 99/4A version of the game, the patient status screen is visible at all times, but on the PCjr, you must periodically access the status screen by pressing [ENTER] or by pressing the two joystick buttons simultaneously.

Moving through the body is accomplished by using the keyboard or joystick. I preferred the keyboard because it gave me more control of the probe than the joystick did. Staying inside the blood vessels or lymph system allows you to move more quickly through the body searching for disease. If you stray out into the organs, your speed is slowed and your probe is subject to attack by white cells which take away power units if they touch your probe.

You must cure all eight conditions in the body and then exit through the eye, ear, nose, or mouth. Leaving by any other route causes the patient to go terminal.

Microsurgeon is quite challenging on the surgeon level and you will lose quite a few patients before your operating skills develop. The game comes with instructions that are very detailed, and it is a good idea to read them thoroughly before beginning to operate.

At the end of the game, assuming that you save your patient, you are awarded research funds based on the number of individual organs you saved, the power units left, and the overall condition of the patient.

The game looks quite different on each computer. The PCjr version has full screen graphics, and they are much more detailed than the 99/4A version of the game. In fact, if you have a weak stomach, I wouldn't recommend the IBM version because it is so realistic.

Another big difference between the IBM and the TI version is in the use of sound effects. *Microsurgeon* on the 99/4A employs sound effects galore. If you have a speech synthesizer, the computer talks to you as you move through the body. If you don't have a speech synthesizer, you are still treated to a wide assortment of sounds—including the patient's heartbeat and the sounds from inside the intensive care room. Unfortunately, implementation of the PCjr's sound effects are almost nonexistent—a difficult situation to understand because both the 99/4A and PCjr have the same TI sound chip.

Even though one version has far superior graphics, and the other, far superior sound, both have one important thing in common: they offer their purchasers an inexpensive and enjoyable way to vicariously experience a situation that very few get the opportunity to participate in in real life. And isn't that what computers are all about . . . HCM

HCM Review

Name:	Trickster Coyote
Program Type:	Educational Game
Machine:	Apple II, II Plus, IIe: 48K
Distributor:	Reader's Digest Microcomputer Software Division Pleasantville, NY 10570
Price:	$39.95

	Poor	Fair	Good	Excellent
Performance:			■	
Engrossment:				■
Documentation:				■

TRICKSTER COYOTE

A review by Wayne Koberstein

HCM Staff

Trickster Coyote has one good trick: he teaches words. Not content to lie there like a dictionary, he leads you on a merry chase throwing words in your path. Will you stumble? Or will you learn? If you match up all the words with their proper synonyms, Coyote will have to drop his disguises and give back the Totem of the Straight Tongue. But new lies and new words always come to Coyote. So if you tire of his old tricks, he will challenge you to learn some new ones yourself.

Native Americans once used Coyote in myth to pass on the subtle teachings of their culture. Coyote was cunning and fast, and you could learn a lot just by trying to catch him. Now Reader's Digest has cast him in a new role, as teacher of vocabulary. Computer wiz kids spend hours chasing sprites—so why not let them learn words in the process? As a teaching game, *Trickster Coyote* is smart if not cunning, and challenging if not exactly fast.

Looked at as a series of stills, this game is very attractive. Colored frames are bright, and the characters—including the "kid", and Coyote with his disguises—are simple but well-drawn against a dark background. Animation consists of intercutting these few stills to create a minimum of movement. Unfortunately, this slow, repeated motion tends to dampen the game's basic visual appeal.

Trickster Coyote actually includes two games in one package. In the title exercise, a young figure and the coyote jump from log to log in a creek. Some of these logs bear words, one of whose meaning agrees with a word at the top of the screen. You must jump over any log bearing the wrong word. Step on the wrong log, and you fall in the creek. Trickster Tag, the second game, requires memorizing in ten seconds four words which appear in a quartered square. When these words disappear, the silhouette of Coyote flashes inside alternating frames as definitions are displayed overhead one at a time. You press the space bar when he appears in a frame that corresponds to the word fitting the current definition. There is no "wipe-out" point. Play continues until you give up or win. As you can see, there is nothing too tricky in the "arcade" features of the game; the trick lies in mastering words and their definitions.

To add to the challenge, you may select from three levels of difficulty. Words like "versatile, cotyledon, and ostentatious" grace the highest level; while "define, wept, and abundant" are likely to appear at the lowest. "Warm-up" is an option to preview words with definitions and in-context sentences. Additionally, *Trickster Coyote*'s dictionary can expand with the player's vocabulary. You can add new words and definitions in the Edit mode. Unfortunately, the instructions do not say whether new words replace old in gameplay, or if they just enter the pile for random recall. Neither is there any mention of a Delete function, which would make the small dictionary infinitely variable, extending its usefulness for repeating players.

Equally unfortunate are the stretches of time when nothing seems to happen but a series of menus leading up to actual play. I have no complaint with the style of these panels; they are simple, clear and direct. But I would have sympathy for any young person who grows tired of responding to menus instead of getting on with the exercise. Maybe I've just been playing too many video games . . .

While I'm airing gripes, why not mention the one tricky physical aspect of playing this game: jumping over logs. You would think that pressing the space bar during the actual jump would be logical. But noooo . . . You press the bar as you come down on the log that is *before* the one you want to jump over. Tricky, eh? I didn't get my timing down on this one until I'd fallen in the creek a few times. Honest, I knew the word!

Trickster Coyote is not really an exciting or suspenseful game. But it is a pleasant way to learn new words. Increasing difficulty means more difficult words, not a faster pace or tests of dexterity. This emphasis on *words* rather than arcade *action* is appropriate, although the slow pace could cause some students, young or old, to lose interest. Perhaps the best setting for this game would be short, frequent sessions with the cooperation of an adult, either at home or in the classroom. Nevertheless, within the context of learning, *Trickster Coyote* presents an alternative much more attractive than most any simple word/picture book for the young.

And for those who say that you can't teach old dogs new tricks, heed the call of the old Coyote—he makes a pretty good teacher.

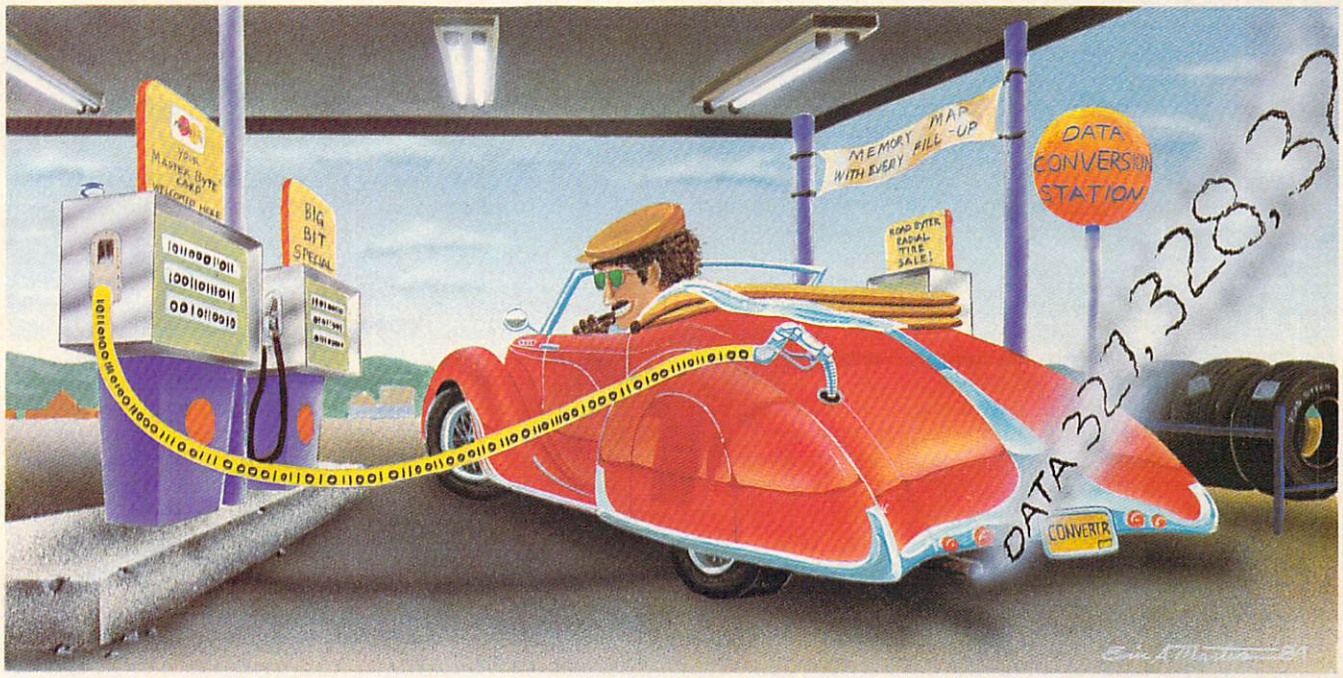
HCM

Converting Machine Language into BASIC Data Statements

by John Thrasher
and the HCM Staff

Are you a Commodore 64 owner who enjoys programming in BASIC? Do you also enjoy writing routines in assembly language? Then pay attention. This article is for you.

CONVERTIBLE



FOR COMFORT

Extending the power of BASIC by including assembled machine language subroutines is usually very difficult—due primarily to the tedium of converting machine language code to BASIC data statements. The routine provided here will allow you to sit back in comfort while your computer does the converting for you. You will save considerable time by eliminating both the old-fashioned hand-generation of data statements, and the painstaking debugging necessary to ensure that these data statements are error-free.

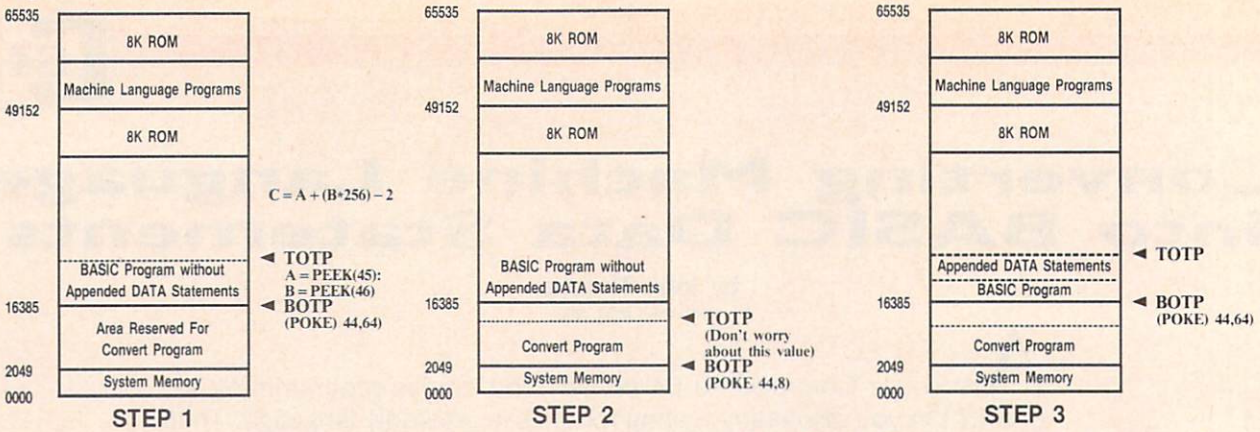
First, we will see how to create room in memory for our labor-saving *Convert Memory to BASIC DATA Statements* program. Then we will examine how the program turns machine language routines into data statements and appends them to a BASIC program. Once all of this is in place, the BASIC program will RUN—automatically POKEing and calling the machine language subroutines with the BASIC data statements.

This program converts machine language programs—stored in *non-consecutive* blocks of memory—to *consecutive* DATA statements. You can also modify this stand-alone program so you can insert it in another BASIC program to convert any memory block—screen memory, programmable character definitions, or sprite definitions—into DATA statements. Last issue's article, *Quick Pixel Trix*, was an example of this; it used the technique to convert sprite and character definitions into BASIC DATA statements.

Pointers to Remember

To use this *Convert* program successfully, you must first make room for it in memory. To do this, you must understand and manipulate two pointers: the BOTP (Bottom of BASIC Text Pointer), located at addresses 43 and 44, and the TOTP (Top of BASIC Text Pointer), located at addresses 45 and 46. These pointers are in the standard low-byte-first and high-byte-second format. They are maintained and changed by the BASIC operating system. The BOTP is always maintained at address 2049 (with a 1 in the low byte and an 8 in the high byte), and

Figure 1. MEMORY MAPS



This diagram shows the relative values of the **TOTP** and **BOTP** to demonstrate how Commodore memory is used.

In step 1 the program (that the **DATA** statements are to be appended to) is loaded into memory starting at address 16385. This starting address is achieved by **POKE**ing 44,64. After this program is loaded, **PEEK** the **TOTP** (addresses 45 and 46) and derive a value for C (first DATA Statement address) as shown. The value of C will be prompted for when you run the *Convert* program.

In step 2 we load the Convert program into memory after moving the **BOTP** to address 2049 (**POKE 44,8**).

Step 3 shows the memory after the Convert program has been **RUN** and the conversion completed. It is then necessary to move the **BOTP** and the **TOTP** so that they surround the program with the appended **DATA** statements. The values to be **POKE**d into the **TOTP** will be displayed on the screen. To move the **BOTP**, **POKE** 44 with 64 (address 16385).

this is the starting address where your BASIC program will be LOADed in memory. The value of this pointer will remain unchanged unless altered by you. When you LOAD a program from disk or tape, the operating system automatically sets the TOTP at the address two bytes past the end of your BASIC program.

If you do change the address of the BOTP, then each time you LOAD or SAVE a BASIC program the operating system will think that the new BOTP address is the starting address for LOADing and SAVEing. Then you can have several BASIC programs in memory at the same time. First, LOAD a program; then set the BOTP to an address above the first program, and you can LOAD another program and still have the first one in memory. You can do this repeatedly, until you run out of memory. It would not be possible, however, to RUN each and every one. You may RUN only the program that is LOADed when the BOTP equals 2049. You can, however, SAVE and LIST any one of the programs in memory by POKEing the BOTP with the address of the start of the program and the TOTP with the address two bytes past the end of the program.

Remember that the TOTP pointer is maintained by the operating system, and set to the address two bytes past the end of your BASIC program, when you LOAD a program from tape or disk. After LOADing your program, PEEK addresses 45 and 46 and remember their values. This will be important later, when we append DATA statements to our programs.

Using the Program

With all this preliminary information digested, you are now ready to progress through the sequence of steps used to append your machine language programs, as DATA statements, to the end of your BASIC program. The following steps implement the conversion:

1) **Reset the system.** Simply hit the [RUN/STOP] [RESTORE] keys, or turn the power off and back on again.
2) **LOAD the machine language programs to be converted to DATA statements.** For example, you might have three machine language routines that were developed with the Commodore *Assembler Development* package. Using the *LOLOADER64*, you can LOAD them into memory.
3) **POKE address 44—the high-order byte—with 64 (BOTP). Now set the BOTP pointer to address 16385.** This will leave plenty of room in memory to LOAD and RUN the *Convert* program starting at address 2049.
4) **LOAD the program that the DATA statements will be appended to.**
5) **PEEK addresses 45 and 46 (TOTP).** The values PEEKed in this step will be used to obtain the starting address where we begin to assemble the DATA statements.
6) **POKE address 44 with 8 (BOTP).** We change the BOTP so that we may LOAD and RUN our *Convert* program starting in the normal programming area.
7) **LOAD *Convert Memory to BASIC DATA Statements* program.**
8) **RUN the program.**
9) **Enter starting and ending addresses for the three machine language routines.** When the program starts running, it will prompt for the beginning and ending addresses (decimal) of up to 11 blocks of memory to be converted to DATA statements. The program will create DATA statements for each block and start a new DATA statement at the beginning of each block. This will make it easier to ascertain divisions between separate blocks of DATA statements.
10) **Press [RETURN] without entering a value, to allow the program to continue.** When you have finished inputting the beginning and ending addresses, the program will prompt for the starting address of the next block of memory to be converted. If you then hit [RETURN] without entering a value, the program will continue into its next phase of operation.
11) **Enter a starting line number higher than the highest line number of the program to which you're appending.** For example, if the BASIC program with DATA statements appended ends at line 950, you could then enter a starting line number of 1000.
12) **Enter an increment value of 10.** We want our DATA statements to begin at line 1000 and progress by increments of 10.
13) **Enter as the starting address the number found by subtracting 2 from the value obtained in step 5.** We convert the values we PEEKed in step 5 into an equivalent decimal address by multiplying the value PEEKed from address 46 by 256 and adding this to the value PEEKed from address 45. From this we subtract 2 and find the starting location for our new DATA statements.
14) **Wait for the program to finish.**
15) **POKE the values returned by the program into TOTP.** With the addition of DATA statements, the program has increased in length. Therefore, in order to SAVE the lengthened program, move the TOTP to the new end of the program. The *Convert* program calculates the values to be POKEd upon completion.
16) **POKE address 44 with 64 (BOTP).** We move the BOTP so we are able to save the new program.
17) **SAVE the new program, which now has the DATA statements appended to it.**
18) **Reset the system.** Now we are finished and can reset the system by turning the machine off and then on again. The system reset automatically restores the TOTP and the BOTP pointers to their default values.

Convert Memory to BASIC DATA Statements (C-64)
Explanation of the Program

Line Nos.

100-170	Header.
180	Initialization.
190-270	Loop to input starting and ending values for memory blocks that will be converted.
280-330	Initialization.
340-550	Main program loop.
560	Insert end of program markers.
570	Derive POKE values for TOTP.
580-710	Print statements telling user how to save or list new program.
720	End.

COMMODORE 64

```
100 REM  ***********************
110 REM  *  CONVERT MEMORY  *
120 REM  *  TO DATA STMTS.  *
130 REM  ***********************
140 REM  BY JOHN THRASHER
150 REM  HOME COMPUTER MAGAZINE
160 REM  VERSION 4.3.1
170 REM  C-64 BASIC
180 AC=1:POKE 53281,1:PRINT"■SHIFT CLR■
    "
190 PRINT "■CTRL BLK■ENTER MACH. LANG.
    START ADDRESS";AC
200 INPUT "";SA(AC)
210 IF SA(AC)=0 THEN 280
220 PRINT "ENTER MACH. LANG. ENDING ADD
    RESS";AC
230 INPUT "";EA(AC)
240 IF EA(AC)>SA(AC) THEN 270
250 PRINT "INPUT ERROR. TRY AGAIN"
260 GOTO 190
270 AC=AC+1:GOTO 190
280 REM
290 INPUT "ENTER STARTING LINE NUMBER";
    SL
300 INPUT "ENTER INCREMENT VALUE";IV
310 INPUT "ENTER STARTING ADDRESS FOR D
    ATA";SD
320 SL=SL-IV:DT=131:CM=44:N1=AC-1:AC=1
330 DP=SD+2:SL=SL+IV:PRINT "■SHIFT CLR■
    ■2CRSRDOWN■■2CRSRRIGHT■";SL
340 IF N1=0 THEN 560
350 LH=INT(SL/256):LL=SL-(256*LH)
360 POKE DP,LL:POKE DP+1,LH:POKE DP+2,D
    T:POKE DP+3,32:DP=DP+4
370 FOR X=1 TO 16
380 SN=PEEK(SA(AC)):CT=2:IF SN=0 THEN C
    A(CT)=10:GOTO 410
390 CA(CT)=INT((SN+.0001)/10↑CT):SN=SN-
    (CA(CT)*10↑CT):CT=CT-1:IF CT>-1 THE
    N 390
400 CT=2
410 IF CA(CT)>9 THEN 440
420 IF CA(CT)=0 THEN CT=CT-1:GOTO 420
430 CA(CT)=CA(CT)+48:POKE DP,CA(CT):DP=
    DP+1:CT=CT-1:GOTO 450
440 POKE DP,48:DP=DP+1:GOTO 460
450 IF CT>-1 THEN 430
460 SA(AC)=SA(AC)+1:IF SA(AC)>EA(AC) TH
    EN X=16:GOTO 480
470 IF X<16 THEN POKE DP,CM:DP=DP+1
480 NEXT
490 POKE DP,0:DP=DP+1
500 SH=INT(DP/256):LS=DP-(SH*256)
510 POKE SD,LS:POKE SD+1,SH
520 SD=DP
530 IF SA(AC)<EA(AC) THEN 330
540 IF SA(AC)=EA(AC) THEN 330
550 AC=AC+1:N1=N1-1:GOTO 330
560 DP=DP-2:POKE DP,0:POKE DP+1,0
570 DP=DP+2:PH=INT(DP/256):PL=DP-(PH*25
    6)
580 PRINT "■SHIFT CLR■DATA STATEMENTS H
    AVE BEEN ASSEMBLED."
590 PRINT "■CRSRDOWN■TO SAVE DATA STATE
    MENTS AS A PROGRAM"
600 PRINT "FILE, THE END OF TEXT POINTE
    RS MUST"
610 PRINT "BE MOVED TO THE END OF THE D
    ATA"
620 PRINT "STATEMENTS. PLEASE POKE THE
    FOLLOWING"
630 PRINT "VALUES."
640 PRINT "■2CRSRDOWN■POKE 45,";PL
650 PRINT "■2CRSRDOWN■POKE 46,";PH
660 PRINT "■2CRSRDOWN■THE BEGINNING OF
    TEXT POINTERS"
670 PRINT "(ADDRESSES 43 AND 44) MUST A
    LSO BE"
680 PRINT "CHANGED. THESE VALUES, HOWE
    VER, ARE"
690 PRINT "DEPENDENT UPON HOW YOU HAVE
    UTILIZED"
700 PRINT "THIS PROGRAM. PLEASE SEE AR
    TICLE ON"
710 PRINT "THIS PROGRAM FOR MORE DETAIL
    S."
720 END
```

HCM

Boolean Brain ... from p. 121

The PCjr version of *Boolean Brain* makes good use of the DRAW command to create graphics on the screen. The DRAW command allows the programmer to give the computer easy drawing instructions such as D for down, and L for left. You can optionally include a number after each of these subcommands to cause that subcommand to repeat. These subcommands are imbedded in a subcommand string that can be used directly with the DRAW command, or placed in a string variable for later use.

