Word Processing In The Home

COMPUTE!'s GAZETTE™

$2.50
January 1984
Issue 7 Vol. 2, No. 1
02220 $3.25 in Canada

For Owners And Users Of **Commodore VIC-20**™ And **64**™ Personal Computers

Computing For Families

Reviews of some of the best educational games on the market for the whole family to enjoy. Evaluated by our crack review panel—columnist Fred D'Ignazio and his own family.

64 ELECTRONIC NOTEPAD

Ever lost the instructions for using a favorite program? Electronic Notepad keeps the information where it belongs—on the same disk with your programs.

SpeedScript Word Processor For VIC And 64

Turn your computer into a superior writing tool with *SpeedScript*, a ready-to-type program with all the major features of commercial word processors. Written entirely in machine language for the 64 and expanded VIC.

Tetracrystals Of Veluria & Cave-In:

Two Unique Games For 64 And VIC

Create multicolored crystals which grow and evolve in *Tetracrystals*, an unusual nonviolent game for VIC and 64. Or rescue trapped miners in *Cave-In*, an exciting 3-D maze game for the VIC.

HARDHAT CLIMBER

Thrilling Action Game For VIC And 64

All you have to do is pick up the toolboxes scattered throughout the building. Seems easy enough. But watch out for the broken ladders, holes in the floor, and tumbling barrels....

Also In This Issue

The Beginner's Corner

64 BASIC Aid

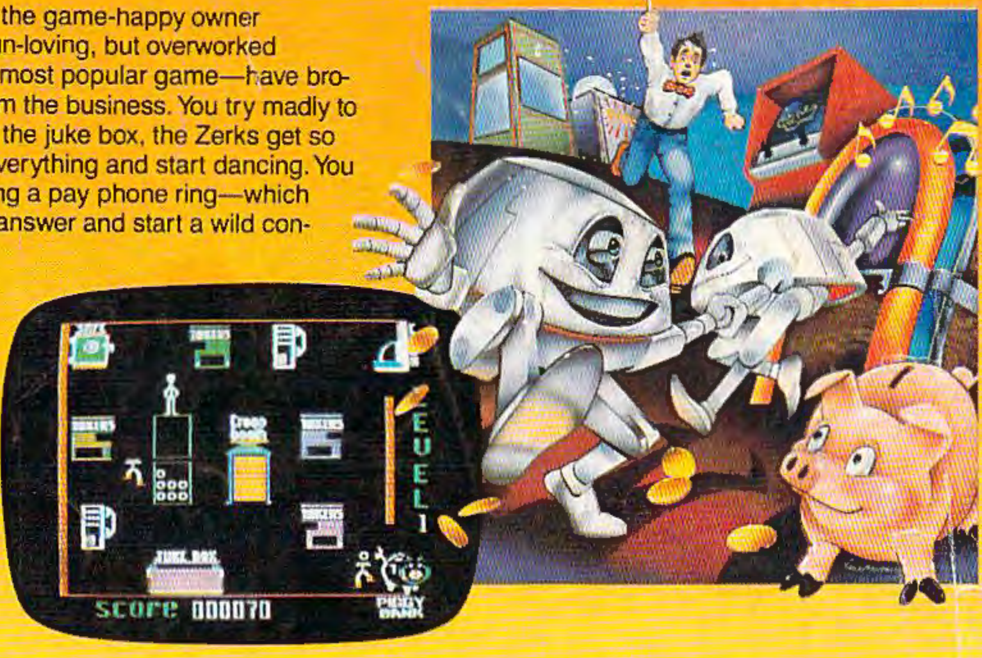
Graph Plotter For VIC And 64

HOTWARE: Best-Selling Programs

0 01
71486 02220

A million laughs

SPARE CHANGE™ You are the game-happy owner of the Spare Change Arcade. Two fun-loving, but overworked Zerks—the main characters in your most popular game—have broken loose and are trying to retire from the business. You try madly to stop them. If you can get a coin into the juke box, the Zerks get so caught up in the music, they drop everything and start dancing. You also try popping popcorn and making a pay phone ring—which immediately makes the Zerks stop, answer and start a wild conversation. If you "win" the game, there are rib tickling cartoons by the Zerks to reward your efforts. It's a game full of sight gags, surprises and comedy. From the best. Brøderbund! **For the Apple® II/II+/IIe, Atari® Computers, and Commodore 64™ in disk format.**



and an endless challenge

LODE RUNNER™ Here's a game that will never stop challenging you. That's because Lode Runner is more than a spellbinding, fast-action game with its 150 different mind-boggling game screens. Lode Runner is also an easy-to-use Game Generator that lets you create your own games. Without any knowledge of programming, you can easily design unique Lode Runnner screens, then bring them to action-packed life. You will maneuver through scene after scene, running, jumping, drilling passages and outfoxing enemy guards in a secret underground hideaway as you pick up chests of gold stolen from citizens of the Bungeling Empire. There's no end to the thrills, chills and challenge. Of course, it's from Brøderbund! **For the Apple® II/II+/IIe; Atari® Computers; Commodore 64™ (disk and cartridge); VIC-20™ (cartridge); IBM® PC.**

Ask your Brøderbund dealer for sneak previews.

Brøderbund Software **Discover the Difference**

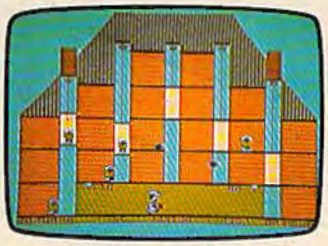
17 Paul Drive San Rafael, CA 94903

First Star Has 4 New Games.

Fernando Herrera, designer of ASTRO CHASE™ and our design team again define "State of the Art." Superior graphics, real-time animations,™ multiple screens, intermissions, arcade-quality sound, innovative gaming, challenge and excitement— we deliver it all!

THE BAD NEWS? You can't play them all at once.

BOING!™
Designed by Alex Leavens & Shirley A. Russell
Atari VCS 2600



BRISTLES™
Starring Peter the Painter
Designed by Fernando Herrera
Atari Home Computers
Commodore Computers

FLIP and FLOP™
Designed by Jim Nangano
Atari Home Computers
Commodore Computers

PANIC BUTTON™
TRS-80 Color Computer by Paul Kanevsky
Vic-20 Home Computer by Wayne Lam

FIRST STAR SOFTWARE INC

"When being first counts...we're number one!"
22 East 41st Street, New York, NY 10017
Distributor and Dealer Inquiries Invited/212 532-4666

"Make note of that name, Electronic Arts.
It may well set the standard for sophisticated
entertainment software in the '80's."

—Creative Computing.

ELECTRONIC ARTS COMES TO THE COMMODORE.

C64, Ap, At

HARD HAT MACK™

by Michael Abbot and Matthew Alexander

"An outstanding, state-of-the-art game. Hard Hat Mack is the blue collar hero of this three screen climbing game and his tasks are far from simple."
—Arcade Express

"If you put Hard Hat Mack in the ring with all other games on the market, it would win most bouts in the opening seconds of the first round. It's that good."
—Softalk

ARCHON™

by Free Fall Associates

"No review could possibly do more than hint at the manifold excellence of Archon. It is truly a landmark in the development of computerized strategy games."
—Video

"...you're bound to fall for it. Imagine a chess game in which you can cast spells."
—Creative Computing

"...the offices of A.N.A.L.O.G. echo with the searing roar of dragon-fire and shouted obscenities from angry players. Archon turns friend against friend and inspires grudges that can last for days. What better compliment can you give to a computer program."
—A.N.A.L.O.G.

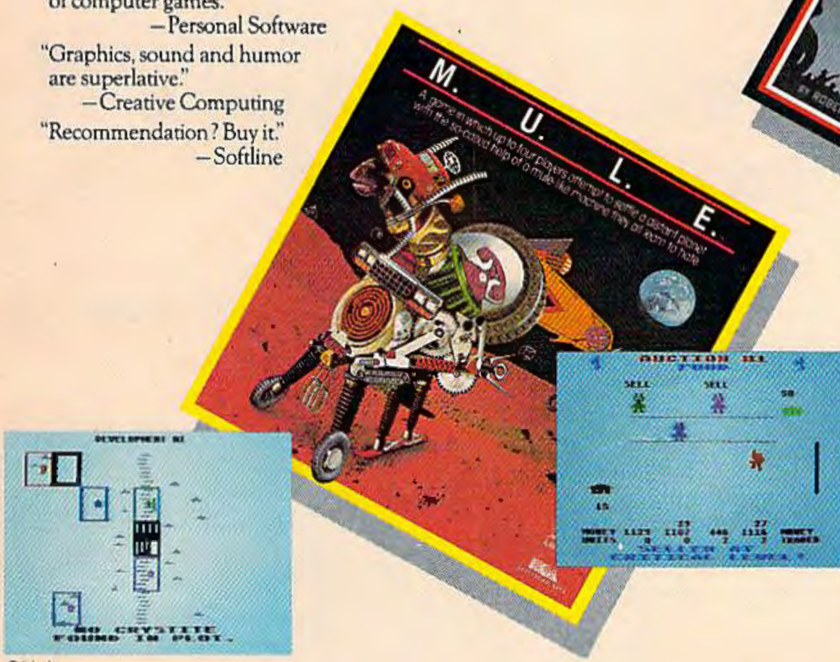
C64, Ap, At

Ap, C64, At, signify availability for Apple II, II+, IIE, Commodore 64, or Atari home computers with disc drive. All screen shots represent Commodore 64. Apple is a registered trademark of Apple Computer, Inc. Commodore 64 is a registered trademark of Commodore Business Machines, Inc. Atari is a registered trademark of Atari, Inc., a Warner Communications Company

MURDER ON THE ZINDERNEUF™
by Free Fall Associates
"Whodunit fans, drop your Agatha Christie and come running. This is your game! The graphics are among the most colorful and attractive I have seen in any game." —Antic

C64, At

M.U.L.E.™
by Ozark Softscape
"A must-have."
—Electronic Fun
"May well become the Monopoly of computer games."
—Personal Software
"Graphics, sound and humor are superlative."
—Creative Computing
"Recommendation? Buy it."
—Softline



C64, At

PINBALL CONSTRUCTION SET™
by Bill Budge
"The best program ever written for an 8-bit machine."
—Steve Wozniak
"A fully integrated graphics-oriented design tool that lets you create your own video pinball games, required study for all serious software authors."
—A.N.A.L.O.G.
"A tour de force."
—Creative Computing

C64, Ap, At

ELECTRONIC ARTS™
Home Software for the Commodore 64

Look for our other Commodore 64 titles like AXIS ASSASSIN,™ WORMS?™ and THE TESSERACT STRATEGY™ at your favorite computer store, software center and at fine department stores throughout the country. For more information about these and other Electronic Arts products, write or call us at 2755 Campus Drive, San Mateo, CA 94403 — (415) 571-7171.