The PALETTE command lets us assign to areas, or objects on the screen, any one of the 16 colors available in the system. It provides a great deal of power—allowing us to change the color of large areas of the screen *in an instant*, without having to go through the much slower process of "re-painting." In this program, the logic gates are painted with color number 4 (initially red). The PALETTE command rapidly changes the color assigned to number 4, imparting a strange flashing effect to the logic gates. Once a gate is turned on, its color number is changed to green, the default color assignment (color number 2).

How The Program Logic Works

The program tests your selected input in line 450 and branches to the first routine. If the player chooses 1 as an input, the program branches to line 460. Here, the score is increased, and the first line to the first gate is painted green. The GT( ) array is updated to keep track of which inputs have been turned on.

The first gate itself is then tested to see what kind of gate it is. If the gate is type 2 (OR gate), then the gate becomes *active*; the program then branches to line 560, where the gate's color and status are changed and the next gate (connected to the first gate's output) is tested.

If the gate has been set to type 1 (AND gate), the second input to the gate would be tested to see if it also was turned on. If the second input to the gate is also turned on, the program could branch again to line 560. (If the gate had been set to 0—indicating a gate already active—the program would simply return to get another input choice.)

In line 560 the gate's color is set to green, and the proper subroutine is called to paint green the output line to the next gate. The next gate is similarly tested to determine whether it's an AND gate or an OR gate. If it's an OR gate, the program continues on to line 610 for the next gate. If the gate is an AND gate, then the second input must be tested to see if it has been turned on.

This procedure continues until the player is successful in connecting the logic gates all the way to the right side of the screen. Once the last gate is turned on (line 640), you are taken back to the view of the room you tried to enter. The door you opened is now behind you and off the screen. The door will remain open throughout the rest of the game.

THE BOOLEAN BRAIN (IBM PCjr) Explanation of the Program

Line Nos.	
100-180	Program header.
190	Initialize the system with 32K for graphics mode 5, and dimension arrays.
200-220	Set up program variables and draw the first room.
230-250	Main control loop. Get direction commands and branch to appropriate routines.
260-410	Check to see if the doors are open or closed for the routines.
420-650	Calculate which gates are to be turned on after the user has selected an input while trying to open a door.
660-670	Display control panel graphics in the CPU room.
680-690	Player has made it to the CPU. Display message and prompt player for another game.
700-710	Player goes into a bad sector on a diskette and is zapped. Display message, and prompt for another game.
720	Input option to play another game.
730-810	Draw the connecting lines between logic gates.
820-830	Display input numbers on the logic gates.
840-890	Control subroutine for displaying the logic screen Also, flash the gates with the PALETTE command.
900	The DRAW command is used to create the 3-D room.
910-930	Draw the three doors, and the control panel for two of the doors.
940-960	Subroutines to paint the doors when they are closed.
970-980	Two subroutines to draw the logic gates with the DRAW command.
990	Display the title of the room that you are in at the bottom of the screen.
1000	Data for the screen position of the logic gates.
1010	Data for the room information includes the name of the room and the adjoining room numbers for each of the four exits from the room.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

Boolean Brain ... from p. 121

The Apple version of *Boolean Brain* is played in exactly the same way as the IBM version—variations in the game are very slight. (For example, the direction you are facing is indicated at the bottom of the screen instead of above the door.) Due to the way the Apple handles high-resolution (hi-res) graphics, however, its program implementation is quite different.

We used three different methods to create the game's graphics. Because numbers had to be placed on the hi-res screen to identify the logic diagrams, we first made up a shape table for the ten digits. The table is in the DATA statements in lines 1600-1630, and is POKEd into memory by line 230.

And because most of the program's other graphics filled the entire screen (the rooms in particular), we draw by using HPLOT statements in FOR_NEXT loops. (Examine lines 1130-1360 to see exactly how the rooms are drawn.) This second method is a little more straightforward than shape tables, because the tables tend to become unwieldy when filling large areas on the screen.

The third method we used displays flashing colors when a player reaches the CPU. Here the graphics are POKEd directly onto hi-res page 1. Meanwhile, the sound effects (either a click of the speaker or the bell) are interspersed with the POKE statements to give the desired effect. The display is handled completely by one line:

```
890 FOR H = 1 TO 2: FOR I = 0 TO 7068 STEP 1024: FOR
    J= 8244 TO 8638 STEP 128: FOR K= 0 TO 15: POKE
    I + J + K,L( INT( RND (1) * 3)):M = PEEK ( - 16336):
    NEXT : PRINT CHR$ (7);: NEXT : NEXT : NEXT
```

In this line, the values of I, J, and K combine to form the address of the hi-res screen location where a byte is to be placed. The L( ) array was DIMed to 3 (line 190), and each member was given a value (line 200) such that it would block one of the six hi-res colors to be placed on the screen. The color selected depends upon the column where the byte is placed. One of the bytes is randomly selected by the **L(INT(RND(1)*3))** in the line above, so the screen location of any color is left quite to chance.

One big advantage of the Apple system is its two separate pages of hi-res graphics available at the same time. Therefore, to help speed up program execution, we display the rooms on page 1 and the logic diagrams on page 2. Then, by changing the graphic's *soft switches*, it is possible to display the room without having to redraw it each time. After the computer initially draws the basic room (subroutine starting on line 1120) all it takes is three POKEs in line 270 to switch the display from page 2 to page 1 and re-display the room:

1. POKE 230,32 selects hi-res page 1 to be the page currently displayed.
2. POKE 49236,0 switches all hi-res commands directed to hi-res page 2.
3. POKE 49235,0 selects mixed-mode (four lines of text at the bottom of the display).

We used these commands instead of the usual HGR, because HGR clears the screen, and we wished to leave most of the display unchanged. Once these POKEs are done, the room is immediately displayed, and the computer has only to open and close doors to make the room correct.

THE BOOLEAN BRAIN (Apple) Explanation of the Program

Line Nos.	
100-170	Program Header.
180-200	Set LOMEM: to protect hi-res pages, DIM arrays, initialize variables.
210-220	READ values into arrays.
230-240	READ and POKE shape table.
250-260	Draw room the first time, and set initial direction room and score.
270-300	Beginning of main loop, display room, and get player's input.
310-320	Analyze input, jump to zap if moved onto disk.
330-380	Switch to logic diagram display, and get player's input.
390-850	Analyze player's input and turn on appropriate gates, until door unlocked.
860-870	Move to next room and repeat main loop, jump to end game.
880-920	End game, display score, and ask for replay.
930-960	Subroutines to print direction.
970-1030	Subroutine to set colors of doors.
1040-1100	Zapped routine, replay option.
1110-1260	Subroutine to draw room first time.
1270-1360	Subroutines to draw doors.
1370-1540	Subroutines to draw logic diagrams.
1550-1560	Data statements for rooms.
1570-1630	Data statements for shape table.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

HCM

EL DORADO TRADING GROUP

Your Exclusive Western U.S. Texas Instruments High Technology Distributor.

With over 1 million 99/4A owners wanting software and accessories, now's the time to jump on our wagon!

YOUR PROFIT GOALS WILL "PAN OUT" WITH US!

Our support services will help your company reach its profit goals by keeping you abreast of the newest and fastest turning high technology items in the line.

"PICK" THE HOTTEST SELLING SOFTWARE

Our trained sales staff will assist you in ordering the "fastest turning" software by suggesting new packages and proven best sellers such as Parsec (PHM 3112), Early Learning Fun (PHM 3002), Dragon Mix (PHM 3117), TI Invaders (PHM 3053), and Munch Man (PHM 3051).

EXPRESS DELIVERY

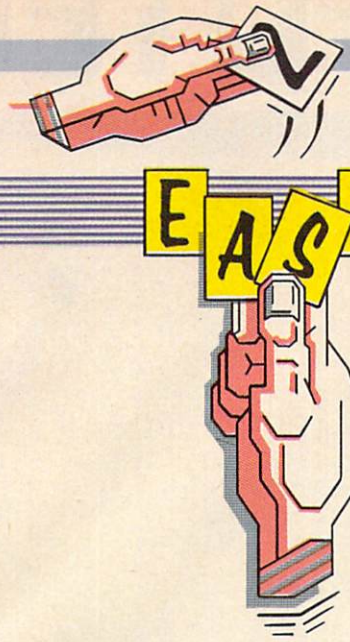
Our extensive inventory allows your order to be processed within a 24 hour time frame.

DON'T MISS OUT ON THE RUSH!

Call us today, **TOLL FREE 800-227-8292,** in California **800-248-2224** except 415 area code call **494-6600** for a confidential "High Technology" dealers price catalog. All dealers with an assigned customer number and a retail storefront will want to act NOW to ensure delivery during the primary 4th quarter selling season.

EL DORADO TRADING GROUP
760 San Antonio Rd.
Palo Alto, CA 94303

Circle 156 on Reader Service Card.



EASY SCRIPT REVIEW

by Greg Roberts

Name:	EasyScript
Program Type:	Word Processor
Author:	Simon Tranmer
Machine:	Commodore 64
Distributor:	Commodore Business Machines 1200 Wilson Drive Westchester, PA 19380
Price:	$99.95, diskette
	Poor Fair Good Excellent
Performance:	■■■
Ease of Use:	■■■■
Documentation:	■

The Commodore 64 has been blessed with dozens of good writing programs, including Commodore's own deluxe model, *EasyScript*. It is a powerful device offering some luxuries not found in most word processors in this price range.

Unlike many word-processing programs, this one gives you an uncluttered writing area, confining the Mode and Command information to a narrow band at the top of the screen. Programs that use line numbers, icons, and other crutches may be friendly at first, but the crutches can get in the way once you've got your writing legs.

With *EasyScript* your text wraps automatically. The words, however, break indiscriminately and make your work hard to scan on the screen. Fortunately, the text is easier to read when properly formatted as a printout. And there is another option: You can get a screen image of the finished format by pressing [F1] plus 0 for output, then V for video.

Because word processing consists mainly of adding and deleting text, the insert and delete commands should be among the first features scrutinized by a prospective buyer. As in any writing program for this machine, *EasyScript* uses the Commodore's powerful insert/delete key for cleaning up minor typing errors as you go along. This key acts as the back-space key on a typewriter, wiping out letters in the process. Combine it with the [SHIFT] key and you can insert spaces in order to add an equivalent number of letters. To insert a large block of text, simply press the [F1] key for the Command mode, then I for Insert. Type in as much text as you wish, then press [F1] plus I again to turn off the Insert command. A variation of this command also lets you merge separate files by pressing the L key for Load.

Besides Edit, this program offers three other modes: Command (which addresses the printer), Disk, and Tape. Naturally, you need be concerned about only one of the last two choices, depending on your method of storage.

Writing in Edit mode most of the time, you will not have to keep switching from one mode to the other; even so, it is simple to change modes. Just press the [Fn] key in combination with letter keys that correspond to the wide variety of functions offered in this program.

Any function of the program can be quickly called up with one or two simple keypresses. For example, to print, press [F1] to enter the Command mode; then press O for Output and P for Print. Equally handy are the commands for formatting and for storage, once you learn them.

EasyScript's Difficulties

There are really only two significant drawbacks to this program. As mentioned earlier, the text in Edit mode breaks strangely at the right margin. The other drawback is the documentation. The manual for *EasyScript* may be complete, but it is not easy to use. As you flip through the sections, wondering how to effect this or that change in your file, you may not find the answers without reading the whole book. For example, in its coverage of formatting, the manual fails to give clear examples of the proper placement of these commands. By roaming through its pages, I eventually discovered that Format commands for spacing, margins, and justification must be placed at the beginning or end of a line. Looking for format commands in the index, I found no mention of them. In fact, the index fails to include a good many important terms.

In short, this manual is a mish-mash—it's very hard to find the basic keystrokes you need. And, there is little help from the much-too-formal decimal breakdown of each chapter and paragraph. There is also a problem in the manual's "tone"—it is encrusted with dead words such as "indicated," "facilitated," and "enabled"—all in the passive voice. A thorough revamping of the manual to better suit average users is definitely in order.

In spite of these problems with the manual, the program is well worth learning. It just takes a while to find out what it can do. A powerful program such as *EasyScript*, used with a Commodore 64, can provide you with a complete word processing system for little more than $600—depending on how carefully you shop for a monitor, disk drive, and printer. You can even use the program with a cassette player instead of a disk system, saving yourself an additional $200. Such a set-up would not be as easy to use as a disk system, but the price deserves mention as a milestone in low-cost word processing.

HCM

How To Build Your Own Cassette Cable

As products manufactured by Texas Instruments evaporate from retailers' shelves, many users fear they'll be unable to obtain necessary equipment for their computers. To assist our readers in getting the most out of their TI computers, **HCM** offers these complete instructions for building a cassette cable. To build the cable assembly, you'll need to get the parts specified in the Materials List below. Then just follow the directions.

INSTRUCTIONS

1) Cut three pairs of wires 26-1/2" long. Using nail polish or plastic model paint, distinctly mark the last inch of both ends of one pair of wires with one large dot. In a similar manner, mark the ends of the next pair of wires with two dots. Finally, repeat this operation using three dots for the last pair of wires. Allow the paint or nail polish to dry completely. This labeling lets you identify individual wire pairs when they are in the heat-shrink tubing.

2) Paint one of the red plastic, miniature plug hoods white. This jack will eventually plug into the monitor or external speaker output.

3) Locate the two 12-inch lengths of 1/8" heat-shrink tubing. Cut both lengths in half. Push all three pairs of wire through one of the 6" lengths. Stop pushing when 1 1/2" of wire is beyond the end of the heat-shrink. This end will be soldered to the D connector, and will not be used until the end of step 8.

4) Starting from the other end, thread all three wire pairs into the 12" length of 3/16" heat-shrink tubing. Slide this tubing up until it overlaps the 1/8" heat-shrink by 1/4".

5) Using a suitable heat source (a heat gun or, if necessary, a candle or cigarette lighter) carefully shrink the tubing.

6) Locate the remaining three 6" lengths of 1/8" heat-shrink. Slide a piece of this tubing over a pair of wires. When this tubing butts up against the 3/16" tubing, heat it until the 1/8" tubing is shrunk. Repeat this step for the two remaining wire pairs.

7) Cut a 1-1/2" piece of 1/4" diameter heat-shrink. Slide this piece up the cable until the 1/4" diameter heat-shrink straddles the junction of the 3/16" and the three 1/8" pieces of heat-shrink. Heat and shrink the tubing. For additional strength at this junction, step 7 can be repeated using a 2" piece of 1/4" heat-shrink.

8) Locate the wire pair marked with one dot. Trim this pair so that about 1/2" of wire protrudes beyond the end of the 1/8" heat-shrink. Slide the painted white plastic jack hood onto this pair of wires. Solder the white wire from this wire pair to the short terminal of the miniature jack. Solder the black wire to the longer terminal. At the other end of the wire pair marked with one dot, solder the black wire to contact 9 on the D connector and solder the white wire to contact 8 on the D connector.

9) Trim the wire pair marked with two dots in the same manner as you trimmed the wire pair in step 8. Slide the unpainted red plastic hood onto the wire. Solder the black wire to the long terminal of the miniature jack, and the white wire to the short terminal. At the other end of the wire marked with two dots, solder the white and black wires to the D connector's contacts 5 and 3, respectively.

10) Trim the remaining pair of wires as you did in the previous two steps. Slide the subminiature plastic hood over this set of wires. Again, solder the white wire to the short terminal and the black wire to the longer terminal. Solder the remaining white wire to contact 1 of the D connector. Solder the black wire to contact 2.

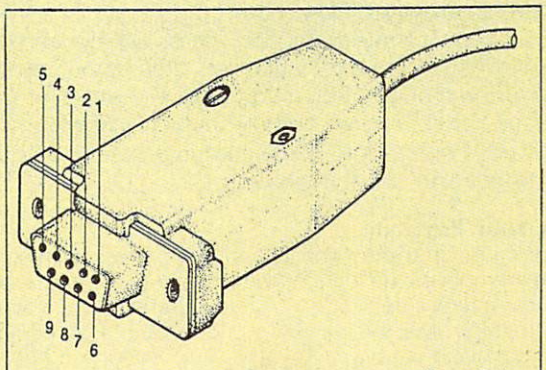
11) Install the plastic hood on the D connector. Screw the three plastic hoods onto the jacks. Plug the cable into your computer and cassette deck. The red jack plugs into the microphone input, the white jack plugs into the monitor or external speaker output, and the subminiature jack plugs into the remote input. If the remote cable does not turn on your cassette deck's motor, reverse the two wires in the subminiature jack. —**by Peter Bloch**

MATERIALS LIST

RADIO SHACK PART NUMBER	QUANTITY OR SIZE	DESCRIPTION OF MATERIAL
274-289	1	1-1/6 subminiature phone plug (male).
274-287	2	1/8" miniature phone plugs (male).
278-755	80"	2-conductor wire. Twisted pair.
276-1538	1	D-subminiature 9 pin solder type connector (female).
276-1539	1	D-subminiature 9 pin connector hood.
278-1627A	24"	1/8" heat-shrink tubing.
(Note: this one part number contains all of these sizes of heat shrink tubing, plus more.)	12"	3/16" heat shrink tubing.
	1-1/2"	1/4" heat shrink tubing.

MISCELLANEOUS

rosin core solder, soldering iron, white and red plastic model paint or nail polish.

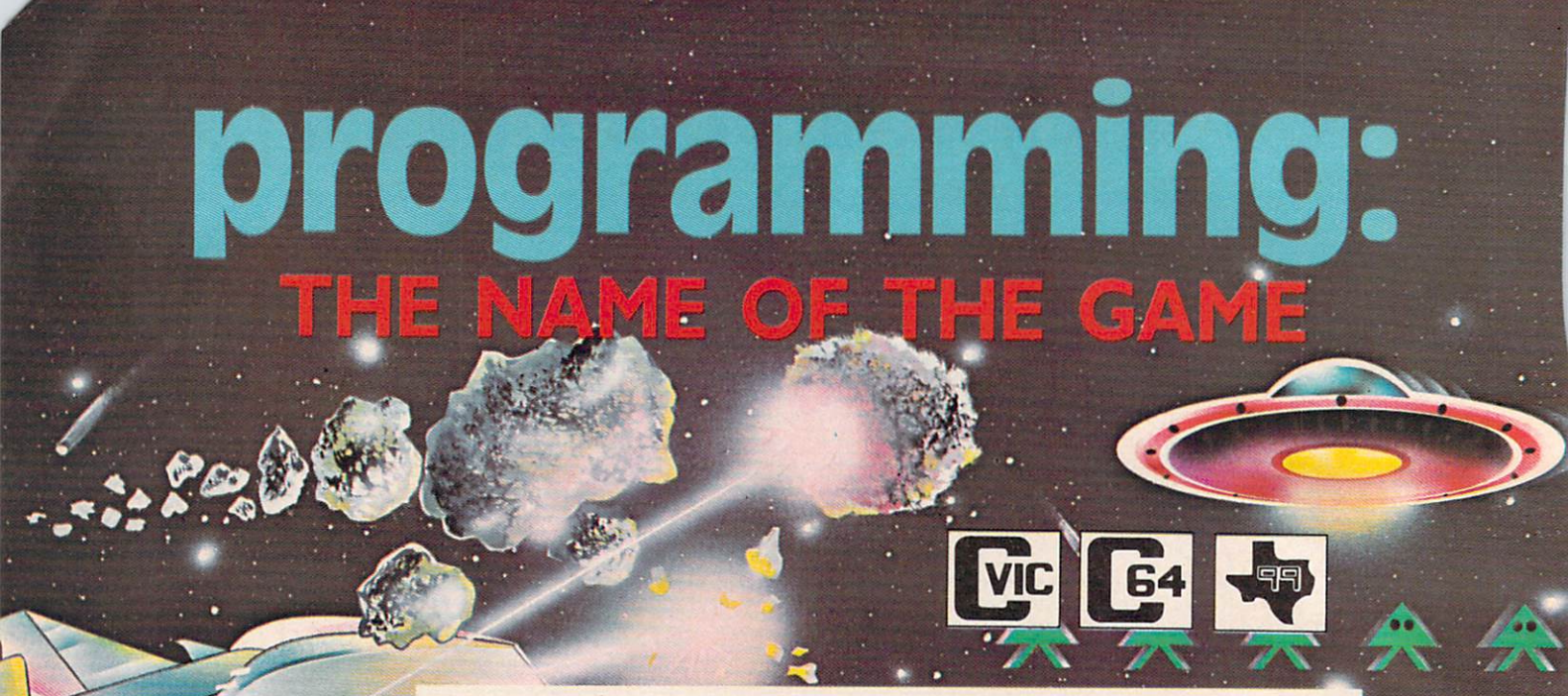


D—subminiature 9 pin connector (female)

PIN #	WIRE COLOR—PLUG
1	white—subminiature plug short terminal (REMOTE)
2	black—subminiature plug long terminal (REMOTE)
3	black—red plug long terminal (MIC)
5	white—red plug short terminal (MIC)
8	white—white plug short terminal (EAR)
9	black—white plug long terminal (EAR)

(4, 6, and 7 unused.)

programming:
THE NAME OF THE GAME



by the HCM Staff

Pssst! I've got a little secret for you, gang: *Designing* and *programming* your own game on your TI or Commodore home computer can be just as much fun as *playing* games produced by others. And best of all, it's really not as hard as you might think . . .

PART 1: GENERAL RECOMMENDATIONS

Pick A Game, Any Game . . .

You can have a maze, a game using dice, a card game, a memory-type game, a board game, a popular sport, a game involving logic, a game dependent on skills or reaction time, some form of hide-and-seek, an adventure, or a myriad of space and shooting games. Still don't have a game plan? Walk through a video arcade to get some ideas.

Let the Computer Play Too

If you write a game of Tic-Tac-Toe or Othello for two players, you're really only utilizing the computer's graphics—the game could just as well be played on paper or on a board. But, if you write the game for one person *against* the computer, you *are* using the computer to help go through a logic process. And because the TI and Commodore home computers are powerful logic tools, this is the best way to learn to use this power.

Plan Your Program

Don't just sit at the console and begin programming your game hoping that you can remember all the logic. Some programmers draw a flowchart first. On logic games you may prefer "tree diagrams"—i.e., if the player chooses one option, the program branches one way; then depending on the next choice, it branches again, and so forth. Other programmers prefer a more structured approach —each process of the game

Continued

is in a subroutine and the main program calls the subroutines in order. This type of program is easy to evaluate and easier for other programmers to follow than a program that has GOTO statements all over the place. What is best for you? Select a planning method that fits your needs, and then plan the logic before writing the program.

Look Through Your Listing

Once your program is written, there are usually still many ways to make it better. By sitting back and looking over the listing of your program's instructions, many things can be spotted. If you use the same group of lines several times, use a GOSUB, and place the subroutine near the end of the program. For example, a subroutine to read numeric key presses can be very useful.

TI BASIC

```
800 CALL KEY(0,KEY,STAT)
810 IF (STAT=0)+(KEY<48)+(KEY>57) THEN 800
820 RETURN
```

C-64 and VIC-20 BASIC

```
800 GET A$
810 IF A$<"0" OR A$>"9" THEN 800
820 RETURN
```

It is a simple matter now to access the routine many times with a simple GOSUB, as you will see in the program lines below:

TI BASIC

```
150 CALL CLEAR
160 GOSUB 800
170 PRINT CHR$(KEY);
180 GOTO 160
```

C-64 and VIC-20 BASIC

```
150 PRINT CHR$(147)
160 GOSUB 800
170 PRINT A$;
180 GOTO 160
```

Check for unnecessary statements. We have seen several listings that contain program statements that can never be executed because they have been by-passed, and subroutines that are never called. Other cases may occur because of editing. For example:

TI, C-64, and VIC-20 BASIC

```
900 GOTO 920
910 X=25
920 GOTO 980
```

or

```
900 GOTO 910
910 Z=Z+1
```

or

```
900 IF X=A THEN 910
910 GOTO 980
```

Be Sure To Include Instructions

Many players are anxious to play the game and won't read anything that comes with the game program, so it is wise to include simple instructions within your program. Players that are already familiar with the game, however, won't want instructions, so you must try to satisfy everyone. One method is to print the instructions on one screen with "PRESS ANY KEY TO START" at the bottom of the screen. The player can then look at the screen or immediately press any key to start the game.

TI BASIC

```
100 CALL CLEAR
110 PRINT "PRESS 'S' & 'D' TO GO"
120 PRINT "LEFT OR RIGHT."
130 PRINT "PRESS 'F' TO SHOOT."
140 PRINT :::"PRESS ANY KEY TO START."
150 CALL KEY(O,KEY,STAT)
160 IF STAT<1 THEN 150
170 REM PROGRAM CONTINUES FOR GAME
```

C-64 and VIC-20 BASIC

```
100 PRINT CHR$(147);"PRESS 'S' & 'D' TO GO LEFT OR
      RIGHT."
110 PRINT "PRESS 'F' TO SHOOT."
120 PRINT:PRINT:PRINT:PRINT "PRESS ANY KEY TO
      START."
130 GET A$
140 IF A$="" THEN 130
```

Another method is to inquire whether the player needs instructions:

TI BASIC

```
100 CALL CLEAR
110 PRINT "NEED INSTRUCTIONS? (Y/N)"
120 CALL KEY (0,K,S)
130 IF K=78 THEN 400
140 IF K<>89 THEN 120
150 REM PROGRAM PRINTS INSTRUCTIONS
400 REM PROGRAM CONTINUES FOR GAME
```

C-64 and VIC-20 BASIC

```
100 PRINT CHR$(147)
110 PRINT "NEED INSTRUCTIONS? (Y/N)"
120 GET A$
130 IF A$="N" THEN 400
140 IF A$<>"Y" THEN 120
150 REM PRINT INSTRUCTIONS
400 REM PROGRAM CONTINUES FOR GAME
```

If the player presses Y, instructions will be printed; if the player presses N, the game starts. Any other key pressed is ignored by the program. Be sure the instructions are as clear and concise as possible. Use enough blank lines to make the instructions easy to read. Make sure words are not divided at the end of lines, be sure to spell correctly, and use correct grammar.

Check for Speed and Captivation

You don't want the player to fall asleep between moves. If you have moving objects in your game, the player wants them to be as fast as possible. The main hints here are to have the moving objects be just one character and to minimize the logic between moves. Note that, generally, the more objects you have to move, the longer it will take.

Make Your Game "User Friendly"

This means considering all possibilities of input. You never know what some players will try to do. If you require an answer of "yes" or "no," can the players just press Y or N, or do they need to spell out the word and press [ENTER] or [RETURN]? Pressing one key has a lot less chance of error than using INPUT. What if you ask for a *number*, and a *letter* is pressed? What if you ask for a choice of 1 through 4 and the number 7 is pressed? If the player needs to use the arrow keys, is there a default value if he or she hits another key, or is that key ignored—or worse yet, does the program crash?

Test Your Game

Again, check all possibilities. If you say your spaceship can move to the right and to the left, be sure to check *both* directions. Make sure positive and negative numbers work correctly in your calculations. Check the scoring to see if it is adding correctly. Test the possibility of hitting the wrong key. Test moving objects at the edges of the screen.

PART 2: SPECIFIC HINTS

Random Numbers

Be sure to use the statement RANDOMIZE before using RND in TI BASIC or use RND (0) in Commodore BASIC so each game played will be different. If random numbers are computed at several different places, consider doing this at each occurance of RND to ensure total randomization throughout the game. Sometimes a single RANDOMIZE function at the beginning of the program does not satisfy the need. Shooting dice would need a random number from 1 to 6:

TI BASIC

```
100 RANDOMIZE
110 D1=INT(RND*6)+1
```

C-64 and VIC-20 BASIC

```
100 D1=INT(RND(0)*6)+1
```

In a space program or skill-type game you may want to place obstacles at random positions. If you have several objects, DEFine a few functions at the beginning of the program, then you can use them later in the coding:

Continued on next page

TI BASIC
```
100 DEF RX=INT(RND*24)+1
110 DEF RY=INT(RND*29)+2
120 CALL CLEAR
130 RANDOMIZE
140 FOR I=1 TO 5
150 CALL HCHAR(RX,RY,65)
160 NEXT I
170 CALL VCHAR(RX,RY,66)
180 STOP
```

C-64 BASIC
```
100 DEF FNRX(X)=INT(RND(0)*24)+1
110 DEF FNRY(X)=INT(RND(0)*40)+1
120 PRINT CHR$(147)
130 FOR I=1 TO 5
140 PP=FNRX(X)*40+FNRY(X)
150 POKE PP+1024,1:POKE PP+55296,1
160 NEXT I
170 PP=FNRX(X)*40+FNRY(X)
180 POKE PP+1024,2:POKE PP+55296,2
```

VIC-20 BASIC
```
100 DEF FNRX(X)=INT(RND(0)*23)+1
110 DEF FNRY(X)=INT(RND(0)*22)+1
120 PRINT CHR$(147)
130 FOR I=1 TO 5
140 PP=FNRX(X)*22+FNRY(X)
150 POKE PP+7680,1:POKE PP+38400,1
160 NEXT I
170 PP=FNRX(X)*22+FNRY(X)
180 POKE PP+7680,2:POKE PP+38400,2
```

The DEFinition statements must be numbered lower than the statements in which the functions are used. Lines 140-170 place five A's and one B in random X and Y positions, for X from 1 to 24 and Y from 2 to 30.

Another use of random numbers is choosing a random message or procedure. For example:

TI BASIC
```
500 PRINT A$(INT(RND*9)+1)
```

C-64 and VIC-20 BASIC
```
500 PRINT A$(INT(RND(0)*9)+1)
```

These lines choose one of nine messages previously stored in the A$ array. For random subroutines, the coding would be:

TI BASIC
```
510 ON INT(RND*5)+1 GOSUB 200,250,300,350,400
```

C-64 and VIC-20 BASIC
```
510 ON INT(RND(0)*5)+1 GOSUB 200,250,300,350,400
```

Games using a deck of cards may use an array to keep track of which cards are dealt. You may use C$(52) for the 52 cards, or a two-dimensional array C(13,4) where the first parameter is the number chosen and the second is the suit. An example for choosing ten cards follows. The values in the card array are initially zero. As a card is chosen, the corresponding C element is set equal to 1. In the following example we printed the card values, but remember you really should take advantage of the computer's graphics to *draw* the cards.

TI BASIC
```
100 REM CARDS
110 CALL CLEAR
120 DIM C(13,4),A$(13)
130 DATA ACE,2,3,4,5,6,7,8,9,10,JACK,QUEEN,KING
140 FOR J=1 TO 13
150 READ A$(J)
160 NEXT J
170 SUIT$(1)="HEARTS"
180 SUIT$(2)="CLUBS"
190 SUIT$(3)="DIAMONDS"
200 SUIT$(4)="SPADES"
210 PRINT "TEN CARDS CHOSEN:"::
220 RANDOMIZE
230 FOR I=1 TO 10
240 N=INT(13*RND)+1
250 S=INT(4*RND)+1
260 IF C(N,S)=1 THEN 240
270 PRINT A$(N);" OF ";SUIT$(S)
280 C(N,S)=1
290 NEXT I
```

C-64 and VIC-20 BASIC
```
100 REM CARDS
110 PRINT CHR$(147)
120 DIM C(13,4),A$(13)
130 DATA ACE,2,3,4,5,6,7,8,9,10,JACK,QUEEN,KING
140 FOR J=1 TO 13
150 READ A$(J)
160 NEXT J
170 SUIT$(1)="HEARTS"
180 SUIT$(2)="CLUBS"
190 SUIT$(3)="DIAMONDS"
200 SUIT$(4)="SPADES"
210 PRINT "TEN CARDS CHOSEN:":PRINT:PRINT
220 FOR I=1 TO 10
230 N=INT(RND(0)*13)+1
240 S=INT(RND(0)*4)+1
250 IF C(N,S)=1 THEN 230
260 PRINT A$(N);" OF ";SUIT$(S)
270 C(N,S)=1
280 NEXT I
```

ARROW KEYS

In games where you move a character up, down, left, or right, you may wish to have the player press the arrow keys. (The arrows are on the keys E, D, X, and S of the TI-99/4A). A CALL KEY statement on the TI machine, or GET statement on the Commodore is used to receive the player's input; the program then branches depending on which arrow is pressed. Any other key pressed should be ignored so your program doesn't crash with bad values. The following routine will draw a trail of asterisks as you press the arrow keys. Remember, you must consider the edges of the screen or you may get a BAD VALUE message. Lines 270-340 test for the edge values and will keep the asterisk at the edge position.

TI BASIC
```
100 REM MAKE-A-TRAIL
110 CALL CLEAR
120 X=12
130 Y=15
140 CALL HCHAR(12,15,42)
150 CALL KEY(0,K,S)
160 IF K<>69 THEN 190
170 X=X-1
180 GOTO 270
190 IF K<>68 THEN 220
200 Y=Y+1
210 GOTO 270
220 IF K<>88 THEN 250
230 X=X+1
240 GOTO 270
250 IF K<>83 THEN 150
260 Y=Y-1
270 IF X>=1 THEN 290
280 X=1
290 IF X<=24 THEN 310
300 X=24
310 IF Y>=1 THEN 330
320 Y=1
330 IF Y<=32 THEN 350
340 Y=32
350 CALL HCHAR(X,Y,42)
360 GOTO 150
```

C-64 and VIC-20 BASIC
**** See note about entering this program on the VIC-20.**
```
100 REM MAKE-A-TRAIL
110 PRINT CHR$(147)
120 X=12
130 Y=13
140 GOSUB 370
150 GET A$
160 IF A$<>"E" THEN 190
170 X=X-1
180 GOTO 270
190 IF A$<>"D" THEN 220
200 Y=Y+1
210 GOTO 270
220 IF A$<>"X" THEN 250
230 X=X+1
240 GOTO 270
250 IF A$<>"S" THEN 150
260 Y=Y-1
270 IF X<=0 THEN 290
280 X=0
290 IF X<=24 THEN 310
300 X=24
310 IF Y>=0 THEN 330
320 Y=0
330 IF Y<=39 THEN 350
340 Y=39
350 GOSUB 370
360 GOTO 150
370 POKE X*40+Y+1024,42:
    POKE X*40+Y+55296,1:
    RETURN
```

**** NOTE: Make the following changes if you enter this program on the VIC-20:**
```
290 IF X<=21 THEN 310
300 X=21
330 IF Y<=22 THEN 350
340 Y=22
370 POKE X*22+Y+7680,42:POKE X*22+Y+38400,1:RETURN
```

Remember, there are many ways of programming to get the same result, and the examples presented here are merely just that—*examples*. The following routine illustrates another way to use the arrow keys to move a character. This time the previous character is deleted. Also, lines 330-410 will make the asterisk scroll to the other side of the screen instead of staying at the edge.

TI BASIC

```
100 REM MOVE-A-STAR
110 CALL CLEAR
120 X=12
130 Y=15
140 CALL HCHAR(X,Y,42)
150 CALL KEY(0,K,S)
160 IF K<>69 THEN 200
170 DX=1
180 DY=0
190 GOTO 310
200 IF K<>68 THEN 240
210 DX=0
220 DY=1
230 GOTO 310
240 IF K<>88 THEN 280
250 DX=1
260 DY=0
270 GOTO 310
280 IF K<>83 THEN 150
290 DX=0
300 DY=-1
310 CALL HCHAR(X,Y,32)
320 X=X+DX
330 IF X>0 THEN 370
360 X=1
370 Y=Y+DY
380 IF Y>0 THEN 400
390 Y=32
400 IF Y<33 THEN 420
410 Y=1
420 CALL HCHAR(X,Y,42)
430 GOTO 150
```

C-64 and VIC-20 BASIC
**** See the note if you enter this program on the VIC-20.**

```
100 REM MOVE-A-STAR
110 PRINT CHR$(147)
120 X=12
130 Y=13
140 GOSUB 440
150 GET A$
160 IF A$<>"E" THEN 200
170 DX=-1
180 DY=0
190 GOTO 310
200 IF A$<>"D" THEN 240
210 DX=0
220 DY=1
230 GOTO 310
240 IF A$<>"X" THEN 280
250 DX=1
260 DY=0
270 GOTO 310
280 IF A$<>"S" THEN 150
290 DX=0
300 DY=-1
310 GOSUB 450
320 X=X+DX
330 IF X>=0 THEN 350
340 X=24
350 IF X<25 THEN 370
360 X=0
370 Y=Y+DY
380 IF Y>=0 THEN 400
390 Y=39
400 IF Y<40 THEN 420
410 Y=0
420 GOSUB 440
430 GOTO 150
440 POKE X*40+Y+1024,42:
    POKE X*40+Y+55296,1:
    RETURN
450 POKE X*40+Y+1024,32:
    RETURN
```

**** NOTE: If you enter this program on the VIC-20, then you will need to make the following changes to the above program:**

```
340 X=21
350 IF X<22 THEN 370
390 Y=22
400 IF Y<23 THEN 420
440 POKE X*22+Y+7680,42:POKE X*22+Y+38400,1:RETURN
450 POKE X*22+Y+7680,32
```

A more compact approach to automatic scrolling is to replace lines 330-360 and 380-410 with these two lines:

TI BASIC

```
330 X=INT(24*((X-1)/24-INT((X-1)/24)))+1
380 Y=INT(32*((Y-1)/32-INT((Y-1)/32)))+1
```

C-64 BASIC

```
330 X=INT(25*((X-1)/25-INT((X-1)/25)))
380 Y=INT(40*((Y-1)/40-INT((Y-1)/40)))
```

VIC-20 BASIC

```
330 X=INT(22*((X-1)/22-INT((X-1)/22)))
380 Y=INT(23*((Y-1)/23-INT((Y-1)/23)))
```

PART 3. DO IT!

We've presented some fundamental hints and ideas for programming; now it's your turn to put on your thinking cap, turn on the computer, and have fun writing your *own* games!

HCM

Want to Get Published?

Home Computer Magazine is looking for articles in all areas of interest that concern Apple, Commodore, IBM, and Texas Instruments home computers. Here are the kinds of articles that we want you to write for us:

- Are you a businessman, professional, hobbyist, scientist, or engineer with an interesting microcomputer application? Tell us how it works, what problems you've had to overcome, and what recommendations you have for others. We're especially interested in sharing user-written software with our readers.
- Have you recently purchased a piece of hardware or software that hasn't quite come up to your expectations, or has, on the other hand, impressed you with its performance? We're looking for comprehensive product and book reviews from different perspectives.
- Are you an educator or parent with something to contribute to computer-assisted instruction (CAI)? We're always looking for new ideas and fresh approaches to educational problems.
- Have you created any unusual computer games or simulations? Let our readers experience your excitement and pleasure.
- Perhaps you've modified your microcomputer or have interfaced it with some unique or useful hardware. Send us your how-to-do-it story.

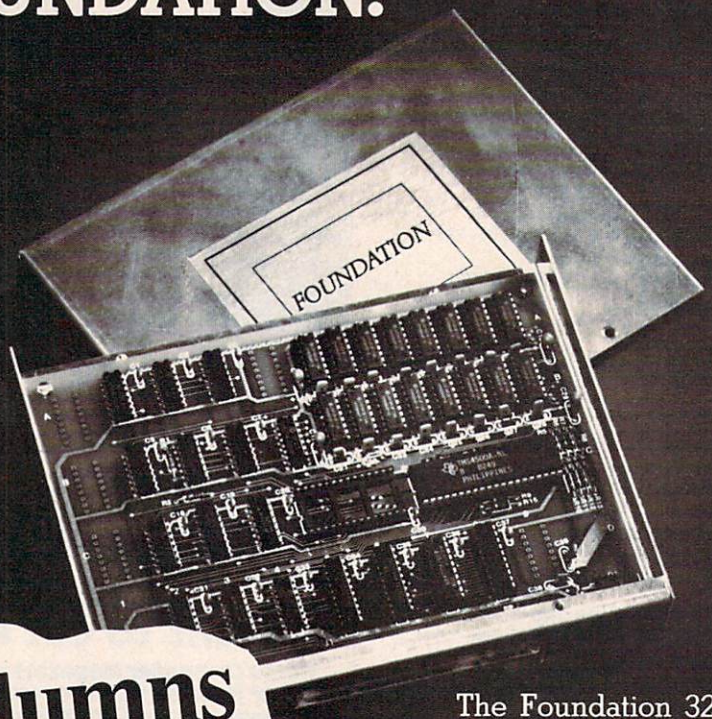
These are just some ideas. Perhaps you have others. Don't worry if you're not a professional writer. Our editorial staff stands ready to help polish up your manuscripts. And we'll be more than happy to send you a copy of our author guidelines.