If you're all thumbs at the keyboard, MasterType™ will make you a computer whiz.

MasterType #1 on everybody's list... the first step in computer literacy.

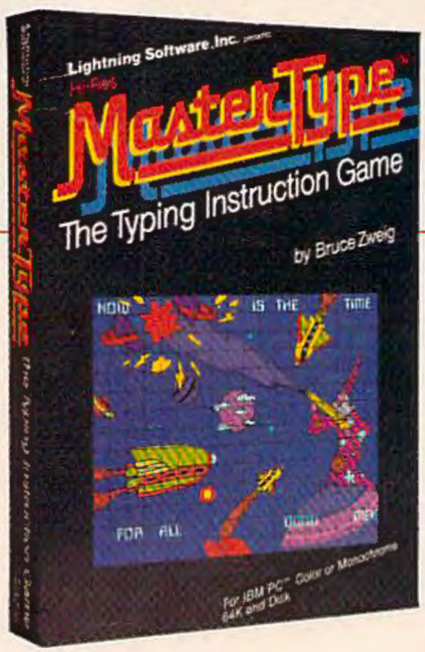
MasterType delivers. It is the one and only typing program that dares to be fun without being intimidating. It combines the fast action of video games with the best typing skills development techniques available. The result? Highly motivated and enjoyable learning.

After each of the 18 action-packed program segments, you'll see how you measure up. And you'll become so caught up in the action that before you know it, you'll become a master at the keyboard, calling all the shots.

On disks for
Apple®, Atari®, Commodore 64® $39.95
IBM-PC® $49.95

MasterType is part of the growing Scarborough family, which includes Songwriter and Picture-Writer, as well as the forthcoming software for home management, science and business games.

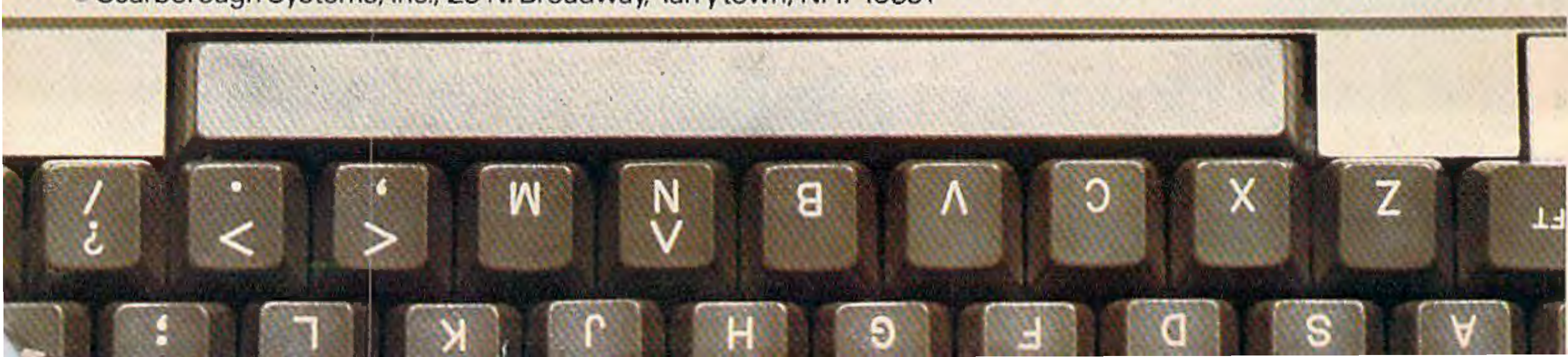
New
Atari® and Commodore 64® cartridges.



The Scarborough System.

You'll grow with us.

© Scarborough Systems, Inc., 25 N. Broadway, Tarrytown, N.Y. 10591



January 1984 Vol. 2, No. 1

FEATURES

GAMES

REVIEWS

EDUCATION/HOME APPLICATIONS

PROGRAMMING

DEPARTMENTS

PROGRAM LISTINGS

* = General, **V** = VIC-20, **64** = Commodore 64.

COMPUTE!'s Gazette is published twelve times each year by **COMPUTE!** Publications, Inc., Post Office Box 5406, Greensboro, NC 27403 USA. Phone (919)275-9809. Editorial offices are located at 505 Edwardia Drive, Greensboro, NC 27409. Domestic subscriptions: 12 issues, $20. Send subscription orders or change of address (P.O. Form 3579) to Circulation Dept., *COMPUTE!'s Gazette*, P.O. Box 5406, Greensboro, NC 27403. Second class application pending at Greensboro, NC 27403 and additional mailing offices. ISSN 0737-3716.

THE EDITOR'S *notes*

An End And A Beginning

The Friday, October 28, announcement by Texas Instruments that they were withdrawing from the home computer industry was met with mixed emotions. While we can't speak directly for the hundreds of thousands of TI owners, we're certain there was disappointment and chagrin. As recently as 14–16 months ago, one highly regarded industry analyst was touting the TI product line as becoming the leader in the industry. Oh, well. We'll be curious to see what TI's promise of continued service and support turns into.

One thing that's noted below in "The Beginning" portion of this editorial regarding IBM's announcement of PCjr is that they've adopted a strategy of open architecture. Third-party developers will be assisted and encouraged in the access to information necessary to help them develop hardware, cartridges, software, etc. This was not the TI approach during product marketing; rather, they chose to make their marketing channels proprietary in many ways, to force vendors to work through them.

Given that many software vendors will probably turn from the TI in favor of other, more active markets, we wonder if TI will release vendors from this restriction. We anticipate that the strong and active TI user groups will be able to maintain support for some time, even if the level from TI begins to decline. Given the merchandising routes used by TI, we expect that support products will be strong sellers through December, and then begin to disappear from many of the single product outlets. After all, no one can realistically expect the local drugstore that sells TI to continue to maintain and rotate TI software and new products from the third-party market after the machine is no longer for sale.

In closing, the news wasn't met negatively by Wall Street.... Within just two days of the announcement, TI's stock rose by almost 30 percent. We assume that TI will think long and hard about any future entries into the home computer market after their several abortive tries since 1980.

The Beginning

IBM's November 1st announcement of PCjr was long awaited, eagerly watched, and disappointing to some. As a home computer, the unit(s) are impressive, powerful, restrained as breakthroughs go, and expensive. All things considered, though, we can be quite confident that PCjr will make a major mark in next year's marketplace. Our editors are hard at work developing materials in support of PCjr (we'll be adding both PC and PCjr to our sister publication COMPUTE!), and hoping anxiously that some kind third-party vendor will quickly develop a keyboard designed for touch typists. At a glance: bottom line PCjr with 64K and cassette BASIC: $689.00 plus $40 per joystick (?!), $30 for a cassette cable, $30 for RF modulator, etc. If you'd like the expanded PCjr with its one (and only one may be used) disk drive, you'll start at $1259. But, as with all top-of-the-line products and prices, you can expect full service, support, and a tremendous amount of sophisticated IBM and third-party software. And we project it's a reasonable bet that IBM won't pull out of the marketplace. Beyond the concern over the keyboard is the lack of extended sound and graphics capabilities on the bottom-line unit. Many of these capabilities can be added by going to the extended BASIC that is available on a $75 plug-in cartridge. But apparently sprites don't exist, and color isn't as extensive as that on the 64 (although resolution is higher).

And in closing, one COMPUTE! pundit had this remark about the new PCjr: "If I could interface it with my 64 I could have great sound and graphics...."

Happy new year! from COMPUTE! Publications.

Robert Lock

Editor In Chief

WordPro 3 Plus™/64

The #1 Selling Word Processor for the Commodore 64™

WordPro 3 Plus™/64 and SpellRight Plus™ provide a total word processing solution for the Commodore 64™ which gives you:

- ★ Sophisticated Word Processing
- ★ Built-in Mail Merging for Form Letters
- ★ Math Functions for Column Totals
- ★ Fast and Complete Spell Checking via SpellRight Plus
- ★ A Super Value (two programs) for Only $99.95!

WordPro and SpellRight are both specifically designed for the novice user with no computer or word processing experience whatsoever. And with over 40,000 WordPro versions sold, you can be sure that WordPro is a very sophisticated word processor loaded with powerful features including: Transfer, Insert, Delete, and Rearrange Text, Auto Page Numbering, Math Functions, Headers, Footers, Global Search and Replace, the Ability to Create Multiple Personalized Letters and Documents, and much more. WordPro can create documents of virtually any length and will print up to 165 columns wide. You get all of this PLUS fast and complete spell checking using SpellRight Plus!

SpellRight Plus locates and highlights misspelled words and then allows you to quickly correct the misspellings — improving the quality of your letters and reports.

And, best of all, WordPro and SpellRight's powerful arsenal of features can be put to use almost immediately — by even the novice user. So whether you're a student, professional writer, in business, education or a hobbyist, you'll quickly become a WordPro Pro!

Both WordPro and SpellRight Plus are also available separately at popular computer outlets nationwide.

Invest in the best . . . WordPro Plus. In a class by itself.

Professional Software Inc.

51 Fremont Street
Needham, MA 02194
(617) 444-5224
Telex: 951579

Dealer and Distributor inquiries are invited.

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

Publisher Gary R. Ingersoll
Editor in Chief Robert C. Lock
Director of Administration Alice S. Wolfe
Senior Editor Richard Mansfield
Managing Editor Kathleen E. Martinek
Art/Design Director Georgia Bikas Davis

Editorial
Tom R. Halfhill, Gazette Editor; Stephen Levy, Editor, COMPUTE! Books Division; Gail Walker, Production Editor; Ottis R. Cowper, Technical Editor; Charles Brannon, Program Editor; Tony Roberts, Assistant Managing Editor

Assistant Editors
Lance Elko, Dan Carmichael (Submissions), Gregg Keizer (Books), John Krause (Technical), Kathy Yakal, Editorial Assistant (Features); Randall Fosner, Editorial Assistant (Books)

Editorial Programmers
Patrick Parrish (Supervisor), Gregg Peele (Assistant), Jeff Hamdani, Kevin Martin, Chris Poer

Technical Assistant
Dale McBane

Programming Assistant
Mark Tuttle

Copy Editing/Proofreading
Juanita Lewis (Assistant), Becky Hall, Linda Shaw, Martha Banks

Administrative Staff
Vicki Jennings, Laura MacFadden, Julia Fleming

Associate Editors
Jim Butterfield (Toronto), Harvey Herman (Greensboro), Fred D'Ignazio (Roanoke), David Thornburg (Los Altos), Bill Wilkinson (Cupertino)

Production
Irma Swain, Assistant Production Manager; De Potter, Mechanical Art Supervisor; Terry Cash, Debi Thomas, Typesetting

Artists
Leslie Jessup, Cindy Mitchell (Publications), Janice Fary, Debbie Bray (Books); Todd Heimarck, Promotions Assistant; Harry Blair, Illustrator

Operations/Customer Service
Carol Lock, Manager; Patty Jones, Customer Coordinator; Assistants: Chris Patty, Chris Gordon; Fran Lyons, Dealer Coordinator; Assistants: Gail Jones, Sharon Minor

Customer Service Staff
Dorothy Bogan, Supervisor; Rhonda Savage, Lisa Flaharty, Anita Roop, Sharon Sebastian, Debi Goforth, Jenna Nash; Operators: Cassandra Robinson, Mary Sprague
Jim Coward (Warehouse Manager), Larry O'Connor, Chris Cain, Dai Rees, Jack McConnell

Data Processing
Leon Stokes, Manager; Joan Compton, Assistant

Accounting
Paul J. Megliola, Manager; James M. Hurst, Comptroller; Assistants: Linda Miller, Doris Hall; Staff: Anna Harris, Emilie Covil, Anne Ferguson

Advertising Sales
Andy Meehan, National Sales Manager; Patti Williams, Production Coordinator; Bonnie Valentino, Accounting Coordinator; Rosemarie Davis, Sales Assistant

Sales Representatives

Jerry Thompson	415-348-8222
Phoebe Thompson	408-354-5553
JoAnn Sullivan	619-941-2313
Ed Winchell	213-378-8361
Harry Blair	919-275-9809

Jules E. Thompson, Inc.
National and Canadian Sales Representatives
1290 Howard Avenue, Suite 303
Burlingame, CA 94010

Address all advertising materials to:
Patti Williams, COMPUTE!'s GAZETTE,
505 Edwardia Drive, Greensboro, NC 27409

Sales Offices, The Thompson Company

New England	617-720-1888
Mid-Atlantic	212-772-0933
Southeast	919-275-9809
Midwest	312-726-6047
Texas	713-731-2605
Northwest	408-354-5553
Northern CA	415-348-8222 or 408-354-5553
Southern CA	619-941-2313 or 213-378-8361
Nevada, Arizona	619-941-2313
New Mexico	213-378-8361
Colorado	303-595-9299

COMPUTE! Publications, Inc., publishes
COMPUTE! **COMPUTE! Books** ***COMPUTE!'s Gazette***

Corporate Office:
505 Edwardia Drive, Greensboro, NC 27409

Mailing Address:
Post Office Box 5406, Greensboro, NC 27403

Telephone: 919-275-9809
Office Hours: 8:30 AM to 4:30 PM Monday-Friday

Chief Executive Officer Robert C. Lock
President Gary R. Ingersoll
Director, Finance & Planning Paul J. Megliola
Comptroller James J. Hurst
Executive Assistant Debi Nash
Assistant Carol Dickerson

Subscription Information
COMPUTE!'s Gazette Circulation Dept.
P.O. Box 5406, Greensboro, NC 27403

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

COMPUTE!'s Gazette
Subscription Rates

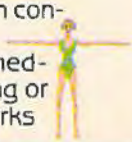
(12 Issue Year): US (one year) $20. Canada, Mexico and Foreign Surface Mail $25. Foreign Air Mail $45.

The COMPUTE!'s Gazette subscriber list is made available to carefully screened organizations with a product or service which may be of interest to our readers. If you prefer not to receive such mailings, please send an exact copy of your subscription label to: COMPUTE!'s Gazette, P.O. Box 961, Farmingdale, NY 11737. Include a note indicating your preference to receive only your subscription.

Spinnaker Aerobics™. The more you do, the less you have to show for it.

Spinnaker's new computer fitness program makes shaping up fun to do. And makes you feel terrific.

AEROBICS gives you everything you need. Warm-ups, stretches, aerobics, cool-downs. It lets you work on overall fitness. Allows you to concentrate on conditioning specific parts of your body. Or both.



Best of all, you can exercise on your own schedule. In your own home. For as long or as little as you like. Whatever works for you.

Whether you're a beginner or already in great shape, you'll love working out with Spinnaker AEROBICS. Which means you'll do it more often. And have even less to show for it.

AEROBICS is compatible with Apple,® Atari,® and Commodore 64™ computers.

SPINNAKER™
We make learning fun.

GAZETTE FEEDBACK

EDITORS AND READERS

Do you have a question or a problem? Have you discovered something that could help other VIC-20 and Commodore 64 users? Do you have a comment about something you've read in COMPUTE!'s GAZETTE? We want to hear from you. Write to Gazette Feedback, COMPUTE!'s GAZETTE, P.O. Box 5406, Greensboro, NC 27403.

1526 Printer Recall

I have recently purchased a Commodore 1526 printer for my Commodore 64 computer system. I have found that I cannot load programs from my 1541 disk drive while the 1526's power switch is on. If the power switch is on and I attempt to load the program, the system locks up after a short period of time and the only way I can reset the system is to turn off the computer. Also, programs I have purchased which require repeated accessing of sequential or relative files will lock up the system if the printer is on.

The dealer who sold me this equipment assured me that this was normal operation. However, I wrote to a software company complaining that their software was not working correctly with my printer and they advised me that the 1526 printer and the 1541 disk drive were incompatible. I have written to Commodore four different times and have received no reply.

Are you aware of any incompatibility problem between the 1526 printer and the 1541 disk drive? If so, could you please explain what the problem is? Do you have any idea what Commodore plans to do to resolve this problem?

Gary L. Martin

The recently introduced Commodore 1526 printer does indeed suffer from serious problems when used with the 1541 disk drive—or any device on the serial port. Commodore has recalled the 1526 from dealers and instructed them to accept returns from any customers experiencing problems.

The 1526 is an 80-column dot matrix printer, similar to the 4023 printer that has been available for the Commodore PETs and CBMs. The 1526 appeared on the market briefly, then rapidly disappeared. According to a Commodore spokesperson, the 1526 suffers from a "firmware problem" that interferes with other devices plugged into the serial port (such as the 1541 disk drive). One Commodore dealer wrote to us saying that in some cases, the problem can be helped if the equipment is switched on in a certain order (in general, turn on the 64, the disk drive(s), and the 1526; see last month's "Gazette Feedback").

If this does not help, we recommend returning the printer to your dealer for a refund. It is not normal operation for any computer system to lock up when correctly interfaced peripherals are being used. At this writing, Commodore does not know when the 1526 will be fixed and remarketed. Perhaps it will be available again by the time you are reading this.

Reruns For Automatic Proofreader?

Before I received the October 1983 issue of COMPUTE!'s GAZETTE, I had many problems getting programs that I typed in from the magazine to come out right. When I read and used the "Automatic Proofreader" it did help me, but only with the programs with the REM statements [Proofreader checksum numbers]. If I used this program to check an earlier program listed in your magazine [without the checksum numbers], I could not understand how to check those lines.

Can you tell me how I could use this helpful checksum program with these other programs? How does it work, and how can I figure out the REM numbers of these other programs? Do you have future plans to relist the earlier programs listed without the checksum numbers?

Jeff Cherkis

In the September issue you asked for feedback on the GAZETTE. First I'd like to say that once in a while a magazine jumps out in front of the pack, sometimes by design and sometimes by doing something lucky. The GAZETTE did it with the program "Proof-

Introducing Snooper Troops™ detective series. Educational games that turn ordinary homes into Sherlock homes.

Where can you find educational computer games that your kids will really enjoy playing?

Elementary, my dear Watson, from Spinnaker.

Our SNOOPER TROOPS™ detective games are fun, exciting and challenging. And best of all, they have real educational value. So while your kids are having fun, they're learning.

As a Snooper Trooper your child will have a great time solving the mysteries. But it will take some daring detective work. They'll have to question suspects, talk to mysterious agents, and even search dark houses to uncover clues.

Luckily, the program provides your kids with everything they need: like a SnoopMobile, a wrist radio, and a SnoopNet computer.

SNOOPER TROOPS detective games help your children learn to take notes, draw maps, organize and classify information and they help develop vocabulary and reasoning skills. All while your kids are having a good time.