Please send your double-spaced typed manuscripts, plus disks or cassettes (recorded on both sides) if the article includes program material, to:

Home Computer Magazine/Editorial Dept.
1500 Valley River Drive, Suite 250
Eugene, OR 97401

NEED MORE MEMORY?

ORDER THE 128K MEMORY CARD BY FOUNDATION.



80 Columns on the 99!

Tiburon. June 29, 1984.

Home computing took another step closer to professional quality today as Foundation Computing unveiled a full-fledged intelligent terminal card that plugs into the TI 99/4 expansion box. Providing a 24x80 display with a 25th line of status information, the $249 card includes its own internal computer and provides features not available on many $1500 display terminals. Any combination of reverse video, half-intensity, double width, double height, and blinking characters can be shown. A complete set of display control codes and line drawing characters are provided. Contacted for comment, Foundation VP Kathy Hunter stated that the product was developed in direct response to customer requests. Expected uses include word processing, telecommunications, and report layout. She also stated that for many purposes, the card can replace the use of an RS232 card.

The Foundation 32K and 128K Memory Cards are designed from the beginning to be compatible with the TI99/4A Computer. The 128K Card comes with solid-state software that makes it easy to use the additional memory. Just write or call us and we'll send you a descriptive brochure.

The choice is yours. Order now and take advantage of Foundation's Special Introductory prices.*

The Foundation 32K Memory Card - $150.
The Foundation 128K Memory Card - $230.
Both with a full three month warranty.

FOUNDATION COMPUTING
74 Claire Way
Tiburon, CA 94920
415/388-3840

*Add 3% shipping and handling charges. California residents add 6% sales tax. Allow 2-4 weeks for delivery.

Circle 157 on Reader Service Card.

COMPANION

A review by Judy Sanoian
and the HCM Staff

The perfect word processor should free you to type in an article or story as it enters your head, without having to worry about carriage returns, word breaks, [ENTER] keys, page numbers, and all the other "secretarial" details that can muddle up the creative process. So claim the Intelpro people, designers of the *Companion* word processor. In developing their program, their goal was to help the writer create text on a "conceptual, rather than secretarial, basis."

Using *Companion* for the first time, this philosophy is apparent. There are literally no distractions from the writing process. The screen is noticeably free of line numbers and other clutter. Any clues that formatting will take place seem carefully hidden. Words break wherever they hit the end of the line. The margins are set automatically. Editing commands are easy to use. You are indeed free to type in exactly what enters your head—no carriage returns, margin-setting or other formatting chores are needed...not, at least, until you are ready to print out your copy.

Different Priorities

In reviewing *Companion*, it is inevitable that we compare it with *TI-WRITER*, probably the best-known word processor for the 99/4A. We found more similarities than contrasts between the two systems, but there are subtle differences—evidence that the programs were designed according to two different philosophies. For example, *TI-Writer* strives to depict the text on your screen as it will appear in the printout (if you use the PRINT FILE command in edit mode). Words are never broken at the end of the line. Tabulated material appears on screen exactly as you will see it in the printout. Copy that will be printed out in 80-character lines can be seen on screen in 80-character-wide format.

With *Companion*, on the other hand, you must take it on faith that your tabs, word breaks, and other formatting will appear correctly in the printout. Although words are kept intact on the printout, there is no on-screen word wrap. Its creators say that this is to permit every bit of the display space to be filled with text.

Name:	Companion
Program Type:	Word Processor
Machine:	TI-99/4A
Distributor:	Intelpro 5825 Baillargeon Street Brossard, Quebec, Canada
Price:	$79.95, diskette
System requirements:	Disk drive, Extended BASIC, 32K memory expansion

	Poor	Fair	Good	Excellent
Performance				
Ease of Use				
Documentation				

So *Companion* lets you see a bit more text at one time than *TI-WRITER*. This does not, incidentally, have anything to do with the character capacity (the amount of text a file can hold) of either system. In fact, *TI-WRITER* (which does have on-screen word wrap) has a greater character capacity (23000) than *Companion* (18000), but these figures are irrelevant anyway since both systems let you link together files to allow virtually unlimited character capacity.

Similarly, *Companion* limits you to a 40-character-wide screen display but lets you print out your text with up to 132-character-wide margins. The manual touts this screen-wide margin limitation as the best compromise for clarity and efficient use of space. I can't quite go along with their logic, especially when I compare *Companion*'s capabilities with what *TI-WRITER* has to offer. *TI-WRITER* lets you see on the screen exactly how your text (up to 80 characters across of it, anyway) will be formatted. Since the 99/4A monitor displays only 40 characters across at a time, *TI-WRITER* uses a horizontal scrolling "window" to let you see the rest of the line. *Companion*'s designer is strong in his condemnation of horizontal scrolling. In the manual he characterizes it as fatiguing and confusing—a practice which should be "banned as a health hazard." While there may be some truth to his remarks, we must keep in mind that *TI-WRITER*'s horizontal scrolling option is just that—an option. You can always set your margins at 39 for viewing, then change to a wider margin before you print your copy.

I should also mention that the current version of *Companion* (2.0) offers no right margin justification. There are, however, parameter codes set aside for offering that capability on a future version of the program. This new version—which will offer several additional features—will be available for under $30 to owners of the present *Companion*.

Editing Ease

Companion's editing commands were also designed according to the "ease of use is paramount" philosophy. For example, when you use the INSERT command, it displays the inserted material and moves the rest of the text simultaneously. With *TI-WRITER* (in the edit mode) the line drops down when you insert the material, and you must press [CTRL][R] to close the line back up—a two-step procedure. In the fixed mode, you can insert material in one step, but only up to the end of the line. After that, your text will start dropping off the line, into the void. *Companion* also provides the usual word processing editing commands: delete, center, skip a line, midline (for letter salutations). It also has a REDO command that zips you to the last line of the text. This is very handy if you want to go back and view text, then return immediately to where you were writing. You can also do this with *TI-WRITER*, but it is a more lengthy (three-step) process.

While I'm on the subject of speed, I should mention *Companion*'s accelerating auto-repeat cursor motion. This means the longer you scroll, the faster your cursor goes. *TI-WRITER*, with its one-speed cursor (and its lack of an equivalent to the REDO command), must take second place to *Companion* in the "cursor cruise speed" competition.

Printout Parameters

Companion recognizes 11 "printout parameters" that determine the physical arrangement of the page. Among them are the following: form length and width, left and right margins, top and bottom margins, spacing between lines, tabs, and paragraph indentation. Each of these

Continued on page 160

Using Batch Files

You can design you own IBM DOS commands with **batch files**. DOS expects whatever you enter at the keyboard to be a command which it can look up either in **memory**—an **internal** command such as **TYPE**, **DIR**, and **COPY**—or **on disk**—one of 21 **external** commands, such as **DISKCOPY**, **SORT** or **FIND**. (The file it is looking for will contain a set of instructions for carrying out your command.)

When DOS does this search of the disk, it looks for a file with an extension of COM, EXE, or BAT, in that order. In other words, if you enter LUMP at the keyboard, it would look for LUMP.COM, LUMP.EXE, then LUMP.BAT. COM and EXE are two formats for machine language files, and the third extension, BAT, tells DOS that the file is written in text characters—just like you enter at the keyboard. BAT stands for **batch file**.

Put any valid combination of DOS and batch **sub-commands** into a BAT extension file, then enter the name of the file (with no BAT extension), and DOS will execute the commands in the file.

Creating Batch Files

DOS provides a difficult-to-use text editor called EDLIN for creating batch files, but you can make them with any text editor that you might have. Make them from BASIC, or make them in DOS by using the copy command like this:

COPY CON: A.BAT [ENTER]

This tells DOS to put what you type at the keyboard (the CONsole) into the file A.BAT until it receives an end-of-file character from you. Press [ENTER] after each command as you enter the batch file contents. Each command in a batch file must be on its own line. Now you can enter:

```
MODE 80
BASICA
```

After the last line, press [FN] 6 if you have a PCjr, or [F6] if you have a PC. ^Z (the end-of-file mark, ASCII character 26) will be displayed on the screen. Press [ENTER], and you will be back in DOS, and A.BAT will now be in your disk directory. You can look at the file contents again with the command TYPE A.BAT. If you now enter the command A, DOS will change to 80-column mode and bring up BASICA.

The batch sub-commands **ECHO**, **FOR**, **GOTO**, **IF**, **SHIFT**, **PAUSE**, and **REM** help control processing within the batch file. See pages 2-13 to 2-32 in the IBM DOS 2.10 manual or 6-28 to 6-49 in the IBM DOS 2.00 manual for further detailed information.

AUTOEXEC.BAT

When DOS starts up, it will immediately search for and execute a special batch file called AUTOEXEC.BAT. AUTOEXEC.BAT doesn't exist until you create it. Put in commands to set the screen mode, start BASIC, or run your favorite program. AUTOEXEC bypasses the normal DOS date and time query, so you need to include TIME and DATE in the file if appropriate. See the example below:

```
A>TYPE AUTOEXEC.BAT
DATE
TIME
MODE 80
BASICA
```

— **Robert Keller**

Color On The Hi-Res Screen

The Apple's memory is divided into "byte addressable" locations—each byte is made up of 8 bits. The hi-res display is "bit mapped"—that is, each dot on the screen corresponds to a bit in the Apple's memory. Two sections of memory are devoted to displaying hi-res graphics: page 1 and page 2. Each page is 8192 bytes long.

Page 1 is positioned at memory locations 8192 to 16383 ($2000 to $3FFF). On a monochrome display, if a bit is a 1, then its corresponding dot is turned on (white). If it is a 0, it is turned off (black). Only 7 of the bits in any byte in hi-res memory represent a dot—the high-order (left most) bit in any byte selects the color of the bits in that byte.

The Applesoft **HGR** statement displays the page and resets every byte in that area to a 0, clearing the screen to black. You can turn on bits by using the **HPLOT** statement. Identify the locations on the screen by dot-column and dot-row, and the computer will turn on the appropriate bits in memory. The color of a dot depends upon three things:
1) The condition of the high-order (left-most) bit in the byte controlling the dot.
2) The column where the dot appears on the screen. (Orange and green only appear in odd-numbered dot-columns, and blue and purple in even-numbered columns.)
3) The condition of the dots on either side of the dot. (If any two adjacent dots are on, they both appear white—in fact, this is the only way to obtain white on a color hi-res screen.)

When you specify the color with the **HCOLOR=** statement, you determine the condition of the high-order bit of the next byte (or bytes) displayed. If you specify 0 through 3, then the high bit is cleared to 0; if you specify 4 through 7, it is set to 1. It helps to picture the screen divided into 40 columns of seven dots each.

Within each of these 7 dot-columns you can specify **only** those colors that have the **same** high-order bit. It's easy to determine which dots are in the same byte—divide the dot-column number in the **HPLOT** statement by 7, and obtain the integer of the result. The dot-column numbers that result in the same integer values are controlled by the same byte. For example, dot-columns 8 and 12 are both controlled by the same byte because **INT(8/7)=1** and **INT(12/7)=1**.

Here's a list of the colors, with their corresponding **HCOLOR** numbers, where the high-order bit is 0:

0=Black1 1=Green 2=Violet 3=White1

These are the colors where the high-order bit is 1:

4=Black2 5=Orange 6=Blue 7=White2

Notice that there are two different blacks and whites. The only difference between the two colors is that Black1 and White1 occur in bytes with the high-bit cleared, whereas Black2 and White2 occur in bytes with the high-bit set. Using the wrong black or white in the same byte with a color from the other group will change the colors in that byte.

Here we draw a vertical blue line in column 8 of the high-res screen:

```
100 HGR : REM CLEAR AND DISPLAY HIGH-RES PAGE 1
110 HCOLOR= 6 : HPLOT 8,5 TO 8,100 : REM SET COLOR TO BLUE AND DRAW LINE
    FROM ROW 5 TO ROW 100 IN COLUMN 8
```

Adding this next line, which should simply draw a purple line in column 12, provides some unexpected results:

```
120 HCOLOR=2 : HPLOT 12,5 TO 12,100 : REM DRAW PURPLE LINE FROM ROW 5 TO
    ROW 100 IN COLUMN 12
```

The program draws both lines purple because both these lines are controlled by the same bytes in memory. However, blue is drawn when the high bit is on, and purple is drawn when the high bit is off. Although line 110 sets the high bit, line 120 resets the bit to 0—changing the blue line to purple. By moving the purple line over two pixels to dot-column 14, a bit controlled by a different byte, both lines will be in their intended colors. Your programming strategy must take into account Apple's unique structural approach to color graphics.

—Roger Wood

VIC

Colorfun: Learning Made Easy

by Ted Martino
and the HCM Staff

I can barely remember my pre-school days—but I know they were full of learning. What a pre-schooler learns soon becomes the background for later memories, while the original lessons fade away. I do recall that flash cards and coloring books played a big part in my pre-education, as did memorization, repetition, and more than a little coercion from concerned adults. Although these techniques are not outdated, something new has been added to the learning process—the home computer. And I feel like a child again as I enter the electronic age with my little VIC-20.

The VIC-20, without peripheral support, is limited in its scope of software applications. However, *Colorfun*—a game for the unexpanded version of the computer—offers plenty of entertainment for youngsters, while teaching fundamental word/color association skills. This program is intended to be a child-parent collaboration because of the screen-displayed prompts and required responses that are beyond the scope of pre-schoolers. Yet the game is *not* overly complicated for the toddler and it has enough action to hold the attention of a curious young mind.

What's This Color?

Subsequent to loading *Colorfun* from disk or cassette, type RUN and the title display will appear. Any input from the keyboard brings the message LEVEL 1 or 2? to the screen. After entering the level number, the next prompt requests an input for the name of the player. This gives the child practice in typing his or her name with the satisfaction of seeing it displayed on the monitor.

Following this input is a salutation using the name, and the question WHAT'S THIS COLOR? with a sample color swatch next to it. The swatch is randomly chosen from a list of five different colors, programmed so that the colors will not be exhibited in the same sequence.

Below the prompt are two choices; each contains the name of a color with a corresponding number, 1 or 2. On Level 1 a color swatch is included next to each color name, so a small child can use color matching to aid in choosing the answer. Level 2 eliminates the matching swatch so the selection is made by name alone. On either level, the selection is made by typing the number 1 or 2.

If the player's response is incorrect, the statement WRONG, (NAME), TRY AGAIN will be displayed with the color swatch until the right answer is entered. When the child selects the correct answer, the screen will clear and a message of congratulations (1 of 12) using the child's name will appear. A sound effect and a flashing screen border will also reward the participant.

If ten correct answers in a row are chosen on Level 1, the game advances to Level 2 automatically. An incorrect response, however, sets the game back to Level 1. At any time during the color swatch display, the player has the option of ending the game or allowing a new player to enter it. When either option is chosen, a score of the last player's total number of correct answers from a total number of tries is displayed.

As we went to press we noticed a small bug in line 340 in the program. It should read:

340 B = VAL(B$) . . .
instead of:
340 V = VAL(B$) . . .

COLORFUN (VIC-20) Explanation of the Program

Line Nos.	
100-170	Program header.
180	Set screen and border color; disable run/stop key (safety feature).
190	Variables for color bars of the title display.
200	Omits white as a variable for Y.
210-230	Y variable denotes color of graphics at top of screen.
240-250	Color bars at bottom of screen.
260-340	Title display; first prompt;
350	prompt input with salutation display.
360-410	Random choice of 5 colors; selects any color other than the previous input.
420	Verifies choice and sends it to appropriate color designator.
430-520	Color designators.
530	Checks second color choice and sends it to appropriate color designator.
540-640	Color designators.
650	Print statement for top of screen.
660-710	Question prompt input.
720-760	Response check; wrong answer statement.
770-850	Correct answer subroutine.
860-910	Summation total of correct answers versus total questions asked.
920-940	Return to line 360.
950-960	Data statements.

For the Key-in listing refer to the Contents of HCM PROGRAM LISTINGS on page 67.

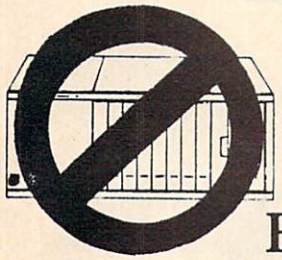
TEXAS INSTRUMENTS COMPUTING PRODUCTS AUTHORIZED SALES

TEX-COMP™

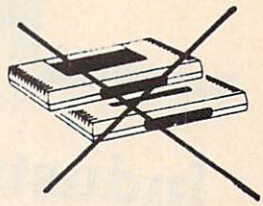
America's Number One TEXAS INSTRUMENTS dealer.

AUTHORIZED DEALER

Proudly Introduces



A COMPLETE LOW PRICE WORD PROCESSING SYSTEM



For the Texas Instruments Home Computer

TEX-COMP NOW OFFERS THE FIRST PROFESSIONAL QUALITY WORD PROCESSING AVAILABLE FOR THE TI-99/4AA EQUIPPED WITH ONLY A CASSETTE RECORDER. THE MINI-WRITER CASSETTE PROGRAM MAKES USE OF THE EXTRA MEMORY POWER OF THE TI MINI-MEMORY COMMAND MODULE . NOW YOU HAVE THE FULL POWER OF PROFESSIONAL WORDPROCESSING WITHOUT THE NEED FOR A DISK DRIVE OR MEMORY EXPANSION. BY ADDING THE NEW AXIOM PRINTER WHICH REQUIRES NO RS232 INTERFACE, LOW COST QUALITY PRINTING ON PLAIN PAPER (NOT SUB-STANDARD THERMAL PAPER) CAN BE COMBINED FOR A COMPLETE WORD PROCESSING SYSTEM AT A VERY MODEST COST.

MINI-MEMORY
$74.95

No More Erasing, Strikeovers or Messy Correcting Fluids!

MINI-WRITER
$19.95

DESCRIPTION: A "MINI" WORD PROCESSOR WHICH FEATURES:
FULL SCREEN TEXT EDITING: 24 X 40 CHARACTER "WINDOW", 80 CHARACTER LINE;
9500 CHARACTERS STORED PER FILE, ADD/DELETE CHARACTERS OR LINES;
MOVABLE COPY; UPPER AND LOWER CASE CHARACTERS;
SAVE OR LOAD TO OR FROM ANY STORAGE DEVICE; PRINT VIA THE PRINTER;
SEARCHES FOR DESIRED INFORMATION;
SCROLLS SCREEN (UP, DOWN, LEFT, RIGHT); PURGES TEXT BUFFER AND HAS CANCEL COMMANDS.
TI-MINIWRITER IS DESIGNED TO LOAD FROM CASSETTE INTO THE MINI-MEMORY CARTRIDGE.
MAKES REPORT AND LETTER WRITING EASY AND FAST.
LOW COST TO CONSUMER FOR WORD PROCESSING CAPABILITY
EASY TO LEARN AND USE

Uses standard width fan-fold paper

Two-year warranty

Graphics capabilities

$249.95

AXIOM
Direct-Connect Printer, GP-100TI
(no interface required)

Powerful Yet Simple To Use

SYSTEM TOTAL $344.85

SPECIAL BUY THE ENTIRE SYSTEM (3 PIECES) AND WE INCLUDE FREE SHIP&INS(UPS ONLY), A FREE TI DUAL CASSETTE CABLE AND A STARTER PACK OF PRINTER PAPER. A $40.00 + RETAIL VALUE!

ADD 3% S,H&I &4.5% W OF MISS ($3.00 MIN).ALL ORDERS SUBJECT TO CURRENT AVAILABILITY. WE RESERVE RIGHT TO LIMIT QUANTITIES. SEND $2.00 FOR OUR CURRENT CATALOG AND ORDER KIT. A $5 CERTIFICATE IS INCLUDED. VISIT OUR WAREHOUSE OUTLET. CALL 818-366-6631 FOR HOURS.