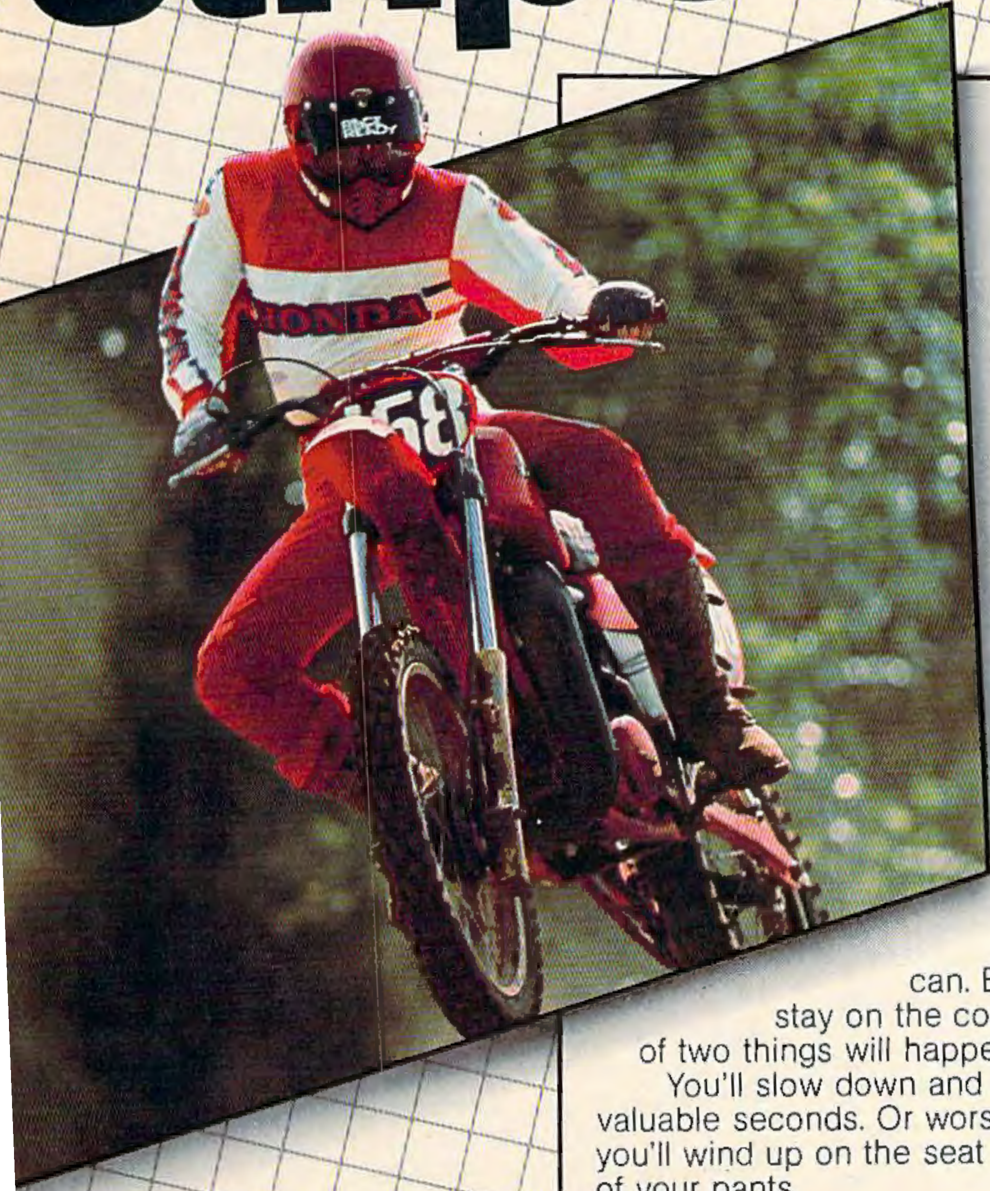
So if you want to find educational games that are really fun, here's a clue: ask your local retailer for SNOOPER TROOPS computer games.*

*Available in disks for IBM,® Atari,® Apple,® Commodore 64.™

SPINNAKER™

We make learning fun.

stripes.



There are those who say that motocross asks more from the athlete than any other sport.

That may or may not be true. But this much *is* true.

Motocross! is unlike any computer game you've ever come across.

At the start/finish line you check your gauges, gun your engine and surge onto the course.

Through the corners, up the hills, into the ruts and down the straights you maneuver for position and try to beat the clock.

Stay ahead if you can. But whatever you do, stay on the course. If you don't, one of two things will happen.

You'll slow down and lose valuable seconds. Or worse, you'll wind up on the seat of your pants.

Who says computers don't like to get dirty?

Developed by Jim Rupp
Available on diskette
for your Commodore 64™

TRONIX™

8295 South La Cienega Blvd.
Inglewood, CA 90301 (213) 215-0529

reader" by Charles Brannon in the October issue. What more can I say—*fantastic*—and now for a suggestion:

Print *just* the line number and checksum for *all* of the programs in your previous issues. Example:

120–147
130–121
etc.

Why bother? Your readers will love you and you'll get reader loyalty.

Stuart B. Wahlberg

We have received many letters from readers complimenting the Automatic Proofreader, including some letters from people who said they had never got a program to work correctly until they used the Proofreader. Almost every letter requested checksum numbers for programs previously published in COMPUTE!'s GAZETTE. *Some people wanted to know how to compute their own checksum numbers for these earlier programs; they didn't understand why the checksums appear inconsistent (i.e., short program lines sometimes have large numbers while long lines sometimes have small numbers).*

We also received a few letters from readers who said the Proofreader doesn't work and neither do the programs they enter with it. We'd like to take this opportunity to discuss possible problems that may be encountered when using the Proofreader to enter programs, and to address your other questions and comments about our program listings as well.

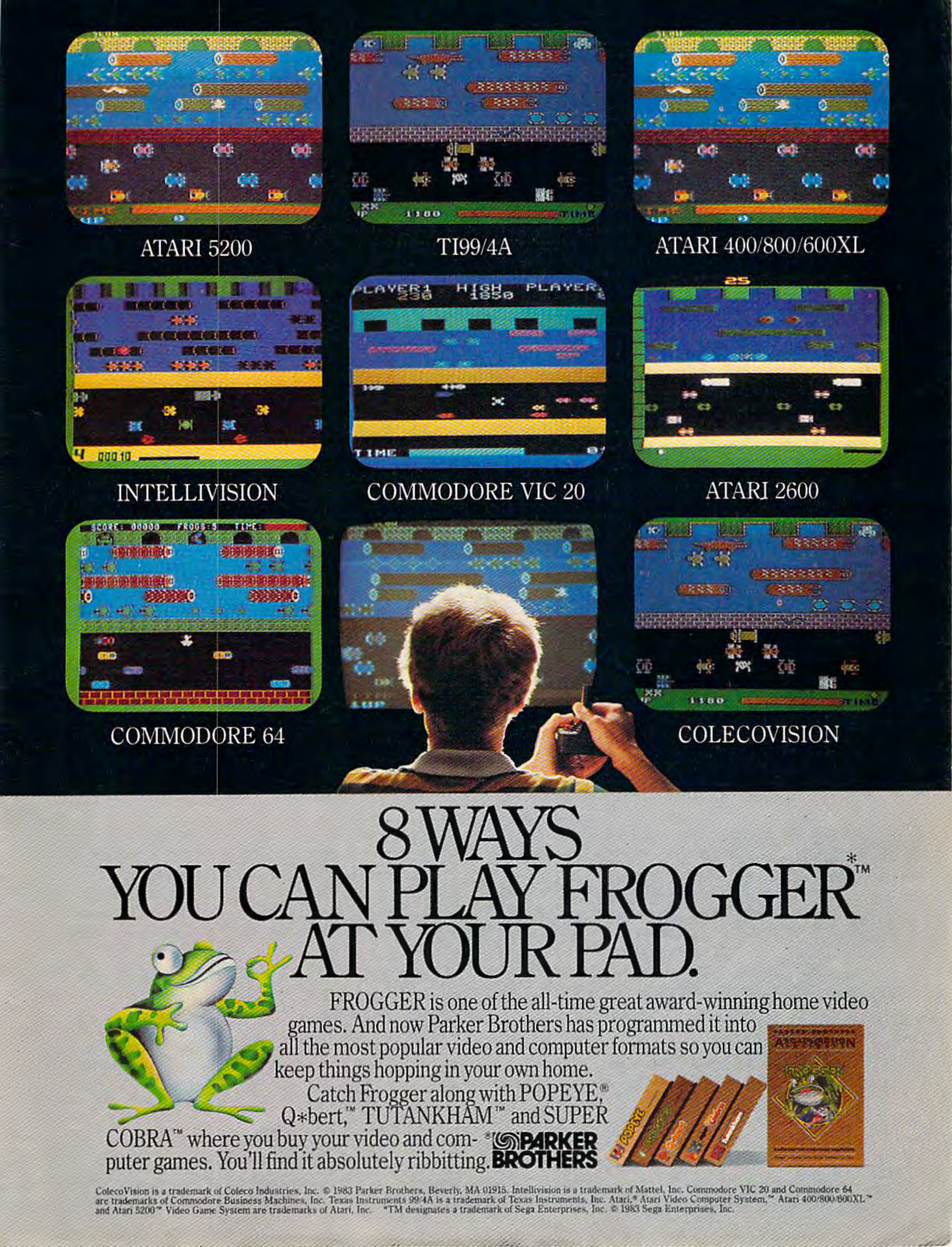
First of all, the Automatic Proofreader does *work. Some VIC-20 tape users had problems reLOADing programs entered with the first version of the Proofreader (see November "Bug-Swatter" and November/December "Automatic Proofreader"). But even this problem never affected the typing or checking of the programs, and it was immediately corrected in the next version of the Proofreader. The Proofreader repeatedly passes all in-house testing, and most readers we hear from have used it with success.*

Readers experiencing problems with the Proofreader should carefully check their typing of the Proofreader program; as we noted in October, unfortunately it can't check itself *(although the current version does check for errors in the DATA statements). If you make a subtle error when typing the Proofreader, it can cause incorrect results when using it to check other programs. A couple of readers who had trouble with the Proofreader saw no difference between the VIC-20 and Commodore 64 versions published in the October issue and concluded that we mistakenly published the same version twice. Both versions are very similar. However, they are not identical. The difference is the fifth DATA element in line 220. To reduce confusion, we rewrote the Proofreader so the same version now works on both computers.*

Assuming the Proofreader program itself has been entered correctly, we have traced most of the problems some readers are encountering to three main causes:

- Transposed keystrokes. *Because of the way the Proofreader checksum numbers are computed (see below), the Proofreader cannot detect transposition errors. In other words, if you type PIRNT instead of PRINT, the Proofreader won't know the difference. Of course, this particular typo would result in a ?SYNTAX ERROR AT LINE xxx when the program is run, but other transpositions might not cause a syntax error. The most common example is numbers in DATA statements. If you type DATA 156 instead of DATA 165, the Proofreader still thinks everything is okay. So does the computer. You probably won't get an error message, but the program won't work right. Solution: Be extra alert for transposition errors.*
- Long program lines. *Normally, you can't enter a program line longer than 80 characters on the Commodore 64 or 88 characters on the VIC-20. However, many programmers abbreviate keywords when writing their programs to save typing and memory. When the programs are listed, the abbreviations automatically expand into the full keywords, and lines longer than 80 or 88 characters often result. The only way these lines can be typed from a listing is to use the same abbreviations (see "Simple Answers To Common Questions" elsewhere in this issue). Since the Proofreader cannot handle abbreviations, it cannot accurately check these lines. Solution: Use abbreviations to type long lines and carefully check the typing yourself. Because long lines cause so many problems for so many readers, we are trying to eliminate them wherever possible, and we urge programmers not to use abbreviations unless absolutely necessary.*
- Mistakes in listings. *Theoretically these should never happen. Theoretically. But sometimes they do. We receive letters from some readers who doubt that we test programs before publication, or who doubt that the programs work in the first place. However, we promise that all programs do work and are tested. (For those who still don't believe it, proof can be seen in the screen photos which accompany almost all programs in* COMPUTE!'s GAZETTE—*if the programs don't work, or if we don't try them, where do the screen photos come from?) After testing, the listings are made on a printer directly from disk and then photographed, not retypeset. In theory this should produce a perfect listing of the program.*

But in practice there are about two dozen things that can go wrong, including some in the printing process which are effectively beyond our control (see this month's "Bug-Swatter"). However, most listing problems are *within our control, and we are constantly striving to reduce them to a minimum. If you discover a subtle error in the operation of a program, chances are it escaped our testing. But if a program runs obviously wrong or crashes altogether at the outset, it is a sure sign of a typo—introduced either during the listing process (us) or the typing process (the user). These typos are extremely hard to track down. Upon receiving the first complaints, we immediately test the program*


ATARI 5200
TI99/4A
ATARI 400/800/600XL
INTELLIVISION
COMMODORE VIC 20
ATARI 2600
COMMODORE 64
COLECOVISION
8 WAYS YOU CAN PLAY FROGGER*™ AT YOUR PAD.
FROGGER is one of the all-time great award-winning home video games. And now Parker Brothers has programmed it into all the most popular video and computer formats so you can keep things hopping in your own home.
Catch Frogger along with POPEYE,® Q*bert,™ TUTANKHAM™ and SUPER COBRA™ where you buy your video and computer games. You'll find it absolutely ribbitting.
®PARKER BROTHERS
ColecoVision is a trademark of Coleco Industries, Inc. © 1983 Parker Brothers, Beverly, MA 01915. Intellivision is a trademark of Mattel, Inc. Commodore VIC 20 and Commodore 64 are trademarks of Commodore Business Machines, Inc. Texas Instruments 99/4A is a trademark of Texas Instruments, Inc. Atari,® Atari Video Computer System,™ Atari 400/800/600XL™ and Atari 5200™ Video Game System are trademarks of Atari, Inc. *TM designates a trademark of Sega Enterprises, Inc. © 1983 Sega Enterprises, Inc.

from our archive disks. So far, a program has never failed to run. Unfortunately, all this tells us is that the typo happened sometime after we tested the program, saved it on disk, and made the listing. Unless we can find a discrepancy between our working copy of the program and our published listing, we cannot assume an error. We are often guided in these cases by reader feedback. If we receive a large number of similar complaints about a certain program, we strongly suspect something went wrong at our end. But if some readers tell us a certain program works fine, while others say it does not, it is difficult for us to conclude the first group of readers somehow made a typo that just happened to correct the alleged typo we *made. Solution: If a program does not work, and neither you nor a proven copy of the Proofreader can find an error, write or call us to describe the exact nature of the problem. Perhaps we will have a fix, or can report that other readers are having no trouble with the program.*

Now to address your other questions. The Proofreader, of course, requires you to compare the number which appears at the top of the screen to the checksum number in the program listing. Since previously published programs lack these checksum numbers, you cannot check them with the Proofreader. Nor can you compute your own checksum numbers. The computer which makes our listings automatically generates the checksums by adding the ASCII values of all the characters in a line and storing the sum in a single byte. Since one byte holds the sum, the checksum never exceeds 255. If the sum is greater than 255, the byte "rolls over" past zero. For example, 240 + 20 = 4 (a principle well known to machine language programmers). That explains why some short program lines have large checksum numbers and vice versa. (Incidentally, it also means that there's a tiny chance that two or more *typos in a line could cancel each other out and yield a correct checksum match.)*

Even if you manually computed your own checksums this way, they would be meaningless, since they would be thrown off by any errors in the line. The checksum must be computed from a working version of the program, as our listing computer does.

Several readers have asked us to republish line numbers of earlier programs with just the checksum numbers appended (there is not enough space to reprint the programs and articles in their entirety). That way, you could check for typos in programs you typed in months ago but never got to work. We are considering this and will do so if there is enough demand. Let us know how you feel.

Copyright Questions

I have a few questions about the programs listed in your magazine. Can I photocopy them? Photocopy machines are in libraries and about everywhere else. I once read a news clipping where the courts have ruled that it's okay to photocopy something for your own personal use and files. Is this so? What is, and what is not public domain? Can I use the programs listed in COMPUTE!'s GAZETTE at my place of business as well as my home?

Clarence C. Hogan

Everything in COMPUTE!'s GAZETTE *is copyrighted, and nothing is in the public domain unless specifically stated. This is true of virtually all magazines and books, unless they specify otherwise. This means that programs you type in from a magazine or book which you have purchased are for your personal use. You may not sell the programs in any form, or give copies to people who have not purchased the same book or magazine issue. Both parties are liable if this federal law is broken. Photocopies are fine as long as they are for your personal use. You can use the programs at your place of business with the same restrictions.*

From VIC To 64

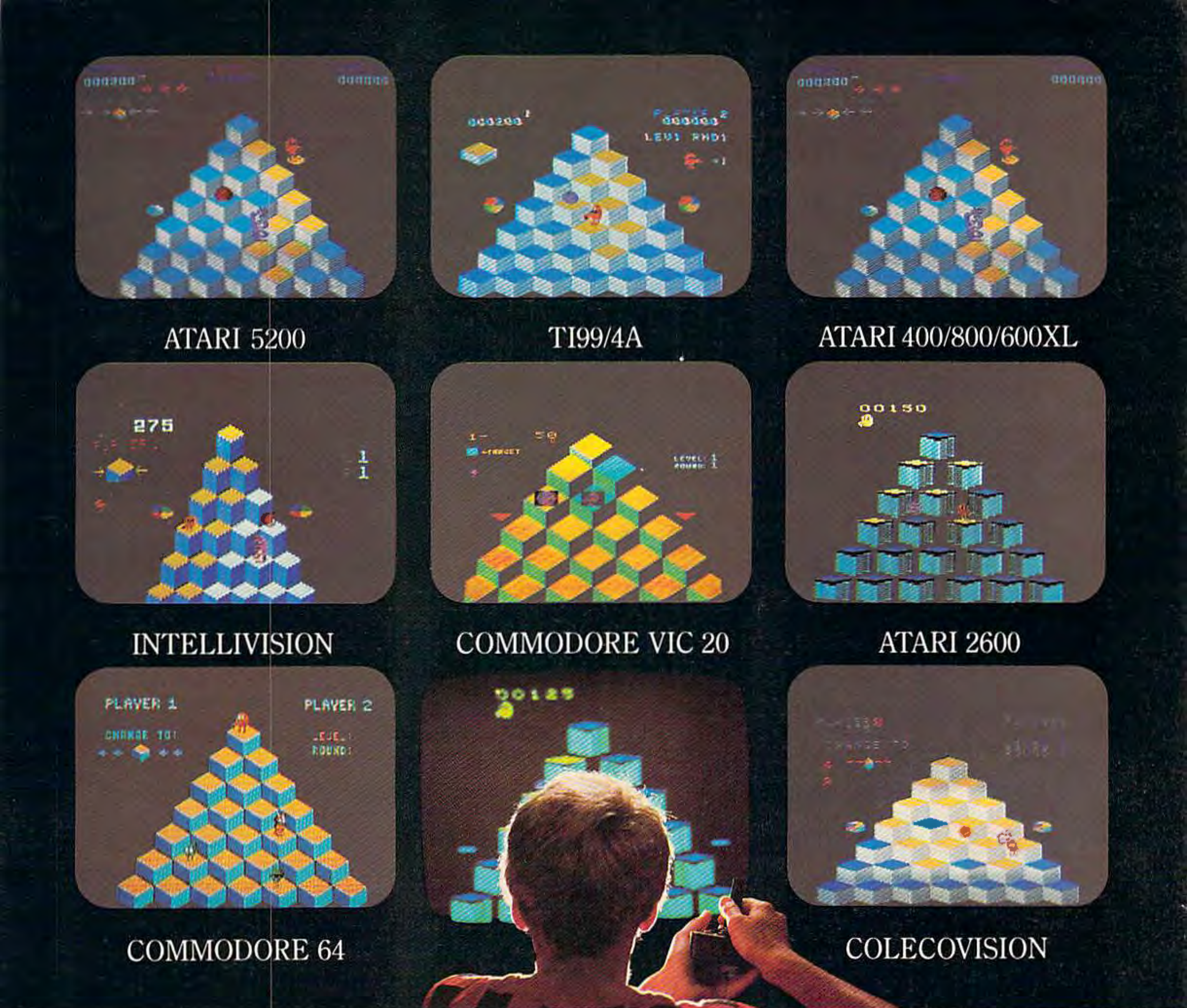
I own a VIC-20, but have decided to purchase a Commodore 64. I would like to know if you could answer some questions. First, are all the cartridges made for the VIC-20 compatible with the 64? Secondly, can machine language be used directly on the 64? I heard that it can be used on the VIC-20, but you're better off buying some kind of software on cartridge. Does the Commodore 64 need any additional software to run machine language easily? Thirdly, do you know where I can write to Commodore to obtain information about software, hardware, maintenance, etc., pertaining to their products? Any information you can give me would be appreciated.

Brian Cummings

No cartridges for the VIC-20 are compatible with the Commodore 64, or vice versa. The cartridge ports (where you plug in the cartridges) are different sizes on the 64 and the VIC. Even if you could make the cartridges fit the slot, the programs encoded in the cartridges would not be compatible.

Neither the VIC nor the 64 needs any additional software to run *machine language programs. Machine language is the native language of all computers—the language with which they "think." BASIC, on the other hand, is a foreign language to computers (just as it is to people) and must be interpreted internally before the computer can understand the instructions. Machine language programs can be loaded and run directly from tape, disk, or cartridge with either the VIC or 64. A machine language program can even be loaded into memory by a BASIC program with the POKE statement. It can then be run with the SYS or USR commands.*

However, to write *machine language programs on a VIC or 64, you generally do need additional software. In theory, you could get by without it by laboriously converting the machine language instructions into decimal numbers by hand and then POKEing them into memory with BASIC. For short routines this might work out. But for more ambitious programs, most people*



HOW TO GET Q*BERT™ OUT OF YOUR SYSTEM.