Send Orders To:

TEX-COMP™
P.O. BOX 33084
GRANADA HILLS, CA 91344

MasterCard VISA

VISA and MASTERCARD HOLDERS CALL DIRECT:
(818) 366-6631

ALL PRICES REFLECT A 3% DISCOUNT FOR CASH
ADD 3% IF PAYING BY CREDIT CARD

NOTE: Payment in full must accompany all orders. Credit-Card, Company Check or Money Order for immediate shipment. Personal checks require up to 4 weeks to clear. California orders add 6½% sales tax.

Circle 158 on Reader Service Card.

Excerpts from the

HOME COMPUTER™ DIGEST

News and Happenings in the Home Computer World

Industry Watch

COMPATIBLE COMMODORE

A new 256K computer in the Commodore works—recently previewed in Europe—threatens to unseat other popular IBM compatibles. Lately the company has run a few products up the flagpole—only to pull them back down before anyone could salute—but initial reports from the Hanover Fair in Germany sound promising. Based on a design licensed from the Canadian firm BytecCommterm, this model is expected to be aggressively priced against the Compaq and the Columbia—as well as the IBM PC itself. Commodore has also reportedly leaked behind-the-scene previews of its new 16-bit Z8000-based computer. Rumored to be ready for shipment by the end of 1984, it is said to feature 256K RAM, two 1.3 megabyte disk-drives, a 15-inch color monitor with extremely high (640 x 400 dot) resolution, and sport a Commodore-specific version of Coherent (a Unix Version VII-compatible operating system from the Mark Williams Company). Meanwhile, Commodore is releasing its previously announced model 264 as the Commodore Plus/4 with integrated productivity software in ROM. Look for a retail price in the $300 range. A 16K machine—replacing the VIC-20—is also in the works, filling the $100 product needs of the mass merchandizers.

HOME COMPUTER U.

Although there is a great deal of educational software for the home computer, none of it will get you a degree—or even academic credit. Now, a new system links established educational institutions with the home-bound student. TeleLearning Systems Inc. has teamed up with a half-dozen schools—including Ohio State University and the University of Nebraska—to offer some of the world's first for-credit computer correspondence courses. With the basic set-up retailing for $99, and courses priced between $45 and $125, many who are limited in either finances or mobility may soon take part in this new form of higher education. An increase in the number of teaching positions, and, of course, more demand for home computers may result. Several universities are jumping on the electronic band-wagon with their own experiments in this field. With TeleLearning, students study at their own pace, take exams, and hold conferences with professors—all by a time- and cost-cutting method of batched data transmission over existing telephone lines.

IBM LOWERS PRICES TO MEET IIc CHALLENGE

Can the Apple IIc knock out the PCjr? Apple's new machine has apparently captured the first few sales rounds with its wealth of existing software and relatively low price. IBM has counter-punched by dropping prices on their entire line of personal computers and add-ons. The unexpanded PCjr is now under $600 and the enhanced version (128K of memory and one 360K disk-drive) is less than $1000—undercutting the IIc's $1295 suggested retail. IBM has also introduced a PC with 256K of memory and one disk drive for under $2000. Big Blue has also hinted that it will upgrade PCjr to be able to run more software designed for the PC, and equip it with a better keyboard. Apple, on the other hand, may be somewhat trapped by its large software inventory—an illusory asset that could be made obsolete by any major technical improvements in the Apple II family.

NEW RAM CHIPS PROMISE MORE MEMORY FOR HOME MARKET

Until recently, it took 32 chips to obtain 256K of RAM—soon it will only take 8. U.S. chip manufacturers are readying volume shipments of the new 256K dynamic random-access memory chips for delivery by the last half of 1984. Although this won't have an immediate effect upon the home market, it does promise "more memory for less money" in the not too distant future. Volume production of the new chips is already under way in Japan, and American companies are hurrying to keep pace. While no American firms are close to matching the Japanese output at present, Motorola is hoping to be in full production by fall. Texas Instruments reportedly has its Miho, Japan plant already producing 100,000 chips per month, with a target of over 600,000 by the end of the third quarter. Meanwhile, IBM may be a jump ahead as they recently announced the development of a 1000K chip that was (according to an IBM spokesman) "fabricated on the same manufacturing line that produces . . . other chips, so no additional technology was necessary."

Continued next page

Excerpts from the

HOME COMPUTER™ DIGEST

News and Happenings in the Home Computer World

Industry Watch

EIA TO DEVELOP STANDARD HOME BUS

The Electronic Industries Association (EIA)—the people who brought us the RS232-C standard for serial communications—is hoping to develop a standard interface bus for home electronics products. If you want to hook-up your computer to your VCR, or your robot, you don't want to spend weeks looking for the proper cables and interface boards. The EIA hopes that a standard bus will make such operations much easier. Of course, developing any sort of standard may be nearly impossible. Considering the number of consumer electronic products coming from Japan these days, the EIA might be well advised to consult with Tokyo on interface protocols. Any standard bus might require a Japanese "driver" if it doesn't want to get lost.

THIRD-PARTY HARDWARE HELPS jr GROW

Thanks to third-party developers, it appears that the PCjr can be made virtually as powerful as the PC—at a lower cost. Legacy Technologies of Lincoln, Nebraska made an early entrance with its Legacy expansion unit, giving the PCjr a second disk drive, up to 512K of memory, and an 80-pin expansion bus for future development. Falcon Industries (Kent, WA.) is now introducing a "jr Extender System" which it boasts gives the PCjr near-PC capability for $634 less than the PC. Meanwhile, Tecmar of Cleveland, Ohio is marketing a package that is the same physical size as the IBM-parallel printer port, but includes 128K extra memory (expandable to 512K), a battery-backup date-time clock, a parallel port, as well as an easy-to-use full software-support package.

VISICALC EXPANDS MARKET TARGETS

Determined to shed the market perception that they are a single-product company, VisiCorp has introduced **FlashCalc**, a high-performance spreadsheet program that is expected to lead a string of VisiCorp products into the low-end market tier. Though the program features enough power to attract business users searching for a single application package, **FlashCalc** is directed toward middle-market home users. The spreadsheet will run on the Apple IIc and other Apple II family systems, and is compatible with ProDOS. VisiCorp officials said the introduction of this $99 program (and a corresponding price cut on the Visi series), positions their product line at lower price points —offering more performance for middle-range products.

ALL ABOARD FOR COMPU-CRUISE

For those who find the hustle-and-bustle atmosphere of traditional computer shows and seminars a bit unnerving, consider this alternative: a leisurely computer conference and seminar conducted during a seven-day Carribean cruise to exotic ports. Sponsored by Compu-Cruise, Inc. of Eugene, Oregon, the seminars are directed toward (1) those who use or purchase personal computer products, and (2) those involved in the marketing and sales of personal computer products. The first trip is scheduled to set sail in mid-December—with plenty of computers and computer products on board, as well as notable speakers from the computer industry. Compu-Cruise will be producing shipboard conferences four to six times a year, with other destinations including the Mexican Riviera and Alaska.

COMMODORE AND IBM TO MAKE INTEL'S 8088 CHIP

Both Commodore International Ltd. and International Business Machines Corp. have signed agreements with Intel Corp. to produce the high-in-demand, short-in-supply 8088 microprocessor. Industry analysts say the license will give IBM more control over production of this chip—a component used in the IBM PC and by some 60 competitors that make similar machines to run PC software. The Intel agreements will significantly increase production of the 16-bit 8088 microprocessor, but will have little effect on quantities of chips going to makers of IBM compatibles. An Intel spokesman said that Commodore is expected to make the chip for its own use, while IBM will be able to triple shipments of its popular personal computers.

HCD

HOME BUDGET, jr.

A review by W. K. Balthrop
HCM Staff

The *Home Budget,jr* package is a personal accounting program designed to help balance almost any budget. With it, you will be able to keep an active accounting of your income and expenses. The program is thorough enough to satisfy a small business budget, yet simple enough for home use.

The *Home Budget,jr* requires a dual-sided disk drive, with DOS version 1.10 or higher; 128K or more of RAM memory; and Cartridge BASIC if running on the PCjr. This program will also work with the PC and PC XT with BASICA (Advanced BASIC). You can use a printer for hard copy reports.

The Program

The program is easy to execute. You must first boot the system up with DOS. While under DOS, insert the program disk and enter G, which will load and run the program. The first time the program is run it will automatically ask you a few questions about your system and how you want it set up. You can select either a color or black and white display, and can adjust the screen position of the display up, down, left, or right. This is extremely handy when using a television which may offset everything to one side of the screen. The system also provides a small color chart to help you adjust your display's color. You only need to go through this procedure the first time the program is run. Thereafter, it leads you directly to the IBM logo screen where you input the current date. After that, you proceed to the master menu screen. Now, you can finally enter one of six modes to create and manipulate your personal budget data base.

The first mode you select will be Mode 5, which lets you Create or Change Budget. When you set up your budget you can create up to 48 different accounts and then assign a budget value to each account. The INCOME account is created automatically, so you may not use INCOME as one of your account titles—although you can create other accounts which serve as income to the budget. You can label each account with a heading such as FOOD for food expenses, AUTO for automobile expenses, or INTEREST for interest income.

Name:	Home Budget,jr.
Program Type:	Home Finance
Machine:	IBM PCjr, PC, or PC XT
Distributor:	IBM Software P.O. Box 1328-s Boca Raton, Fla. 33432
Price:	$45, diskette

System Requirements: Color monitor, or TV, 128K of memory, one double-sided disk drive, Cartridge BASIC on PCjr, BASICA on PC/XT, and optional printer

	Poor	Fair	Good	Excellent
Performance			■	
Ease of Use				■
Documentation				■

Data Entry

Now you are ready to start entering information into each account by selecting option 1 from the main menu. This information will be your actual expenditures and income. This is where *Home Budget,jr* really shines compared to many other budget programs: The program lets you enter each item into the account with a description of the item, a three-character note useful for searching, the value of the item, and the month and day of the transaction. Many other programs allow you to enter only a gross value for each category or account. *Home Budget,jr* lets you keep every transaction in the file, no matter how small it is.

After entering each transaction you must tell the file what kind of transaction it was. This section has six options that declare the type of transaction: INCOME, EXPENSE, REFUND, CHARGE, CHARGE ACCOUNT PAYMENT, and MULTIPLE ACCOUNT ENTRY. In addition, you can go to a HELP screen for more information, or clear from memory the entry you just made.

The INCOME, EXPENSE, and REFUND categories are self-explanatory. The CHARGE, and CHARGE ACCOUNT PAYMENT categories allow you to not only keep track of payments, but also of total charges. This way, an unpaid balance can be calculated. MULTIPLE ACCOUNT PAYMENT is used when you have an item that belongs in more than one account. For example, if you charge gas for your car, you may want to keep the total in both the AUTO account and the GAS-CA account for the gas charge card. The -CA indicates that the account is a charge account. When totals are accumulated, the computer will know it is a multiple entry and will not dock you for it twice.

Record Maintenance

Maintaining your records is one of the most important functions of this program. The *Home Budget,jr* program handles this task with ease. From the master menu you can select option 2 to Change and Review Accounts. Once you have selected this option you are asked to enter the desired account. The computer then asks you if you want to make changes, or simply review the information. If you choose to review the account, all of the account entries are displayed on the screen, six at a time. You can, of course, go back to the last menu screen at any time during the

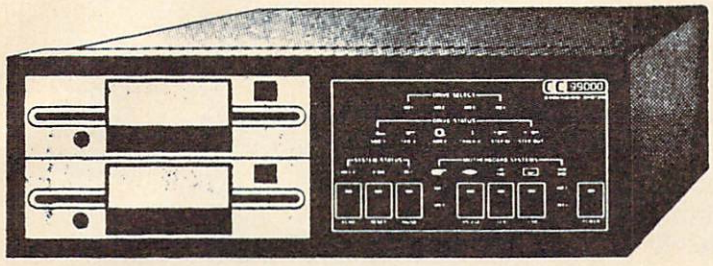
Continued on page 160

!! ATTENTION DEALERS !!
AND DISTRIBUTORS

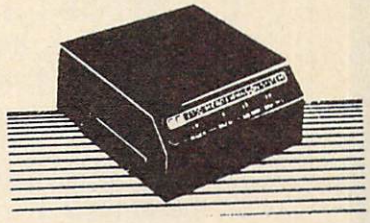
Experience the incredible market demand for the high tech peripheral line from

CorComp

THE MOST ADVANCED TECHNOLOGY EVER OFFERED FOR THE TI HOME COMPUTER



THE ULTIMATE-99000 FULL EXPANSION SYSTEM



9900 Micro-Expansion System

We now stock TIBBS(tm), the world's most popular TI operated bulletin Board System.

FEATURING TI-COUNT

TI-COUNT

1. GENERAL LEDGER
2. ACCOUNTS RECEIVABLE
3. ACCOUNTS PAYABLE
4. INVENTORY SYSTEM
5. PAYROLL SYSTEM
6. MAIL SYSTEM
7. OTHER_____

THE ONLY TRUE BUSINESS SOFTWARE FOR THE 99/4A

1-800-845-8380

FROM AMERISOFT INTERNATIONAL

John E. Brown

COPY-CAT AND GRAPHICS GRABBER

THE WORLDS FINEST HIGH SPEED DISK COPY AND DOUBLE SIZE ASSEMBLER BASED GRAPHIC DUMP UTILITIES

CR DISTRIBUTING

P.O. Box 2168
Acworth, GA 30101
(404) 928-8791

TEAC-STAR MICRONICS-HAYSE-TYMAC-PROSTICK-

Circle 159 on Reader Service Card.

Using The WEDGE

The manual that comes with the Commodore VIC-1541 disk drive refers to the **DOS Support** programs on the demo disk supplied with the drive. On the demo disk, these programs are actually called the **C-64 WEDGE** and the **VIC-20 WEDGE**. Due to a general lack of documentation, however, the inexperienced C-64 or VIC-20 user might well be mystified by these programs.

The WEDGE programs are short-cuts to disk access. They save time by cutting down on the number of keystrokes you need to make to accomplish a particular disk-related operation. Some commands just save a line of typing—others save writing whole programs.

Both the **C-64 WEDGE** and **VIC-20 WEDGE** reside in memory safely outside the BASIC programming area. To "install" one of these programs, simply load and run it. Once installed, the program becomes "hidden" and looks for certain keystrokes which act as links to activating it. The program remains active until you either turn off the power on your computer, or type @**Q**—which "Quits" or disables the WEDGE until it is loaded and run again. Because the WEDGE is set up to work with device number 8, if you are using two or more drives (designated by numbers 9, 10, or 11) its commands will **only** be directed to device 8 and not to any of the additional drives (designated by numbers 9,10, or 11).

The @ command becomes your major link with Commodore DOS once the WEDGE is installed. For example, when writing a program and accessing the disk, let's say the red light begins blinking, indicating a disk error. You'd like to know what error occurred, but without the WEDGE there isn't a **simple** way to read the disk drive's status: You'd have to enter and run a three-line program to get the status—a time consuming process, and prone to losing part of the program resident in memory. With the WEDGE installed, however, simply type @ followed by [RETURN], and the present disk drive status will be displayed. This status consists of four parameters: the error number, the error identification by name, and the track and sector where the error occurred (if applicable).

It's easy to see the value of this if we take the case of (unknown to us) a write-protected disk being in the drive when we attempt to save a program. After the red light starts flashing, we can simply type @, and witness the resulting message "26, WRITE PROTECT ON, 18,4" appear on the screen. Thus informed of the problem, we can easily rectify it—keeping intact the program to be saved—without entering the separate error-checking program.

The @ symbol also links to other functions of the DOS WEDGE. Normally it isn't possible to look at a disk's directory without overwriting memory. But with our handy friend installed, just type **@$**, and the directory is displayed without disturbing the program in memory.

To accomplish many disk-oriented housekeeping functions, you normally have to open the command channel with the **OPEN15,8,15** statement, followed by the appropriate **PRINT#15** command. The @ symbol makes these operations simpler, too. You can rename a file by simply typing **@R0:new filename=old filename**, or scratch (delete) a file by typing **@S0:filename**.

The WEDGE even allows you to format a disk by simply typing **@N0:diskname,id**. Note that the **diskname** can be any name (maximum 16 characters) and that the **id** number is any two characters. The **id** number is used by DOS to make sure it's accessing the proper disk, so it's a good idea to have different **id** numbers on each of your disks.

Here's a command that will be of special interest to VIC-20 users. The 1541 disk drive manual states that the 1541 will work with both the C-64 and the VIC-20, but that "each computer has different requirements for speed of incoming data." Without the WEDGE you have to open the command channel with the **OPEN15,8,15** statement followed by **PRINT#15,"UI−"**. We've found that if you install the WEDGE when you first turn on the VIC-20, and then just type **@UI−**, the transfer rate is properly set for the computer, and fewer disk errors occur.

The **@ symbol** is not the only link to WEDGE commands: the / helps speed up loading BASIC programs. To use it, just type **/filename**, and the program specified by **filename** is loaded into memory. Don't include the usual quotes (") around the filename when using this command—this saves a couple of keystrokes.

C-64 Users Only

Several other commands are available only with the **C-64 WEDGE** program. To SAVE a program, type **← filename**, and the file in memory is saved to disk. Be aware that you will still need to use **@0:** to replace a file with an updated version in memory.

There are two other commands available with the **C-64 WEDGE** that make loading and running programs much faster. If you type **↑** before a BASIC program filename, the computer searches the disk for that file, loads it into memory, and then automatically runs the program. Typing **%filename** will load a program file at a location specified within the file itself. This option is normally used when loading machine language programs that need to reside in memory at a location other than the normal BASIC starting point (i.e., other than address 2049).

—Roger Wood

LIMITED SUPPLY

TEXAS INSTRUMENTS 99/4A PRODUCTS

Back Issues of 99'er HOME COMPUTER magazine

Each Only $3.95 postpaid

Canada $4.50, Foreign Surface $5.50, Foreign Air $7.50

ISSUE #1 ISSUE #3 ISSUE #5
ISSUE #2 OUT OF PRINT ISSUE #4

ISSUE #6 (Partial Contents)
• How To Produce Sound Effects • Debugging a Game Program • How to Start a User's Group • Verbose: A Speech Vocabulary Expansion Aid • Color Mapping • Dynamic Manipulation of Screen Character Graphics • The Beginner's Guide to Cassette Operation With the Home Computer • Pre-School Block Letters and Data Compaction • Picking the Ponies in TI BASIC • Battle Star Space Game • 3-D Animation on the Home Computer • Programming Tips • Who is LOGO for? • Tower of Hanoi in TI LOGO • A Review of the TI Lesson-Development Software • An Interview with a Game Designer • Learning Assembly Language with a Magic Crayon • and much, much more.

NOVEMBER 1982 (Partial Contents)
• Chatting with Your Micro: Languages for the Home Computer • A Review of the Smith Corona TP-1 Daisy Wheel Printer • The Micro Jaws Arcade Game • A Knight's Tour in TI BASIC • LOGO Has Style • ASPIC: A Language for Children • A p-System Beginners' Tutorial • An Interview with a p-System Pioneer • A Mini-Memory Screen Dump to the Home Computer Printer • Up Scope!—An exciting Undersea Combat Game • Strategy for Munch Man • A Brief Encounter with a TI Hand-Held Computer • 99'er Shopping Bus • A Pocket Battleship • Sub-Programs in Extended BASIC • Arcade & Adventure Game Reviews • and much, much more.

DECEMBER 1982 (Partial Contents)
• Tex-Scribe: A Text Editor for the Home Computer • A Christmas Computer Carol • Managing a Mailing List the Futura Way • Parsec: The Arcade Game • Plotting With the Home Computer—Pixel by Pixel • Preventing the Situation—Oh No! Memory Full • A Colorful Tour of TI-Fest: The Home Computer Show • Santa's Workshop: The Making of a Home Computer • The Turtle Arcade: Movies & Video Games in LOGO • Controlling a BASIC Termite • The 99'er Gold Rush—An Arcade/Adventure in the Home • 99'er Digest of News & Happenings in the TI World • Plus Games, Reviews, and much, much more.

JANUARY 1983 (Partial Contents)
• Computer Assisted Instruction for the Handicapped • p-System Basics • Debugging in LOGO • The Dow-4 Gazelle Flight Simulator • Note Whiz and Pitch Master Musical Game Reviews • Learning With the PLATO Computer Library • Strategies for Adventure Gaming • Death Drones • Using the Line-By-Line Assembler • Close Encounters of the Simon Kind • Electrical Engineering Education Program • Interview With an Arcade Game Designer • TI Invaders • Programming With Pascal • Cyber-Dice • News and Happenings in the Home Computer World • Arcade Game Reviews • The Thief Adventure Game • Programming Tips • and much, much more.