If you've been wanting to play Q*bert, but haven't been able to find it available for your home system, your time has come. Because now you can keep things hopping with any of these popular home video and computer formats.

Get going to your nearest video store and get Q*bert today. And while you're there, check out Parker Brothers' POPEYE,® FROGGER,™ TUTANKHAM,™ and SUPER COBRA.™ All the great Arcade Action® games, now in all the great home formats.

PARKER BROTHERS

who value their sanity prefer to use a monitor *or an* assembler *to write machine language. Some computers (such as the earlier Commodore PETs and the Apple) have built-in monitors, but consumer computers aimed at the home market generally do not. Most people find that assemblers are the easiest way to write machine language, especially if their previous programming experience is with high-level languages such as BASIC. Monitors and assemblers are available on cartridge, tape, and disk for either the VIC or 64. To learn more about monitors, assemblers, and machine language, see "Machine Language For Beginners," a regular monthly column in* COMPUTE!'s GAZETTE.

To write to Commodore for more information about its computers and other products, use this address:

Commodore Business Machines, Inc.
1200 Wilson Drive
West Chester, PA 19380

For maintenance information, you might try this address:

Commodore Service Center
950 Airport Road
West Chester, PA 19380

An authorized Commodore dealer in your area may also be able to answer your inquiries. You can call 1-408-727-3754 for repair cost and full service information. Commodore also offers a customer assistance number, 1-215-436-4200. As of this writing, Commodore's toll-free customer assistance number is no longer active.

Expanded VIC Memory

I have a Commodore VIC-20 computer and would like to know what you mean when you say (before a long program) "for VIC-20 expanded to 8K." Does this mean total RAM or user RAM? With my Super Expander cartridge I have 8K total RAM (the VIC has 5K, and my expander adds an additional 3K of RAM). I would like to know if I can now run some of your programs which say this.

Steve Medendorp

All VICs have the built-in 5K of Random Access Memory (RAM), so when we say "for the 8K expanded VIC-20," we are referring to the 8K expansion memory only. In other words, you would need an 8K expansion cartridge. Similarly, "16K expanded VIC" means a VIC with two 8K cartridges plugged into a motherboard or one 16K memory expander, and "3K expanded VIC" means the Super Expander or another 3K expansion cartridge is required. Occasionally we publish a program that specifically requires the Super Expander because it adds special graphics commands to the VIC as well as 3K of RAM. We try to make most of the programs we publish run on unexpanded VICs so the greatest number of readers can use them.

VIC Memory Expansion

Is it possible to add more than 24K to the VIC-20?

Charles Q. Berkey, Jr.

Yes and no. The VIC-20 has 4K of Random Access Memory (RAM) built in, at 4096 to 8191 (hex $1000–1FFF), plus 1K for overhead: pointers, the stack, and so on, for a total of 5K RAM. You can buy memory expanders which add 3K, 8K, 16K, or 24K from Commodore or third-party manufacturers.

If you program in BASIC, 24K is the most memory you can add to your 5K VIC. If you use machine language, you can add up to 35K, for a total of 40K.

The first thing you have to remember when you add memory is that a VIC has only one expansion port. That means if you own an 8K expander and want to add 8K more, you have two choices. You can buy memory chips and rewire your expansion cartridge (not recommended unless you know exactly what you are doing). Or you can buy a multiple cartridge board "motherboard" that allows you to plug more than one cartridge into the expansion port. It is similar to an electrical extension cord you might use in your home. Often these motherboards have switches so you can select one or more cartridges which are plugged in. (For example, you could "switch off" the memory expanders without physically unplugging them, in case you want to run a program that is designed exclusively for the unexpanded VIC.)

Memory can be added to the VIC in the following blocks:

3K	1024–4095	($0400–0FFF)
8K	8192–16383	($2000–3FFF)
8K	16384–24575	($4000–5FFF)
8K	24576–32767	($6000–7FFF)
8K	40960–49151	($A000–BFFF)

Adding expansion memory to the VIC can cause complications, however. If you plug in the 3K expander, screen memory (7680–8191 in the unexpanded VIC) remains in the same place (7680–8191). But if you add more than 3K, screen memory moves to 4096–4607. This can result in compatibility problems with some programs written for the unexpanded VIC.

Whether your VIC has 5K or 40K, it wants to put BASIC programs in a continuous section of memory. The BASIC program goes at the bottom, followed by free memory and variables (at the top of memory). When you add 8K or more, any memory below screen memory (4096) becomes invisible to BASIC. That's why only a maximum 24K can be added for BASIC programming.

Once you add memory to locations 8192–32767, the other memory expansion is available only in machine language (or PEEKs and POKEs).

What Is A Utility?

What is a utility program? Does this type of software make it easier to program?

Fred Soderlund

A utility is a program that programmers use for a specific purpose. Many utilities provide new BASIC commands or disk commands. For example, let's say you want to add a menu to a program you have written. If you already have a program that makes menus, you could add it to your program by typing it in line by line. But if you own a utility with an append command, you simply merge the two programs—which results in a lot of saved time.

If both programs use lines 900–1000, you could get around the problem with a RENUMBER utility which changes the line numbers (you would have to RENUMBER before you append). If both programs use the variable DN and R$, you could use a utility which searches the programs and tells you which variables you used and where.

If you use certain routines in many programs (reading the joystick, figuring compound interest, etc.), a utility can save you time.

Some utilities contain certain commands that work only with disk drives or printers. Certain commands will be most useful to a machine language programmer (for example, copying one block of memory to another, or hex to decimal conversions). Other utilities are designed to help you write programs with graphics or sound routines.

Utilities are programming tools. Their usefulness depends on what they do and what you need. If you are just getting into programming, you probably don't need many programming tools. But as you start writing larger and more complex programs, you will find that a collection of the right utilities can save you lots of time and work. Utilities are sold commercially, and many public domain utilities are available for free through local user groups. Also, nearly every issue of COMPUTE!'s GAZETTE *includes ready-to-type program listings for useful utilities.*

HOTWARE

A Look At This Month's Best Sellers And The Software Industry

Kathy Yakal, Editorial Assistant

Commodore 64 Entertainment

This Month		Last Month
1	*Fort Apocalypse* (Synapse)	2
2	*Jumpman* (Epyx)	1
3	*Temple of Apshai* (Epyx)	3
4	*Frogger* (Sierra On-Line)	4
5	*Planetfall* (Infocom)	10
6	*Choplifter* (Creative)	–
7	*Gridrunner* (HesWare)	7
8	*Neutral Zone* (Access)	5
9	*Sword of Fargoal* (Epyx)	6
10	*Shamus* (HesWare)	–

Commodore 64 Home/Business/Utility

This Month		Last Month
1	*WordPro 3 Plus/64* With *Spell Right* (Professional)	1
2	*Quick Brown Fox* (Quick Brown Fox)	2
3	*M File* (M Soft)	11
4	*Money Manager* (Timeworks)	5
5	*Practicalc* (Computer Software Associates)	4
6	*Calc Result* (Handic)	–
7	*Data Manager* (Timeworks)	–
8	*Electronic Checkbook* (Timeworks)	6
9	*PaperClip* (Batteries Included)	8
10	*Oracle* (Batteries Included)	–

Commodore 64 Educational

This Month		Last Month
1	*Facemaker* (Spinnaker)	4
2	*Fraction Fever* (Spinnaker)	–
3	*Up For Grabs* (Spinnaker)	3
4	*Primary Math Tutor* (Comm*Data)	5
5	*Dungeons of the Algebra Dragons* (Timeworks)	–
6	*Kindercomp* (Spinnaker)	1
7	*Touch Typing Tutor* (Taylormade)	2

VIC-20 Entertainment

This Month		Last Month
1	*Gridrunner* (HesWare)	1
2	*Shamus* (HesWare)	3
3	*Choplifter* (Creative)	2
4	*Temple of Apshai* (HesWare)	4
5	*Crush, Crumble and Chomp* (HesWare)	–
6	*Amok* (UMI)	10
7	*Predator* (HesWare)	9
8	*Escape MCP* (Comm*Data)	–
9	*Exterminator* (Nüfekop)	7

VIC-20 Home/Business/Utility

This Month		Last Month
1	*Quick Brown Fox* (Quick Brown Fox)	1
2	*Turtle Graphics* (HesWare)	2
3	*HES Writer* (HesWare)	3
4	*HES Mon* (HesWare)	4
5	*Household Finance* (Creative)	5
6	*Practicalc* (Computer Software Associates)	–

VIC-20 Educational

This Month		Last Month
1	*Touch Typing Tutor* (Taylormade)	1
2	*Primary Math Tutor* (Comm*Data)	–
3	*Type Attack* (Sirius)	2
4	*Gotcha Math Games* (Comm*Data)	5
5	*Hangman/Hangmath* (Creative)	4

Don't let price get in the way of owning a quality printer.

Adding a printer to your computer makes sense. But deciding which printer to add can be tricky. Do you settle for a printer with limited functions and an inexpensive price tag or buy a more versatile printer that costs more than your computer? Neither choice makes sense.

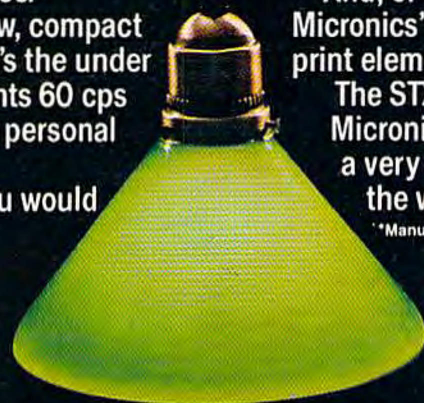
Here's a refreshing option—the new, compact STX-80 printer from Star Micronics. It's the under $200 printer that's whisper-quiet, prints 60 cps and is ready to run with most popular personal computers.

The STX-80 has deluxe features you would expect in higher priced models. It prints a full 80 columns of crisp, attractive characters with true descenders, foreign language characters and special symbols. It offers both finely detailed dot-addressable graphics and block graphics.

And, of course, the STX-80 comes with Star Micronics' 180 day warranty (90 days on the print element).

The STX-80 thermal printer from Star Micronics. It combines high performance with a very low price. So now, there is nothing in the way of owning a quality printer.

*Manufacturer's suggested retail price.



star™ micronics • inc

THE POWER BEHIND THE PRINTED WORD.

Computer Peripherals Division
P.O. Box 612186, Dallas/Ft. Worth Airport, TX 75261 (214) 456-0052

Now with more computer connections.

The new STX-80 printer for only $199*


NOW YOUR
COMPUTER FITS THE
ORIGINAL
ARCADE HITS.
TI 99/4A
IBM PC
COMMODORE 64
COMMODORE VIC 20
DIG DUG
FROM ATARISOFT
STARGATE
FROM ATARISOFT
ROBOTRON: 2084
FROM ATARISOFT
PAC-MA
FROM ATARISOF
CENTIPEDE
FROM ATARISOFT
DONKEY KONG
BY NINTENDO*
OM
TARISOFT

Now the excitement of original arcade graphics and sound effects comes home to your computer.

Introducing ATARISOFT.™ A new source for computer software.

If you own a Commodore VIC 20 or 64, a Texas Instruments 99/4A, an IBM or an Apple II, you can play the original arcade hits.

DONKEY KONG by Nintendo, CENTIPEDE,™ PAC-MAN, DEFENDER, ROBOTRON: 2084, STARGATE and DIG DUG. (On the TI 99/4A you can also play Protector II, Shamus, Picnic Paranoia and Super Storm.)

So, start playing the original hits on your computer.

Only from ATARISOFT.

Some games also available on ColecoVision and Intellivision.

ATARISOFT™

Now your computer fits the arcade hits.

DONKEY KONG, Mario and NINTENDO are trademarks and © Nintendo 1981, 1983. PAC-MAN and characters are trademarks of Bally Midway Mfg. Co. sublicensed to Atari, Inc. by Namco-America, Inc. DEFENDER is a trademark of Williams Electronics, Inc., manufactured under license from Williams Electronics, Inc. ROBOTRON: 2084 is a trademark and © of Williams 1982, manufactured under license from Williams Electronics, Inc. STARGATE is a trademark and © of Williams 1981, manufactured under license from Williams Electronics, Inc. DIG DUG is created and designed by Namco Ltd. manufactured under license by Atari, Inc. Trademarks and © Namco 1982. PROTECTOR II is a trademark of Synapse Software Corporation, manufactured under license by Atari, Inc. SHAMUS is a trademark of Synapse Software Corporation, manufactured under license by Atari, Inc. PICNIC PARANOIA is a trademark of Synapse Software Corporation, manufactured by Atari, Inc. SUPER STORM is engineered and designed by Synapse Software Corporation, manufactured under license by Atari, Inc. ATARISOFT™ products are manufactured by Atari, Inc. for use on the above referenced machines and are not made, licensed or approved by the manufacturers of these machines. COMMODORE 64, VIC 20, TEXAS INSTRUMENTS 99/4A, IBM, APPLE, COLECOVISION and INTELLIVISION are respectively trademarks of Commodore Electronics Limited, Texas Instruments, International Business Machines Corp., Apple Computer, Inc., Coleco Industries, Inc. and Mattel, Inc. A Warner Communications Company.

Complete this coupon and we'll keep you up to date on the newest hits from ATARISOFT.™

Name ____________________

Address ____________________

City __________ State __________ Zip __________

Telephone ____________________

PRODUCT OWNED: (Check one)

1 ☐ TI-99/4A	5 ☐ Commodore Vic 20
2 ☐ IBM PC	6 ☐ Intellivision
3 ☐ Commodore 64	7 ☐ Apple II
4 ☐ ColecoVision	8 ☐ ________

Mail to:
Atari, Inc., P.O. Box 2943,
So. San Francisco, CA 94080. ASM 6

SIMPLE ANSWERS TO COMMON QUESTIONS

TOM R. HALFHILL, EDITOR

QA

Each month, COMPUTE!'s GAZETTE *will tackle some questions commonly asked by new VIC-20/Commodore 64 users and by people shopping for their first home computer.*

Q. ***I have a 1541 disk drive and a friend has one of the older 1540 disk drives. Is there any problem in trading disks back and forth?***

A. Yes, there is a potential compatibility problem. If you're merely *reading* from each other's disks, you should be safe. But *writing* to them could be hazardous to the files stored on the disks.

The reason is that the 1541 disk drive runs slightly slower than the 1540. In our experience, the difference is insignificant when reading disks formatted on one drive or the other. But if you try to write, the speed difference could cause adjacent blocks of data to be overwritten.

The 1541 drive can be accelerated to the 1540's speed by entering this statement:

CLOSE15:OPEN15,8,15,"UI–"

This makes it safe to write to a 1540 disk on the 1541 drive. To restore the 1541's original speed, initialize the disk or enter:

CLOSE15:OPEN15,8,15,"UI+"

(Notice that "UI–" *speeds up* the drive and "UI+" *slows it down.* This syntax might be the opposite of what you'd expect, but it's straight from pages 8-9 of the *VIC-1541 User's Manual.)*

Unfortunately, you can't slow down a 1540 drive the same way, so it's risky to write to 1541 disks on the 1540. If you use both kinds of drives or frequently swap disks with someone who uses a different drive, you should mark all your disks "1541" or "1540" to avoid problems.

Q. ***Some programs in your magazine I cannot get to run, and I've traced the problem to certain lines which are very long. I type the line exactly as printed in the listing, but when I press RETURN and re-LIST the line, only part of it is there—the rest was chopped off somehow. Even the "Automatic Proofreader" doesn't help. Are these lines, indeed, the problem? If so, is there any way to type these lines and get these programs to work? Why didn't you test the programs first to make sure they worked?***

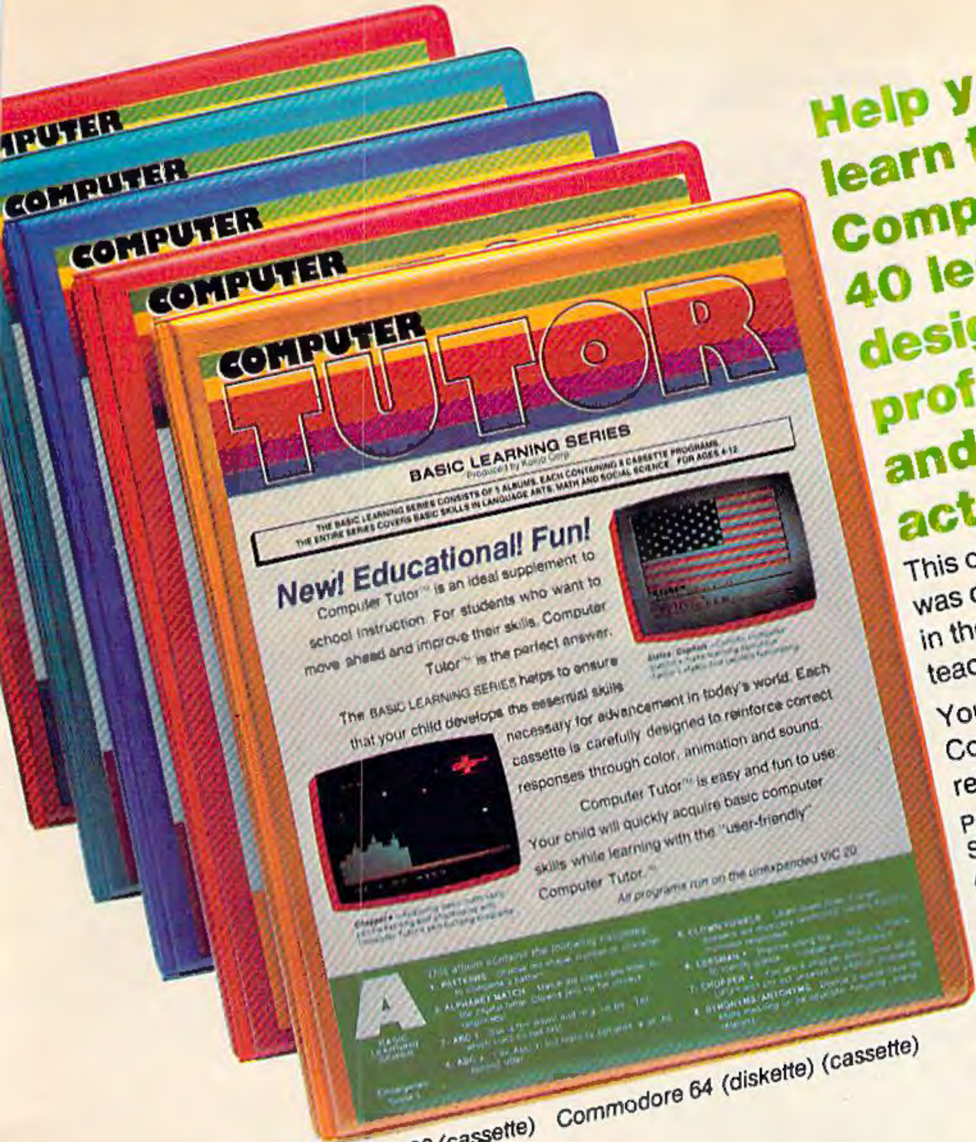
A. You've zeroed in on a problem which seems to have troubled many other readers. Until a number of similar letters and phone calls came in, we had not realized how many readers are unfamiliar with BASIC line-length limits and with the use of keyword abbreviations to solve the problem. Nor had we realized how many programmers routinely use long lines in their programs.

The problem, as you deduced, is that certain lines are too long to type in—at least, too long to type in *normally.* But there is a way to enter them.

Normally, the Commodore 64 does not allow entry of BASIC lines which exceed 80 characters (two screen lines). The VIC-20's limit is 88 characters (four screen lines). If you type in more characters than these limits allow, the extra characters will be discarded when you press RETURN. Unfortunately, the computer does not warn you that the line has been truncated. If you re-LIST the line, you'll see the difference, but most people don't find out until they attempt to RUN the program for the first time. The program either fails to run properly or crashes altogether, often with a cryptic error message as the only clue. The problem can be hard to isolate, especially for nonprogrammers. Frustrated, many people blame a bug in the program or the listing and give up.

But the problem is easy to fix once discovered. The trick is to enter the long line as the programmer did when he or she wrote the program.