FEBRUARY 1983 (Partial Contents)
• Texas Instruments at the Winter Consumer Electronics Show • Home Computer Printers on Review • How to Create Math Daisies in LOGO • Vectors in LOGO • ASPIC: A Language for Teachers • The Joys of Adventuring—Part 2 • Programming Pointers with Chuck-A-Luck—Part 4 • Interview With the Voice of Parsec • Why You Need a Printer for Your Home Computer • Lifeline to Titan Space Game • Night Blockade Battleship Game • Tower of Hanoi Pocket Program • Computer Gaming Software Reviews • News of Late Developments in the World of Home Computers • and much, much, more.

MARCH 1983 (Partial Contents)
• An Introduction to the TI-99/2 Basic Computer • The Hex-bus and the 4/A Connection • Making Your Own Say and Spell Game • Disabled Children Learn and Grow • Super Cataloger—A Review of a Disk Library Utility Program • TI's New CC-40 Compact Computer • Robots and Their Social Impact • Twenty Questions With Robot Redford • The Gravity of LOGO • Joystick Jockey—An Overview of Remote Controllers • Parsec Strategy • Converting Extended BASIC to Assembly Language • Matrix Muncher • Mini Memory Disassembler Utility • Pulling the Shade on Sprites • Letters on LOGO • Tiny Tutorials • Games programs, reviews, and much, much more.

APRIL 1983 (Partial Contents)
• Computer Assisted Savings Planning to Build Your Nest Egg • Tex-Cipher Writes and Decodes Secret Messages • Crossbytes—Computer Vocabulary Crossword Puzzle • Cutting Corners On Your Food Budget Using Coupons • Introducing Financial Planning with Multiplan • The Design Philosophy of the Compact Computer • LOGO Takes On the Popular Fifteen Puzzle • Super Language—Programming Sprites in Mini Memory • Colorful Word World—Reading Readiness for Pre-schoolers • Gameware Buffet's A-Maze-ing Boa Alley Game • Giant and Dwarfs Entrapment Game • Game Reviews • Programming Tips • Money Saving Hints • and much, much more.

MAY 1983 (Partial Contents)
• A Consumer's Guide to Word Processing • Word Processing Market Basket • A Generalized Filing Program for VIPs • The Multiplan Medium Balances Your Checkbook and Budget • Activity Accountant Helps School Secretaries with Extracurricular Activities • Maximizing Your Mini Memory's 4K of RAM • Exploring Enhanced BASIC on the Compact Computer • The LOGO Tortoise Debates the BASIC Hare • A Pocket Program to Organize Data with Linked Lists • Mentally Handicapped Learners Team Up with the TI-99/4A • The Wonders of Diskette Storage • Beeline—a Multi-Screen Strategy Game • Lost Ruins—an Archeological Dig Game • 3-D Illusions with Sprites in Depth • Game reviews, Group Grapevine, and much, much more.

JUNE 1983 (Partial Contents)
• Children and Computers Make the 99/4H Connection • Tune Your Guitar with Our TI Tuning Fork • Talk to Your Computer—Voice Technology Is Here • Gameware Buffet's Eat or Be Eaten Aardvark Game • Protect Your Station in the Space Zapper Game • What Multiplan Can and Can't Do • Understanding Inputs and Outputs in Drive For Diskettes—Part 2 • Calculate Loan Schedules on the CC-40 • Go on a LOGO Vacation • Letters on LOGO • A Review of Upper Room Software's Programs for Special Learners • Construct an RS232/Joystick Interface • Group Grapevine • Shopping Bus • A Natural Language Interface for the Professional • Game Reviews • and much, much more.

JULY 1983 (Partial Contents)
• The Evolution of Home Computer Graphics Comes Alive in Graphics Grows Up • Five Data Organizers in Never Out of Sorts • TI & 99'er at the Consumer Electronics Show • WarGames: The Movie and the Book • Editing with Multiplan • The LOGO Logician Presents To Model Is to Learn • LOGO Mosaic Designs Fill the Screen • Your Speech Synthesizer as a Spelling and Foreign Language Teacher • Software for Your Low-cost Printer Port • Gameware Buffet's Treasure Island and the Colorful Switch-A-Row • A Book Review of Learn BASIC for CC-40 Users • 3-D Animation with the TMS 9918A Video Chip • Games Reviews • Group Grapevine • and much, much more.

AUGUST, 1983 (Partial Contents)
• The Home Computer Goes To Work • Bit One, Purl Two at the Fashion Factory • Better Business Bar Graphs in Graphic Persuasion • An Ensemble of Assemblers • Cashflow Helps Money Management • Keystrokes for Thrifty Folks—a Review of Typwriter • Game Reviews of Cavern Quest and Starprobe 99 • Counting Fun for Preschoolers • Peripheral Vision 99 • Mean Machines and Small Potatoes • Multiplan Medium Groups Cells into Rectangles • Turtle Text, a LOGO Word Processor • Group Grapevine • Hello, Little Brother—CC-40 Speaks to 99/4A • Plato's Progress Looks at Student Assessment • Pocket Sunrise • Public Investigator • Gameware Buffet's Jungle Jim, Success Formula, and much, much more!

SEPTEMBER, 1983 (Partial Contents)
•Adding on to Your Home Computer • 99'er Directory of Commercially Available Software, Accessories and Peripherals • Peripheral Vision 99 Hardware Reviews • Pocket Sunrise, Part Two in Extended BASIC • Byte Lightning—Mini Memory Plays a Game • TI-WRITER Tutorial • LOGO's Functions, Sets and Turtles • The CC-40 and 99/4A Take the Data, and RUN • PLATO's Progress Looks at Geometry Courseware and the Shape of Things to Come • Gameware Buffet's Challenge of Camelot and The Fly • Game Reviews of Nit Wit and Crime and Punishment • Group Grapevine • 99'er Hall of Fame • 99'er Digest Update on New Products, and much, much more!

OCTOBER, 1983 (Partial Contents)
• Adventures and Fantasy with Your Home Computer • Do-It-Yourself Adventure • Picoprocessor Emulator, a 4-Bit Microprocessor • Have No Fear: Assembly Language Won't Byte, Part 1 • Make Your Mark • Les Izmore and Debug Computing Cartoon • Once Upon a Tortoise Shell—A LOGO Adventure • Turtle Strut • Number Nibbler for Children • Lots of Plots on Your CC-40 and HX1000 Four-Color Printer/Plotter • Multiplan Medium—Bartender • A Grisly Adventure Bear Hunt • Escape From Wizard's Keep in Extended BASIC • Game Reviews • Shopping Bus • and much, much more.

NOVEMBER, 1983 (Partial Contents)
• Education with your Home Computer • Five Creative Learning Activities for Children • Let's Build America • Have No Fear: Assembly Language Won't Byte, Part 2 • Squeezing the Most Out of TI BASIC • TI-WRITER Tutorial • LOGO Lexicon • Interview with Dale Osborn • TI-WRITER: At Home In the Office • PLATO's Progress • Les Izmore and Debug • The Multiplan Medium • Computer Arrested Instruction—99'er Interviews the Kids • Gameware Buffet: Taco Man and Robo Chase • Game Reviews of Jail Break and Arithmetex • Hall of Fame • 99'er Digest • and much, much more.

Keep Your Back Issues & Tapes Together With a

FINDER BINDER™ 99'er

Only $11.95

- Big enough to hold 6 magazines and 12 tapes
- Uses wire straps to hold magazines so that no hole punching is necessary
- Attractive and Durable

$16.95 with 6 Blank Tapes

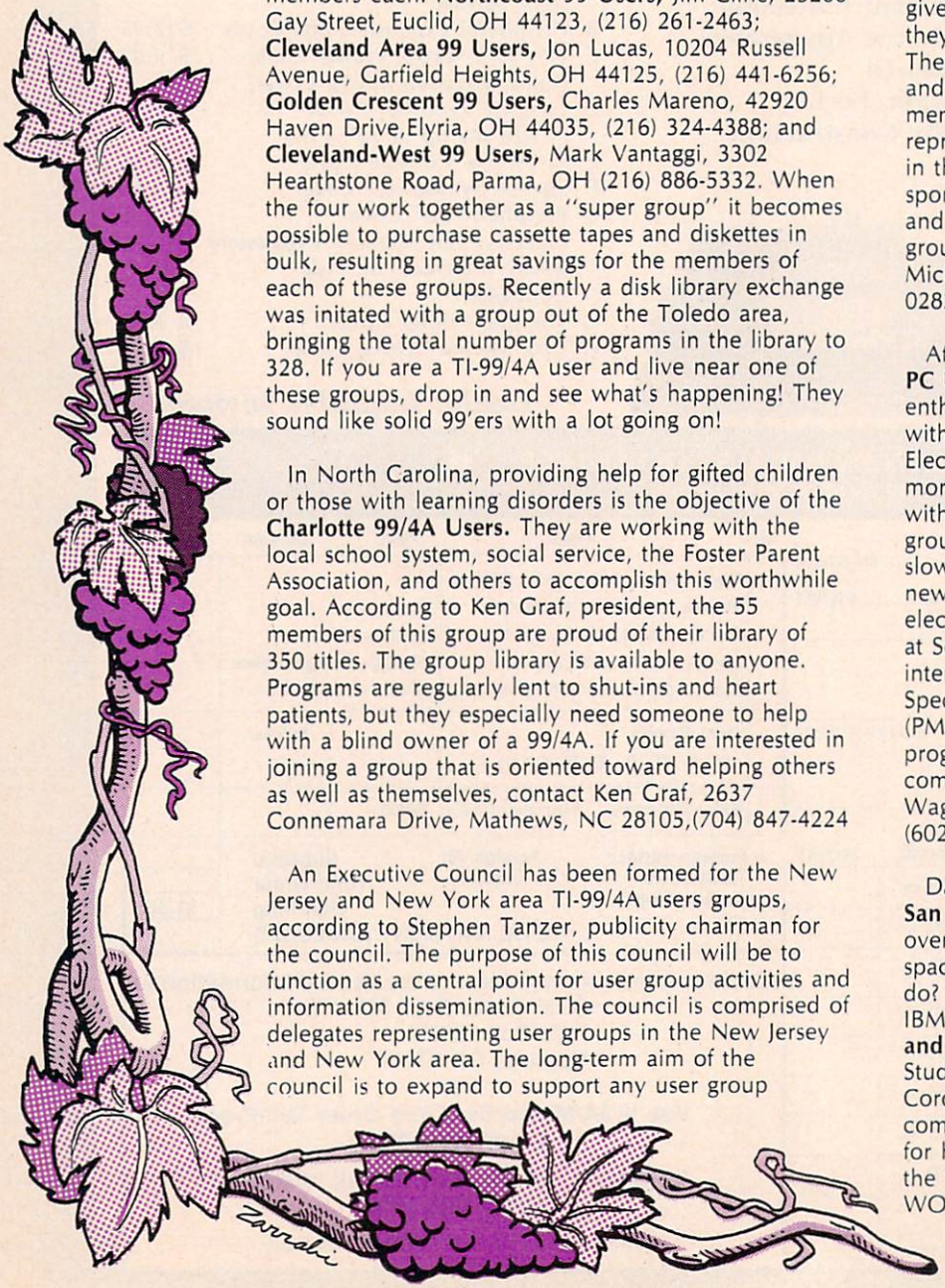
$21.95 with 12 Blank Tapes

12 High-Quality 99'er-WARE C-10 Digital Computer Cassettes Available Separately for $14.00

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.



John Wardrop of the large **Cleveland Area 99/4A Computer User Group** has informed us that they have split into four groups ranging in size from 50 to 70 members each: **Northcoast 99 Users,** Jim Cline, 23200 Gay Street, Euclid, OH 44123, (216) 261-2463; **Cleveland Area 99 Users,** Jon Lucas, 10204 Russell Avenue, Garfield Heights, OH 44125, (216) 441-6256; **Golden Crescent 99 Users,** Charles Mareno, 42920 Haven Drive,Elyria, OH 44035, (216) 324-4388; and **Cleveland-West 99 Users,** Mark Vantaggi, 3302 Hearthstone Road, Parma, OH (216) 886-5332. When the four work together as a "super group" it becomes possible to purchase cassette tapes and diskettes in bulk, resulting in great savings for the members of each of these groups. Recently a disk library exchange was initated with a group out of the Toledo area, bringing the total number of programs in the library to 328. If you are a TI-99/4A user and live near one of these groups, drop in and see what's happening! They sound like solid 99'ers with a lot going on!

In North Carolina, providing help for gifted children or those with learning disorders is the objective of the **Charlotte 99/4A Users.** They are working with the local school system, social service, the Foster Parent Association, and others to accomplish this worthwhile goal. According to Ken Graf, president, the 55 members of this group are proud of their library of 350 titles. The group library is available to anyone. Programs are regularly lent to shut-ins and heart patients, but they especially need someone to help with a blind owner of a 99/4A. If you are interested in joining a group that is oriented toward helping others as well as themselves, contact Ken Graf, 2637 Connemara Drive, Mathews, NC 28105,(704) 847-4224

An Executive Council has been formed for the New Jersey and New York area TI-99/4A users groups, according to Stephen Tanzer, publicity chairman for the council. The purpose of this council will be to function as a central point for user group activities and information dissemination. The council is comprised of delegates representing user groups in the New Jersey and New York area. The long-term aim of the council is to expand to support any user group needing help. Short-term plans of the council include a family picnic in the spring, and hosting an all-day fest devoted to TMS9900-based systems (such as the TI-99/4A). If any user group wishes more information about the Executive Council, they can write to the following address: **Executive Council of Home Computer User Groups,** P.O. Box 84, Dumont, NJ 07621.

The **Greater Rhode Island IBM Users Group** is a relatively new kid on the block, and the fact that they've acquired 100 members in just seven months shows that they are a serious group as well. They are affiliated with the Boston Computer Society, which gives them more depth and also more exposure, since they can put their group news in the BCS newsletter. They already have a library of 15 double-sided disks and will continue to add new programs. According to member Michael Mahoney, they have had an IBM representative demonstrate the PCjr, although no one in the group is a Peanut owner as yet. They have also sponsored demonstrations of the IBM PC XT and DOS, and IBM PC compatibles. If you are interested in this group and would like more information contact Michael Mahoney, Bryer Avenue, Jamestown, RI 02835, (401) 423-2253.

After talking with Brian Wagner of the **Tucson IBM-PC User's Group,** we learned that this band of enthusiasts, 175 strong, is in the process of merging with the much larger Institute of Electronics and Electrical Engineering which will make them an even more attractive and informative group to be associated with. So far there is only one PCjr owner in the group. Brian said sales of the PCjr have been pretty slow in the Tucson area. The group produces a newsletter called "Bits and PCs" and also has two electronic bulletin boards—one at Entre' and a second at Software Land. If you're interested in a special interest group, this is the place to go. They have Special Interest Groups for BASIC, word processing (PMATE—an early word processing program used by programmers), data base, investments, and communications. For more information contact Brian Wagner, 2330 East Edison Street, Tucson, AZ 85719, (602) 795-9437.

David Nussbaum is one busy fellow! His first group, **San Fernando Valley IBM PC Users Group,** swelled to over 250 members, causing a problem in finding space to hold such large meetings. So, what did he do? He opened a new club for all those owners of IBM compatibles and called it **Studio City IBM PC and C-O-M-P-A-T-I-B-L-E-S Computer Club!** The Studio City group serves owners of the IBM PCjr, Corona, Compaq, Eagle, Columbia and other compatibles. As a result of David's "vehement" cry for help, several SIG groups are being formed within the San Fernando Valley group. SIGs such as WORDSTAR, electronic spread sheets, word

processing, accounting, database, Pascal and BASIC, and a beginners group have surfaced. Dynamic speakers with programs relevant to what's happening in the world of the PC seem to be the rule rather than the exception at these monthly meetings. For more information, contact David Nussbaum, 11558 Riverside Drive, #207, (213) 985-8337.

Before you can have apple cores, apple pies, and apple worms, you have to have an apple tree, right? Well, Group Grapevine has heard from our first apple tree—**Suncoast Apple Tree** (SCAT)—a users group from Clearwater, Florida. Some of its members, such as Chuck Quenzler, have taken a unique approach to the multitude of everyday questions that crop up among new and experienced computer users: They invite small groups into their homes for informal "classes." The group library is so extensive that an exact count was difficult, but they have enough titles to fill four large notebooks, including titles from Washington Apple Pi and the International Apple Corps. Members are encouraged to bring in their own disks and duplicate the group catalogs. Special Interest Groups include telecommunications, bulletin boards, VISICALC, investments (they have their own in-house stockbroker), and general information. SCAT has two electronic bulletin boards which operate 24 hours a day and average 1000 calls per month. SCAT has also been working with other users groups in the area as well as the Computer Talk Show (on television station WPLP), to organize a swap meet for the near future. For more information on this group and their swap meet, contact Chuck Quenzler, 2038 Temple Terrace, Clearwater, FL 33546, (813) 531-7190.

According to Stuart Greenfield of the Austin, Texas-based **River City Apple Corps,** "He **WOZ** here!" Yes, Steve Wozniak visited the March 19 meeting and gave a presentation which included the real, complete Apple Story. "I really didn't expect him to be so honest and down-to-earth. Once again Apple computer has demonstrated its concern for its owners and its image as a family," Greenfield said. The WOZ also mentioned new products, one of which (the Apple IIc) was unveiled on April 24 in San Francisco. SIGs include Game Interest Group (GIG), MAC interest group (MACig), Investors Interest Group ($IG), and Kids Interested in Koalas (KIK). We don't know whether KIK is for kids (the smaller under 18 version) or kids (the larger version of a more advanced age)! River City Apple Corps members number 289, so if you'd like to help them reach the big 300, contact: Barry Wulfe (512) 474-1393.

The year was 1978 when five Apple "worms" first met in a garage. Since then, they have multiplied over the years to 283 members! This bit of user group trivia comes from G. F. Rowe, an original member of the **Tidewater Apple Worms** in Norfolk, Virginia. Mr. Rowe's love affair with the Apple is representative of many Apple users. It's the "do anything" machine. "You can get all the time, entertainment, and education you want from it and you can take the whole thing apart and put it back together with little bother!" Mr. Rowe feels that the Macintosh is fast, that the graphics can't be beat, and that in the not too distant future it will be more popular than the IBM PC for business uses. The Tidewater Apple Worms have several SIGs, including Green Apples (newcomers), VISICALC, word processing, simulation, and games. They will also have an exhibit at a computer show held in the Southeast in May. If you'd like to "wiggle" into this apple, contact G.F. Rowe, 3025 Vendome Terrace, Norfolk, VA (804) 625-0133.

You Commodore users in the Newark, Delaware area now have a users group of your very own. The **Newark Commodore Users Group (NCUG)** has been meeting on a regular basis since October 1983. At the present they have 30 plus members (mainly VIC—20 and C-64 owners) attending meetings, and they are growing each month. If you are interested in more information on this group, contact Bob Black, 210 Durso Drive, Newark, DE 19711, (302) 737-4686.

8 ^ 2 (Eight Squared), a computer club for Central Pennsylvania Commodore 64 users, has incorporated a club store into their monthly meetings. The store offers software, books and blank disks, as well as a Program of the Month, on a first-come, first-served basis. 8 ^ 2's Special Interest Groups include beginners, business/spreadsheet, alternate programming languages, modem/telecommunications, and adventurers. Besides SIG's, this Central Pennsylvania group offers classes in Beginning BASIC, Intermediate BASIC Programming, and Introductory Machine Language. The Introductory Machine Language class covers memory registers, 6510/6502 microprocessor instructions, and memory addressing modes. Programs will be written using Supermon. For more information, contact Andy Skelton (717) 486-3274.

The **Rockville VIC/64 Users Group** of Rockville, Maryland makes it possible for the whole family to attend their meetings. While the adults are having their meetings, the kids can see demonstrations of games such as Invaders, Jupiter Lander, Nite Rider, Centipede, Frogger, etc. Also, the Kid's Corner will be trying out a new idea. The club will give them disk mailers that will make it possible to swap programs by mail with their friends. If you are interested in the Kid's Corner, contact Patrick pounds, P.O. Box 8805, Rockville, MD. 20856, (301) 231-7823.

HCM

Companion ... from p. 145

parameters can be revised within specified minimum and maximum limitations. These commands can also be set retroactively. If you decide, halfway through your article, that you want it double-spaced, you simply go back to the beginning of the text and insert a parameter revision command. The printout parameter changes are carried out on the line following the command. A REVISE IMMEDIATE command lets you make printout parameter changes on the line of text in which the command is made, and a TOGGLE REVISION command is useful when a parameter must be repetitively alternated between two particular values.

Companion includes commands that allow it to print in fancier modes, such as double or compressed mode, or emphasized mode. There is also a text blocking command to prevent certain passages of text from being split between two pages. And *Companion*'s formatting commands are always within one or two keystrokes' reach—not in separate routines that must be reloaded from diskette. This is yet another case where *Companion* is more convenient to use than *TI-WRITER*, which requires you to reload the diskette to access formatting commands.

While I'm hopping in and out of modes and menus, I should mention one aspect of *Companion* that is baffling to the uninitiated, but quite useful if used correctly. If you work on a file, go back to the menu, and then reload it, *Companion* will load the file starting wherever you left your cursor. Those foolhardy individuals who use *Companion* without first reading the manual will find themselves floundering in confusion, try-

Home Budget,jr ... from p. 152

program by pressing the [ESC] key. If you choose to change the account, a screen comes into view with instructions on using the editing commands. From this point on, editing the items is a simple task. You can even insert new items into the account with this option, although the manual suggests that you use only the Enter Expenses or Income option from the master menu rather than the account editor.

None of this information is of any use unless it is totaled and listed in the form of a report. Account Listing, option 4 from the main menu, does just this. You can elect to report on a single account or on all accounts. You can also select entries for one month or for the whole data base (generally you will keep one year in a single data base). This option lists all of the entries for the account with a total of expenses or income at the bottom of the report.

Now you will probably want to find out just how well you have been sticking to your budget. Option 6, Examine Budget, posts all of the accounts by comparing the amount provided for in the budget with the actual amount spent. You can elect to examine one account or the whole budget. If you examine one account, the report will include the budget amount (the amount you expected to spend in this account), the actual amount spent, and the balance. These figures are reported for month-to-date, and also for year-to-date, so that you can see not only how you are doing for the current month, but also for the whole year, at the same time.

The display for charge accounts is slightly different. The Charge Account option will show you how much you have spent as well as your year-to-date balance.

You can also get a month-to-date, or year-to-date total for all accounts. Here you see all of your accounts listed with the same values displayed as those shown with each single account. After viewing all of the accounts' values, you are guided to the totals screen, where you will see just how well you have been doing. The program will show how much you allowed for in your budget, how much you spent, and how much is left over. In addition, you will be shown your total income, and the difference between your total income and the amount you budgeted.

NOTE:

When selecting software of this nature you should keep in mind one important thing: With many programs there is a trade-off between program power and ease of use. Programs are available which are so easy to use they don't even require that you open the manual. These programs generally don't allow for much detail, and are designed for very small applications. An example of this might be *Home Budget Manager*, made for the TI-99/4A home computer. On the other hand, you have programs like *Home Budget,jr* which require some instruction, but only because of their increased complexities.

You should also consider this when purchasing any financial software: Will the program fulfill your needs now and in the near future, or is it "over-qualified" for the job presently at hand?

Ease of Use

Once you have gone through the manual and learned what each option does and how it affects the budget, you will have no trouble using and updating your budget records. I have seen easier programs to use, but they had a lot less to them. The *Home Budget,jr's* numerous options and attention to detail make it a bit more difficult to use than some of these. However, if you take the time to sit down at the computer and follow the tutorial format of the manual, you will have no problems. To assist the beginner, IBM has placed a number of options throughout the program which direct the user to HELP screens for crucial information and instructions. This is a feature that every program should have.

The entire program is menu-driven, making operation a snap. At any point in the program you can press the [ESC] key and go back to the previous menu to abort your present operation. All of the prompts are easy to understand and are sequenced in logical order. I cannot imagine any home budget being too big or too complicated for *Home Budget,jr*. Yet, the program has retained the operator friendliness of a less-comprehensive program. Clear, easy-to-read error messages let you know when you make a mistake. Any time you enter data into the system, you are asked whether the data you just entered is correct, and are given the option to re-enter it.

Documentation

A 98-page manual comes with *Home Budget,jr*. Set-up instructions are included, as well as a description of the account structure and how it works. The bulk of the manual is a "Do As You Read" tutorial which guides you through every aspect of a demonstration budget. The best way to understand a program is to sit down and use it, and that is just what the manual makes you do. You are carefully guided through every keystroke to build a budget, to add to it, to maintain it by correcting entries, and finally, to get reports from it.

Good programs provide some indication when an error has occurred. Unfortunately, most of them give you cryptic messages which can only be understood by programmers. *Home Budget,jr*, however, provides error messages that are easy to understand. And just in case, there are descriptions of each error message in the documentation, along with helpful hints as to what caused the problem and how you can avoid it. This is an area in which most other programs and documentation fall far short.

Saving Money

If you are constantly trying to put yourself on a budget—only to find that you can't maintain it—*Home Budget,jr* may be just the answer to your prayers. With this program you will be better able to track your budget's progress, and to pinpoint those trouble areas. And, if used regularly, it should be able to straighten out anyone's budget—probably saving them money in the process.

The details that this program is capable of retaining for each item makes it perfect not only for budgeting, but for a number of other uses. For instance, the information included in the records is adequate for most tax record-keeping purposes, or to see if you can afford a new car or home. The only feature missing in *Home Budget,jr* is a graphics option to plot the budget on the screen. The excellent graphics capabilities of the PCjr would have made this a valuable part of the program. Despite this lack of graphics output, *Home Budget,jr* is well worth the asking price, and a valuable addition to anyone's software library.

HCM

ing to figure out why their file is suddenly too large to load or their introduction is inserted in the middle of the bibliography. This problem is easily solved by purging your file before you load it. On the plus side, this makes it extremely easy to work with "boilerplate" documents—loading chunks from other files at specified points within a form file.

Search and Destroy Missions

Companion also features the popular and useful search commands. These let you locate a specified word or phrase everywhere it appears in the text. There are four of these: FIND lets you locate the specified word or phrase each time it occurs. COUNT tells you how many times the specified sequence occurs (useful for linguistic analyses of one's prose). REPLACE lets you exchange the specified item with another by entering the word to be replaced and its replacement between slashes (e.g., /scum/undesirable element/). DELETE simply removes the not-so-bon mot wherever it appears.

Commands for moving, deleting and copying blocks of copy are the heart of a word processor's real value. *Companion*'s copy-moving functions are very convenient to use. Instead of hassling with line numbers or other representations, you simply put the copy to be moved within heavy brackets, then position your cursor where it is to go. The brackets will not print out, so you have the option of leaving them in, should you want to leave behind a history of your editing process for literary scholars studying your work.

Documentation Debatable

Companion's documentation gets a mixed review. As a "how-to" manual, it is excellent—clear and easy to understand. I especially like the way the first chapter lets you jump right in and use the system immediately, without first wading through pages of dull procedural descriptions. A short, step-by-step script explains how to load the diskette, enter text, and print out your file.

On the minus side, it is not designed for use as a reference tool. It has no index. It does have a fairly detailed table of contents, but the items are not arranged alphabetically or according to any easily understood logic. So you have to read through the entire table of contents to find each item. And individual commands are not necessarily listed there at all. This can be very irritating if, for example, you have made a colossal error and are frantically searching for the "I-take-back-what-I-just-did" command.

This brings up another problem: *Companion* has no "oops" command. This is a provision found in *TI-WRITER* and many other word processors that lets you cancel any mistaken command you have just made. So if you accidentally delete the dynamite ending you wrote at the terminal, it is not lost forever. Actually, I never thought the oops key was very important until I was using *Companion* and accidentally pressed [FCTN][3]. The entire screen (the entire file!) filled with dots between every word. Dismayed at the thought of having to go through and space between each word in a 1000-word text, I flipped through the manual, searching in vain for a handy reference chart of commands. A complete list of commands should be a requirement for any word processing manual. And a removable reference card—such as you get with *TI-WRITER*—is even better. (Incidentally, you can cancel the polka-dot effect with another [FCTN][3].)

The Final Verdict

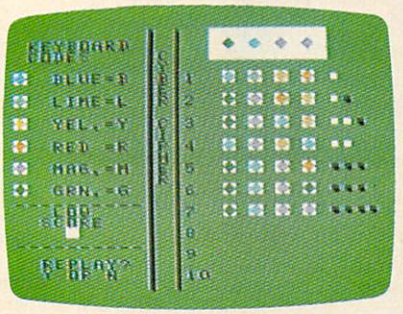
Companion certainly ranks among the best word processors for the 99/4A. It contains all the most important editing, text-moving and formatting functions and is exceptionally easy to use. Whether it is the best system for *you* depends on three factors: 1) your particular word processing needs, 2) whether or not you have or want *Extended BASIC*, (required for *Companion*), and 3) the availability and prices for both *Extended BASIC* and *TI-Writer*. Keep in mind that both systems also require a 32K memory expansion device. [For availability and prices, see ads in this issue.—Ed.]

For serious word processing, *TI-WRITER* will probably perform best—especially if you need to see your text formatted on screen and utilize formatting features not found in *Companion* (on-screen word wrap, 40+ screen margins, and mailing list options). Also, because *TI-WRITER* is the de facto standard in the 99/4A world, there remains a good possibility for third-party enhancement products that link to *TI-WRITER* files. [TI has generously provided the "hooks" for these add-ons. At the time this issue is going to press, we've already heard of a spelling-checker enhancement that may shortly be available.—Ed.]

But if you already own (or want to own and can find) *Extended BASIC*, and desire a word procesor for the ultimate in writing ease, you should check out Intelpro's product. For just writing (as opposed to formatting), I found it faster and easier to use than *TI-WRITER*. Writing is, after all, a lonely business; a helpful *Companion* at your side could ease that burden. HCM

HCM Program Bug

DeBUGS on DisPlay



An incorrect screen photo for Cyber-Cipher was inadvertently printed on page 113 of this issue. The above screen photo is correct.

HCM CLASSIFIEDS
RATES AND INFORMATION

The *HCM* Classifieds is an economical alternative to general display advertising. All listings appear in a standard format, typeset by *HCM* and are published in the first available issue after receipt.

RATES: $175 each ad, per issue

AD FORMAT: 1 line heading (25 characters maxium)
8 lines body copy (40 characters per line)
4 lines for company name, address and telephone

ORDERING: Send listing information with payment* to:
HCM Classifieds,
1500 Valley River Drive, Suite 250
Eugene, Oregon 97401

*Check, Money Order, VISA or MasterCard are accepted.

COMPUTER AGE
One of the largest selections of software in town! Hardware and software for the TI PC, Allen UPC and the Commodore 64. Texas Instruments 99/4A $79.95, RS232 Interface Card $117.95, Peripheral Expansion System $549.95. Includes: peripheral expansion box, disk drive, disk drive controller card, 32K memory card. Limited Quantities!

COMPUTER AGE Prestonwood Town Ctr.
160 Sharpstown Ctr. or 5301 Belt Line #1089
Houston, TX 77036 Dallas, TX 75240
(713) 270-1257 (214) 458-8309

TI HEBREW/JEWISH PROGRAMS
Complete set of Hebrew language & Jewish educational software. XBasic optional. Exciting method of religious education. Ride Jewish Time Machine! Dig up ancient artifacts in Israel! Be a Mossad agent! Play Hebrew vocabulary bingo! Turn your TI into a Hebrew typewriter! Have mystic Kaballah adventure! Get free catalog.

MICRO-MELAMED SOFTWARE COMPANY
6130 Coralridge Drive
Corpus Christi, TX 78413
(512) 853-2808

TI-XBASIC OR IBM PC/PCjr
HOME BUDJET-user defined expenses that let you track your income using GRAPHS. CHECK MANAGER—Multi—feature program that lets you look forward to your statement. SEARAIDER—fast action game on the high seas and outer space. . . then the tunnel! * You get all 3 on disk or cassette * Send 19.95 + 1.50 postage/handling to

NOVA SOFTWARE
10374 White Elm
Dallas, TX 75243
(214) 231-4980 VISA OR MASTERCARD

AC GAMBLER—FOR TI/994A
Atlantic City Gambler includes BLACKJACK & CRAPS. 1-5 players are dealt cards across the table (Hit, Stay, Dbl Down). Craps table includes PLACE BETS-COME-PASS LINE-FIELD & more (proper odds payoff). Includes rules, music, moving graphics (Cards, Dice, Chips). $16.95
REQ'D TI-EX-BASIC, MEM-EXP, DISK, JOYSTICK.

WAYNE & SON SOFTWARE
16 Dogwood Dr.
Andover, N.J. 07821

TI-FOOTBALL, INVENTORY TOO
FOOTBALL ANALYST—Beat the point spread! $35. PUZZLER—Non—violent educ. game for kids. $25. PERSONAL SOCIAL REGISTER—automates your address book and more!! Disk, 32K. $50. PERSONAL INVENTORY—keep track of all your possesions. $35. All reg. XB. Catalog avail. $1 ref. w/order. Dealer inquiries welcome. NY res. add tax.

RIDGE SERVICES
170 Broadway, Suite 201
New York, NY 10038
(212) 833-6335

QUALITY TI 99/4A SOFTWARE
For Home, School and Business featuring CREATE-A-FILE menu driven DATABASE management program. Keyword Search, Scan/Print Update. Print Labels or Text. Disk $39.95. TEACHER'S RECORD Grading/Storage programs. Cassette $24.95. Expanded on Disk $44.95. Will contract to custom write or modify. Send $1.00 for full catalog. Refundable.

ARRO-SOFT SYSTEMS
P.O. Box 1761
Edmond, OK 73083
405-341-8567

YOUR TI-99/4A SPECIALIST
True standard IEEE cards by Cor-Comp with 1 year warranty. Includes: RS-232 card for peripheral expansion box, 2 serial ports and 1 parallel port, only $119.95. Also RS-232 Stand Alone, $139.95. And 32 K cards for expansion box, $149.95. We're adding new products for your TI-99/4A.

BAY CITY ELECTRONICS
624 Woodside Road
Redwood City, CA 94061
(415) 367-1880

2 PROGRAMS FOR THE 99/4A
Expense record keeping—ASSEM LAN and XBAS, E/A or XBAS, required, please specify. Mem. exp. req. store & process up to 7500 individual income & expen. items, printer opt., $25 per disk. Racquetball plus assembly language XBAS or ED/ASM & memory expansion req. 1-4 plyrs(joystks req. for 4), $9 per disk. Send check or money order to:

KRONITON SOFTWARE
4007 ANNIE STREET
ALEXANDRIA, LOUISIANNA 71301

ATTENTION 99'ERS "GOSUB"
Have you ever spent weeks looking for that certain subroutine? Well, help is here with GOSUB. A dictionary of over 75 subroutines and hints for the 99/4A. All of which can be listed and/or run separately for a demonstration. Three full programs. TI Basic. Cassette only.
All three for only $15.95

K C F Software
5 Pearl Street
Schuylerville, NY 12871
(518) 695-6709

TI-BASIC SPREADSHEET!
All you need is the TI-99/4A console and the cassette interface cable. This program lets you unlock the power of your TI console. TI-MiniCalc allows you, the small businessman or home owner, to look at those what-if situations. The TI-MiniCalc comes on cassette with tutorial and sample spreadsheets. For only $19.95.

ORION SOFTWARE
P.O. Box 391
St. Joseph, IL 61873
(217) 469-2354

INDEX TO ADVERTISERS

SENSATIONAL SOFTWARE GIVEAWAY

1 YOUR CHOICE OF THIS ISSUE'S PROGRAMS AT A REMARKABLE PRICE—

ONLY $3.95!

Proof Of Purchase Necessary

To participate in our monthly Software Giveaway, you need to be a bona-fide purchaser of ***Home Computer Magazine*** and fill out the questionnaire on the reverse side.

Have you taken advantage of our Software Giveaway before?
☐ Yes ☐ No

You will receive all the programs **ON TAPE™** or **ON DISK™** (which have versions for your selected machine) whose listings appear in this issue.

IMPORTANT: The order form below and the questionnaire on the reverse side must be completed. Tear out this entire page and enclose in an envelope along with $3.95 ($5.95 in Canada & Mexico). Payment must be made by check, money order, or VISA/MasterCard. Proof of purchase (subscriber label number, sales receipt or any reasonable facsimile thereof) must also accompany this form.

*FREE SOFTWARE!

2 When Subscribing To HOME COMPUTER™ magazine

Subscribe or Renew today, and with your **paid** subscription you will receive **FREE** software — **ON TAPE™** or **ON DISK.™** With a 1-year subscription to ***Home Computer Magazine*** you get **2 FREE months** of **ON TAPE** or **ON DISK.** Subscribe for 2 years and receive **4 months.** And with a 3-year subscription we'll give you **6 full months** of this convenient software on cassette tape or floppy disk.

***DON'T HAVE A COMPUTER? TAKE A RAINCHECK!**
We'll give you a raincheck for the FREE software so that when you buy a computer, we will send you your choice of ON TAPE or ON DISK as a FREE BONUS as stated in this offer.

And You Save 30% Off The Single-Copy Price Of The Magazine!

SAVE EVEN MORE!

3 AND ENJOY THE CONVENIENCE OF A PROGRAM SUBSCRIPTION

By subscribing to **ON TAPE™** or **ON DISK™** you will save money off the single-copy price ***and*** receive the same high-quality programs published monthly in the magazine—delivered right to your door ***each and every month!*** This cassette tape or floppy disk program service is the convenient, accurate, and affordable way to save hundreds of typing hours.

The Perfect Addition To Your Magazine Subscription!

3-IN-1 ORDER FORM

PLEASE PRINT

Name_____

Address_____

City _____ St____ Zip_____

☐ Check or Money Order Enclosed

MUST BE IN U.S. FUNDS DRAWN ON A U.S. BANK

Bill my ☐ VISA ☐ MasterCard

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Account No.

_____ _____
Signature Exp. Date

Enclose payment or credit card information and mail entire page with completed form to:

HOME COMPUTER MAGAZINE
P. O. Box 5537, Eugene, OR 97405
-OR-

Use our TOLL-FREE LINE for Visa/MasterCard only:
1-800-828-2212 In Oregon, Alaska, Hawaii Tel.(503)485-8796

THIS SECTION MUST BE COMPLETED

INDICATE YOUR CHOICE OF MEDIA: ☐ **ON TAPE™** ☐ **ON DISK™**

INDICATE WHICH COMPUTER MEDIA IS FOR: (pick one)

☐ **Apple II family** ☐ **C-64** ☐ **VIC-20** ☐ **IBM PC** ☐ **PCjr** ☐ **TI 99/4A**

☐ **Please send a raincheck for the FREE software.**

1 **YES!** Send me all the programs in this issue: **Vol.** _____ **No.** _____ which have versions for my selected machine. I have indicated my choice of media and have enclosed $3.95 ($5.95 in Canada & Mexico).
Proof of purchase and completed questionnaire must be included.

2 **YES!** Enter my subscription to ***Home Computer Magazine*** for the term below:

Please check one: ☐ **New** ☐ **Renewal (include subscriber number)**

☐ 1-yr (12 issues) $25 ☐ 2-yr (24 issues) $45 ☐ 3-yr (36 issues) $63

PLUS 2 FREE **ON TAPE or ON DISK** ***PLUS 4 FREE*** **ON TAPE or ON DISK** ***PLUS 6 FREE*** **ON TAPE or ON DISK**

Canada add $7; Foreign Surface add $18 for 1-yr magazine subscription. Free software not available outside USA.

3 **YES!** I want to save time and money. Please enter my subscription to ON TAPE or ON DISK for the term listed below:

☐ 10 MONTHS—*ONLY $38* ☐ 12 MONTHS—*ONLY $45*

Canada add $15 for software subscription.
Software subscription not available in other countries at this time.

Please allow 6-8 weeks for your first issue—Magazine & Media shipped separately.

SGSO/8-84

Satisfaction Guaranteed—or the unfilled portion of your subscription will be refunded, less the cost of any premiums you have received.

3 FANTASTIC OFFERS

HOME COMPUTER™ magazine

QUESTIONNAIRE

Complete and mail to: Home Computer Magazine • P.O. Box 5537 • Eugene, Oregon 97405

FOR ALL READERS

1. **Where did you obtain this copy of Home Computer Magazine?** ☐Subscriber ☐Supermarket ☐Bookstore ☐Users group ☐Newsstand ☐Computer Store ☐Friend ☐Library ☐Other _____
2. **What types of software are you most interested in?** ☐Educational ☐Entertainment ☐Computer Literacy ☐Household Management ☐Job-Related Applications ☐Business ☐Other _____
3. **Are you** ☐Male ☐Female ☐14 or younger ☐15-24 ☐25-34 ☐35-44 ☐45-54 ☐55+
4. **Annual Household Income?** ☐Under $10,000 ☐$10,000-$14,999 ☐$15,000-$19,999 ☐$20,000-$24,999 ☐$25,000-$29,999 ☐$30,000-$39,999 ☐$40,000-$49,999 ☐$50,000+
5. **Occupation?** ☐Professional ☐Management ☐Teacher ☐Student ☐Other _____
6. **What is your ZIP code?** ☐☐☐☐☐
7. **What is the current month and year?** _____
8. **Do you presently own a Home Computer?** ☐No ☐Yes. It is a ☐TI-99/4A ☐Apple II/II+/IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐Other _____

FOR READERS WHO PLAN TO BUY A HOME COMPUTER

9. **Which model do you think you'll purchase?** ☐Apple IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐TI-99/4A ☐Other _____
10. **When do you expect that purchase to be?** ☐less than 3 months ☐3-6 months ☐7-12 months ☐at least 1 year
11. **What do you anticipate your primary use of a home computer will be?** ☐Entertainment ☐Education ☐Computer Literacy ☐Household Management ☐Job-Related Applications ☐Business ☐Other _____

FOR PRESENT HOME COMPUTER USERS

12. **Which home computer(s) do you currently own?** ☐Apple II/II+/IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐TI-99/4A ☐Other _____
13. **What is the primary use of your home computer?** ☐Entertainment ☐Education ☐Computer Literacy ☐Business ☐Job-Related Applications ☐Household Management ☐Other _____
14. **How often is your computer in use?** ☐Less than 1 hour per week ☐1-4 hours ☐5-10 hours ☐11-15 hours ☐16-20 hours ☐over 20 hours
15. **On the average, about how many program listings in each issue of HCM do you key into your computer and use?** ☐None ☐1 ☐2 or 3 ☐4 or more
16. **What peripherals do you currently use?** ☐Disk System ☐Printer ☐Modem ☐Monochrome/Color Monitor ☐Other _____
17. **What do you expect to buy within the next year?** ☐Software ☐Disk system ☐Printer ☐Modem ☐Books ☐Magnetic Media ☐Monochrome/Color Monitor ☐Furniture & Accessories
18. **How much do you expect to spend on computer-related products during the next year?** ☐Less than $25 ☐$25-$49 ☐$50-$99 ☐$100-$249 ☐$250-$490 ☐$500-$999 ☐$1000-$2499 ☐$2500 or more

OPTIONAL: If you would like to help us by participating in a telephone interview, please include your telephone number (____) - _____ here and the most convenient time you can be reached _____:_____ ☐AM ☐PM

SENSATIONAL SOFTWARE GIVEAWAY

3 FANTASTIC OFFERS

1 YOUR CHOICE OF THIS ISSUE'S PROGRAMS AT A REMARKABLE PRICE—

ONLY $3.95!

Proof Of Purchase Necessary

To participate in our monthly Software Giveaway, you need to be a bona-fide purchaser of ***Home Computer Magazine*** and fill out the questionnaire on the reverse side.

Have you taken advantage of our Software Giveaway before?
☐ Yes ☐ No

You will receive all the programs **ON TAPE™** or **ON DISK™** (which have versions for your selected machine) whose listings appear in this issue.

IMPORTANT: The order form below and the questionnaire on the reverse side must be completed. Tear out this entire page and enclose in an envelope along with $3.95 ($5.95 in Canada & Mexico). Payment must be made by check, money order, or VISA/MasterCard. Proof of purchase (subscriber label number, sales receipt or any reasonable facsimile thereof) must also accompany this form.

*FREE SOFTWARE!

2 When Subscribing To HOME COMPUTER™ magazine

Subscribe or Renew today, and with your **paid** subscription you will receive **FREE** software — **ON TAPE™** or **ON DISK.™** With a 1-year subscription to ***Home Computer Magazine*** you get **2 FREE months** of **ON TAPE** or **ON DISK.** Subscribe for 2 years and receive **4 months.** And with a 3-year subscription we'll give you **6 full months** of this convenient software on cassette tape or floppy disk.

***DON'T HAVE A COMPUTER? TAKE A RAINCHECK!**
We'll give you a raincheck for the FREE software so that when you buy a computer, we will send you your choice of ON TAPE or ON DISK as a FREE BONUS as stated in this offer.

And You Save 30% Off The Single-Copy Price Of The Magazine!

SAVE EVEN MORE!

3 AND ENJOY THE CONVENIENCE OF A PROGRAM SUBSCRIPTION

By subscribing to **ON TAPE™** or **ON DISK™** you will save money off the single-copy price ***and*** receive the same high-quality programs published monthly in the magazine—delivered right to your door ***each and every month!*** This cassette tape or floppy disk program service is the convenient, accurate, and affordable way to save hundreds of typing hours.

The Perfect Addition To Your Magazine Subscription!

3-IN-1 ORDER FORM

PLEASE PRINT

Name _____

Address _____

City _____ St____ Zip_____

☐ Check or Money Order Enclosed

MUST BE IN U.S. FUNDS DRAWN ON A U.S. BANK

Bill my ☐ VISA ☐ MasterCard

Account No.

_____ _____
Signature Exp. Date

Enclose payment or credit card information and mail entire page with completed form to:
HOME COMPUTER MAGAZINE
P. O. Box 5537, Eugene, OR 97405
-OR-
Use our TOLL-FREE LINE for Visa/MasterCard only:
1-800-828-2212 In Oregon, Alaska, Hawaii Tel.(503)485-8796

THIS SECTION MUST BE COMPLETED

INDICATE YOUR CHOICE OF MEDIA: ☐ **ON TAPE™** ☐ **ON DISK™**

INDICATE WHICH COMPUTER MEDIA IS FOR: (pick one)

☐ **Apple II family** ☐ **C-64** ☐ **VIC-20** ☐ **IBM PC** ☐ **PCjr** ☐ **TI 99/4A**

☐ **Please send a raincheck for the FREE software.**

1 **YES!** Send me all the programs in this issue: **Vol.** ____ **No.** ____ which have versions for my selected machine. I have indicated my choice of media and have enclosed $3.95 ($5.95 in Canada & Mexico).
Proof of purchase and completed questionnaire must be included.

2 **YES!** Enter my subscription to ***Home Computer Magazine*** for the term below:

Please check one: ☐ **New** ☐ **Renewal (include subscriber number)**

☐ 1-yr (12 issues) $25 ☐ 2-yr (24 issues) $45 ☐ 3-yr (36 issues) $63

PLUS 2 FREE	*PLUS 4 FREE*	*PLUS 6 FREE*
ON TAPE or ON DISK	ON TAPE or ON DISK	ON TAPE or ON DISK

Canada add $7; Foreign Surface add $18 for 1-yr magazine subscription. Free software not available outside USA.

3 **YES!** I want to save time and money. Please enter my subscription to ON TAPE or ON DISK for the term listed below:

☐ 10 MONTHS—*ONLY $38* ☐ 12 MONTHS—*ONLY $45*

Canada add $15 for software subscription.
Software subscription not available in other countries at this time.

SGSO/8-84

Please allow 6-8 weeks for your first issue—Magazine & Media shipped separately.

Satisfaction Guaranteed—or the unfilled portion of your subscription will be refunded, less the cost of any premiums you have received.

HOME COMPUTER™ magazine

QUESTIONNAIRE

Complete and mail to: Home Computer Magazine • P.O. Box 5537 • Eugene, Oregon 97405

FOR ALL READERS

1. **Where did you obtain this copy of Home Computer Magazine?** ☐Subscriber ☐Supermarket ☐Bookstore ☐Users group ☐Newsstand ☐Computer Store ☐Friend ☐Library ☐Other _____
2. **What types of software are you most interested in?** ☐Educational ☐Entertainment ☐Computer Literacy ☐Household Management ☐Job-Related Applications ☐Business ☐Other _____
3. **Are you** ☐Male ☐Female ☐14 or younger ☐15-24 ☐25-34 ☐35-44 ☐45-54 ☐55+
4. **Annual Household Income?** ☐Under $10,000 ☐$10,000-$14,999 ☐$15,000-$19,999 ☐$20,000-$24,999 ☐$25,000-$29,999 ☐$30,000-$39,999 ☐$40,000-$49,999 ☐$50,000+
5. **Occupation?** ☐Professional ☐Management ☐Teacher ☐Student ☐Other _____
6. **What is your ZIP code?** ☐☐☐☐☐
7. **What is the current month and year?** _____
8. **Do you presently own a Home Computer?** ☐No ☐Yes. It is a ☐TI-99/4A ☐Apple II/II+/IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐Other _____

FOR READERS WHO PLAN TO BUY A HOME COMPUTER

9. **Which model do you think you'll purchase?** ☐Apple IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐TI-99/4A ☐Other _____
10. **When do you expect that purchase to be?** ☐less than 3 months ☐3-6 months ☐7-12 months ☐at least 1 year
11. **What do you anticipate your primary use of a home computer will be?** ☐Entertainment ☐Education ☐Computer Literacy ☐Household Management ☐Job-Related Applications ☐Business ☐Other _____

FOR PRESENT HOME COMPUTER USERS

12. **Which home computer(s) do you currently own?** ☐Apple II/II+/IIe ☐Commodore 64 ☐VIC-20 ☐IBM PC ☐PCjr ☐TI-99/4A ☐Other _____
13. **What is the primary use of your home computer?** ☐Entertainment ☐Education ☐Computer Literacy ☐Business ☐Job-Related Applications ☐Household Management ☐Other _____
14. **How often is your computer in use?** ☐Less than 1 hour per week ☐1-4 hours ☐5-10 hours ☐11-15 hours ☐16-20 hours ☐over 20 hours
15. **On the average, about how many program listings in each issue of HCM do you key into your computer and use?** ☐None ☐1 ☐2 or 3 ☐4 or more
16. **What peripherals do you currently use?** ☐Disk System ☐Printer ☐Modem ☐Monochrome/Color Monitor ☐Other _____
17. **What do you expect to buy within the next year?** ☐Software ☐Disk system ☐Printer ☐Modem ☐Books ☐Magnetic Media ☐Monochrome/Color Monitor ☐Furniture & Accessories
18. **How much do you expect to spend on computer-related products during the next year?** ☐Less than $25 ☐$25-$49 ☐$50-$99 ☐$100-$249 ☐$250-$490 ☐$500-$999 ☐$1000-$2499 ☐$2500 or more

OPTIONAL: If you would like to help us by participating in a telephone interview, please include your telephone number (____) -_____ here and the most convenient time you can be reached _____:_____ ☐AM ☐PM

FREE

FOR THE ASKING

See Something you like? Want more information?

SEND IT IN ►

Please use these cards for requesting product information only. Do not enclose money with this card. Editorial and Customer Service inquiries should be addressed to:

Home Computer Magazine
1500 Valley River Dr., Suite 250
Eugene, OR 97401

Check the expiration date on these cards prior to mailing.

PASS IT ON ►

Use these convenient inquiry cards to request **free** information about the products or services advertised in this issue of HOME COMPUTER™ magazine.

For information about specific advertisements: Circle the key number on the card that matches the number on the bottom of the ad.

Just drop one of the cards in the mail and your request will be immediately forwarded to the company or companies.

HOME COMPUTER™ magazine

Name _____

Address _____

City _____ St _____ Zip _____

Please circle an entry for each category.

1. Where did you obtain this copy?
   1. Newsstand 4. Bookstore
   2. Subscriber 5. Users Group
   3. Friend 6. Computer Store
2. Which home computer(s) do you own?
   1. Don't presently own a computer.
   2. Apple II/II+ 3. Apple IIe
   4. Apple IIc 5. Commodore 64
   6. VIC-20 7. IBM PC
   8. IBM PCjr 9. TI-99/4A
   10. Other brand
3. Primary use of your home computer
   1. Entertainment
   2. Education
   3. Computer Literacy
   4. Household Management
   5. Job-Related Applications
   6. Business
   7. Other

Annual household income
1. $10,000-14,999
2. $15,000-19,999
3. $20,000-24,999
4. $25,000-29,999
5. $30,000-39,999
6. $40,000-49,999
7. $50,000+

5. Occupation
   1. Professional
   2. Management
   3. Teacher
   4. Student
   5. Other
6. Computer-related purchases within next 12 months
   1. less than $25
   2. $25-49
   3. $50-99
   4. $100-249
   5. $250-499
   6. $500-999
   7. $1,000-2,499
   8. $2,500+
7. I am
   1. Male 2. Female
8. My age is
   1. 14 or younger 4. 35-44
   2. 15-24 4. 45-54
   3. 25-34 6. 65+

PLEASE USE PRIOR TO OCTOBER 31, 1984

CHECK EACH ADVERTISEMENT FOR CORRESPONDING NUMBER AND CIRCLE BELOW.

001	031	061	091	121	151	181
002	032	062	092	122	152	182
003	033	063	093	123	153	183
004	034	064	094	124	154	184
005	035	065	095	125	155	185
006	036	066	096	126	156	186
007	037	067	097	127	157	187
008	038	068	098	128	158	188
009	039	069	099	129	159	189
010	040	070	100	130	160	190
011	041	071	101	131	161	191
012	042	072	102	132	162	192
013	043	073	103	133	163	193
014	044	074	104	134	164	194
015	045	075	105	135	165	195
016	046	076	106	136	166	196
017	047	077	107	137	167	197
018	048	078	108	138	168	198
019	049	079	109	139	169	199
020	050	080	110	140	170	200
021	051	081	111	141	171	201
022	052	082	112	142	172	202
023	053	083	113	143	173	203
024	054	084	114	144	174	204
025	055	085	115	145	175	205
026	056	086	116	146	176	206
027	057	087	117	147	177	207
028	058	088	118	148	178	208
029	059	089	119	149	179	209
030	060	090	120	150	180	**999**

KEY NUMBERS FOR ADVERTISED PRODUCTS ALSO APPEAR IN THE ADVERTISER'S INDEX.

NOTE:

To speed the processing of your inquiry, please provide all the information requested on the card. This will make it easier for the companies to follow up on your request as soon as possible.

HOME COMPUTER™ magazine

Name _____

Address _____

City _____ St _____ Zip _____

Please circle an entry for each category.

1. Where did you obtain this copy?
   1. Newsstand 4. Bookstore
   2. Subscriber 5. Users Group
   3. Friend 6. Computer Store
2. Which home computer(s) do you own?
   1. Don't presently own a computer.
   2. Apple II/II+ 3. Apple IIe
   4. Apple IIc 5. Commodore 64
   6. VIC-20 7. IBM PC
   8. IBM PCjr 9. TI-99/4A
   10. Other brand
3. Primary use of your home computer
   1. Entertainment
   2. Education
   3. Computer Literacy
   4. Household Management
   5. Job-Related Applications
   6. Business
   7. Other

Annual household income
1. $10,000-14,999
2. $15,000-19,999
3. $20,000-24,999
4. $25,000-29,999
5. $30,000-39,999
6. $40,000-49,999
7. $50,000+

5. Occupation
   1. Professional
   2. Management
   3. Teacher
   4. Student
   5. Other
6. Computer-related purchases within next 12 months
   1. less than $25
   2. $25-49
   3. $50-99
   4. $100-249
   5. $250-499
   6. $500-999
   7. $1,000-2,499
   8. $2,500+
7. I am
   1. Male 2. Female
8. My age is
   1. 14 or younger 4. 35-44
   2. 15-24 4. 45-54
   3. 25-34 6. 65+

PLEASE USE PRIOR TO OCTOBER 31, 1984

CHECK EACH ADVERTISEMENT FOR CORRESPONDING NUMBER AND CIRCLE BELOW.

001	031	061	091	121	151	181
002	032	062	092	122	152	182
003	033	063	093	123	153	183
004	034	064	094	124	154	184
005	035	065	095	125	155	185
006	036	066	096	126	156	186
007	037	067	097	127	157	187
008	038	068	098	128	158	188
009	039	069	099	129	159	189
010	040	070	100	130	160	190
011	041	071	101	131	161	191
012	042	072	102	132	162	192
013	043	073	103	133	163	193
014	044	074	104	134	164	194
015	045	075	105	135	165	195
016	046	076	106	136	166	196
017	047	077	107	137	167	197
018	048	078	108	138	168	198
019	049	079	109	139	169	199
020	050	080	110	140	170	200
021	051	081	111	141	171	201
022	052	082	112	142	172	202
023	053	083	113	143	173	203
024	054	084	114	144	174	204
025	055	085	115	145	175	205
026	056	086	116	146	176	206
027	057	087	117	147	177	207
028	058	088	118	148	178	208
029	059	089	119	149	179	209
030	060	090	120	150	180	**999**

KEY NUMBERS FOR ADVERTISED PRODUCTS ALSO APPEAR IN THE ADVERTISER'S INDEX.

PLEASE AFFIX SUFFICIENT POSTAGE

HOME COMPUTER™ magazine

ATTN: READER SERVICE DEPT.
P.O. BOX 2992
CLINTON, IOWA 52735

LEARN MORE ABOUT PRODUCTS THAT ARE ADVERTISED IN THIS MONTH'S ISSUE OF

HOME COMPUTER™ magazine

FREE PRODUCT INFORMATION

FREE PRODUCT INFORMATION

FREE PRODUCT INFORMATION

PLEASE AFFIX SUFFICIENT POSTAGE

HOME COMPUTER™ magazine

ATTN: READER SERVICE DEPT.
P.O. BOX 2992
CLINTON, IOWA 52735

FREE INFORMATION about the products advertised in this issue of

HOME COMPUTER™ magazine

See other side for directions

WHAT'S A FLIPPER/SNAPPER™?

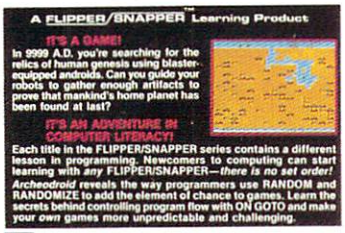
IT'S A 3-IN-ONE VALUE PACK!

- *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.*

TI-99/4A cassette versions now available! Apple, Commodore, and IBM disk & cassette versions coming soon.

DISCOVER HOW MUCH FUN LEARNING CAN BE!

MAKE YOUR SELECTION



☐ **Archeodroid™**

☐ **Midnight Gunner™**

☐ **Simon's Saucer™**

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

- ☐ Please send me 1 FLIPPER/SNAPPER for $10.95 plus $2.00 shipping and handling.
- ☐ Please send me 2 FLIPPER/SNAPPERs for $19.95 plus $3.00 shipping and handling.
- ☐ Please send me all 3 FLIPPER/SNAPPERs for $29.95 plus $3.50 shipping and handling.

BE SURE TO SELECT YOUR TITLE CHOICE(s) BY CHECKING APPROPRIATE BOX(es) AT LEFT.

Bill my ☐ VISA ☐ MasterCard

Expiration Date Tel. No.

Signature

☐ **Check or Money Order Enclosed**

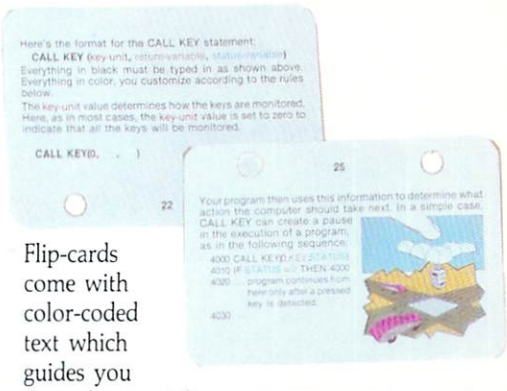
MUST BE IN U.S. FUNDS DRAWN ON A U.S. BANK

Foreign surface add $3.00 for shipping.

Enclose check, money order or credit card information to:

WORLD CLASS SOFTWARE
P.O. Box 5537
Eugene, OR. 97405

Use VISA/MasterCard and order Toll-Free
1-800-828-2212
In Oregon, Alaska, and Hawaii
telephone (503) 485-8796



Flip-cards come with color-coded text which guides you through essential programming practices and concepts in an easy-to-follow lesson format.

WORLD CLASS™ SOFTWARE

An Emerald Valley Company

Circle 178 on Reader Service Card.

Commodore Software– The Best Game in Town.

...Take on the world, toughen up your trigger finger and fire away...

Commodore is the best computer value in town...at home, at school and at work... with our exciting, easy to use, inexpensive VIC 20 and C64 computers.

We're fast becoming the best game in town when it comes to entertainment for the whole family...and at affordable prices.

THE BEST ARCADE IN TOWN can be in your own home with our exciting, faithful reproductions of the best of Bally Midway arcade games. Our **Kickman,** (which just received a coveted "Electronic Games" award for an arcade translation) lets you steer the unicycle to catch the falling objects, as they fall quicker and quicker!!

Gorf, Lazarian, and Omega Race give you the best in classic space action against the one-eyed leviathon, the droids or the evil Empire.

In **The Wizard of Wor** you attempt to defeat the Wizard and the Warriors, fighting your way through to the end. With the new Commodore "MAGIC VOICE"... It talks back to you too!!

You commandeer the fleet at sea with our version of **Seawolf,** and become the master tactician as you battle "it out" with enemy fleet.

Clowns and **Blueprint** round out our arcade entertainment package to keep your fingers nimble and your mind in gear.

commodore COMPUTERS

First In Quality Software

See your local dealer now... He's got the best game in town... just for you.

Circle 200 on Reader Service Card.