In an appendix of the user manual which came with your VIC or 64 is a table of keyword abbreviations (a summary of the most commonly used abbreviations appeared in last month's "Horizons: 64" column). Abbreviations allow you to enter BASIC keywords without typing all the characters. Usually an abbreviation consists of the



VIC 20 (cassette) Commodore 64 (diskette) (cassette)

Help your child learn the basics with Computer Tutor!™ 40 learning programs designed by professional educators and tested in actual classrooms.

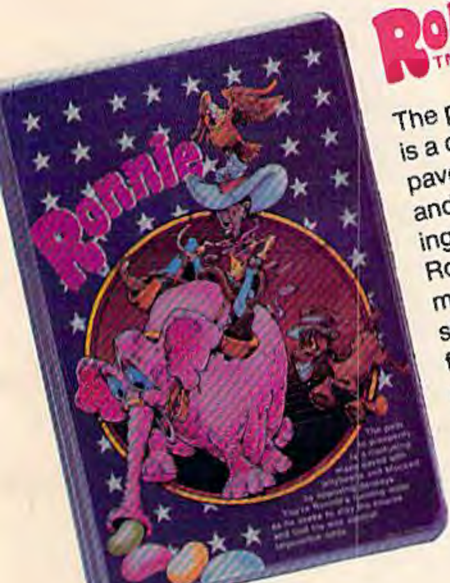
This outstanding set of educational programs was developed by professional educators in the classroom to meet the highest teaching standards.

Your child will delight in learning with Computer Tutor!™ Correct responses are reinforced with colorful graphics and sound.

Programs include Language Arts, Math and Social Sciences.

A—Kindergarten—Grade 3
B—Grade 2—Grade 4
C—Grade 3—Grade 5
D—Grade 4—Grade 6
E—Grade 5—Grade 7

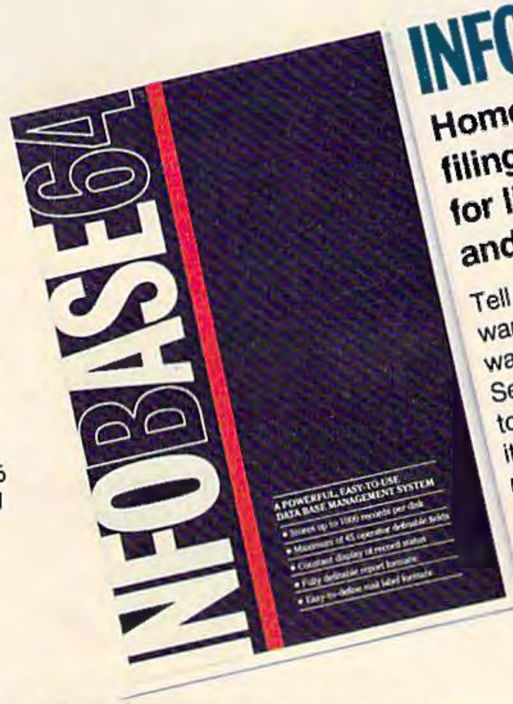
Sugg. Retail per 8 program Album $99.00.
Sugg. Retail per 40 program set of 5 Albums $449.00



Ronnie™

The path to prosperity is a confusing maze paved with jellybeans and blocked by opposing donkeys. You're Ronnie's running mate as he seeks to stay the course and find his way against impossible odds!

SUGG. RETAIL $19.95
Runs on Unexpanded VIC 20 (cassette)



INFOBASE64™

Home/Business computer filing system. Ideal for lists, reports and inventories.

Tell the computer what you want filed and how you want it arranged. See it on the monitor screen or print it. Design the format yourself. Store up to 1000 records on a single diskette.

SUGG. RETAIL $59.95
Runs on Commodore 64 (diskette)

KOLOB

SOFTWARE PRODUCED BY KOLOB MARKETING INC.

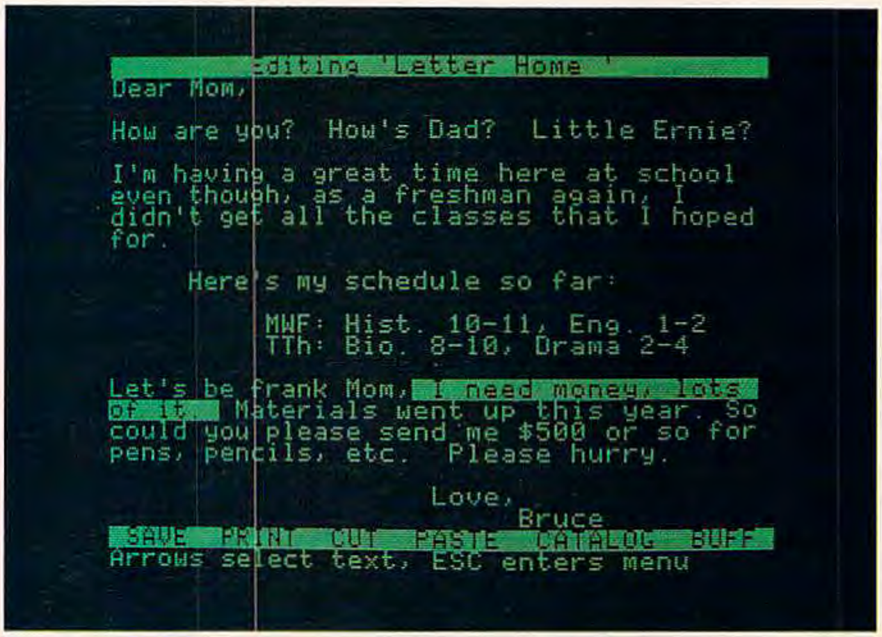
Dealer Inquiries Welcome

Impact Marketing Inc. • P.O. Box 21445 • Salt Lake City, Utah 84121

VISA MasterCard

HOME MANAGEMENT SOFTWARE FROM ELECTRONIC ARTS

If you can learn to use this word processor in 90 seconds, can it really be any good?



CUT & PASTE™ *displays its commands on a single line at the bottom of the screen. This makes working with it easier and also gives you more usable space on the screen.*

Of all word processors on the market today, Cut & Paste may well be the easiest to use. In fact, by the time you finish reading this section of the ad, you'll know how to work with Cut & Paste. So read on.

START TYPING. Working with Cut & Paste is like working with a typewriter. If you know how to use a typewriter, you already know how to type in your draft with Cut & Paste. The only real difference is, with Cut & Paste it's easier to correct typos.

MAKING CHANGES. Let's say you've decided to make a cut in your rough draft. To do this you put the cursor (the bright block) at the start of the text you want to delete, and stretch it through to the end of your cut. Then you send the cursor down to the "CUT" command on the bottom of the screen. Done.

If, on the other hand, you want to keep that line, but put it in a different part of your draft, you use the "PASTE" command. You mark the point of insert with the cursor. Then you put the cursor over "PASTE." That's all there is to it.

PRINTING IT OUT. When you like the way your work looks, you print it. Put the cursor on the "PRINT" command. Then set your margins, in inches. That's it.

You now know how to use Cut & Paste.

OKAY, IT'S SIMPLE. BUT HOW GOOD IS IT? Cut & Paste has all the features you'll ever need to use at home. Here are a few of them:

1. Scrolling dynamic menus
2. Automatic word wrap
3. Simple cut & paste editing
4. Block indenting
5. Set margins and paper size in inches
6. Tabs
7. Automatic page numbering
8. Controllable page breaks
9. Headings
10. Scrolling text windows
11. Automatic widow and orphan control
12. Clear and concise manual

In other words, Cut & Paste will do just about everything other word processors do. But Cut & Paste will do it more easily. Without complex commands and modes.

If you think about a word processor in terms of what it replaces (typewriters, pens and paper, files), Cut & Paste begins to look very good indeed.

And when you consider that *all this power can be had for approximately* $50, we think you'll see why we believe Cut & Paste is something of an achievement.

A PHILOSOPHY OF DESIGN. The people who designed, developed and programmed Cut & Paste have some fairly heavy credentials.

They are people who worked on the internationally-famous user interface designs that led to the Xerox Star® and Apple's Lisa.® They are also

THE CHANGING OF THE GUARD. *Until quite recently we used pens and paper and typewriters to write with, mostly because we knew how to use them. They have been good tools, but limited. You tend to make messes when you work with them, and getting rid of those messes makes extra work. Cut & Paste is an inexpensive and practical alternative. Because it is as easy to use as a typewriter, you really will use it. Which may make it the first sensible word processor for the home. Thus an alleged labor-saving device has come to a position where it really can save a significant amount of labor, i.e., yours.*

THE MEN WHO MADE CUT & PASTE. *The Linotype machine pictured here was the 19th century's most important contribution to word processing technology. It let typesetters compose and rearrange text in the form of metal castings. The importance of Cut & Paste, of course, must await the judgment of history. Nevertheless, the seven men who developed it look confident here. Standing left to right, they are: Norm Lane, Steve Shaw, David Maynard, Dan Silva, Steve Hayes and Jerry Morrison. Seated at the console is Tim Mott, whose idea this was in the first place.*

people who have in common a very lucid philosophy of design.

Computers and the programs they run are tools, they believe. Tools are never noticed unless they are bad tools. When they're good, they become, in effect, invisible. And if you want to make a good tool—an invisible tool—you'd best study the way people use the tools they already have.

As a result of this thinking, Cut & Paste was designed to work much in the same way that you already work with a typewriter or with pen and paper. The most complex and powerful parts of the program are hidden from view. The work they do takes place deep in the machine. All you get to see are the results.

But beyond that, there is something almost indefinable about a good design. Things about it just seem to work crisply. Little touches and features that you notice make you want to smile. If it's really good, it *feels* good.

Cut & Paste feels good.

ELECTRONIC ARTS™

THE PRODUCTS *of Electronic Arts can be found in your favorite computer stores, software centers, and in leading department stores throughout the country. Both Cut & Paste and Financial Cookbook™ are now available at a suggested retail price of $50 for the Apple IIe and the Commodore 64 and will soon be available for the IBM-PC and Atari.*

OUR COMMITMENT TO HOME MANAGEMENT.

Cut & Paste is just one of a growing number of products we're publishing within the category of "home management software." These products are all built around the same program architecture, making them all equally "friendly," as well as remarkably straightforward and practical. We believe that designs like these will soon make home computers as functional and efficient as today's basic appliances.

Our next product in this line is called *Financial Cookbook*. It's a realistic alternative to the complex, pre-programmed financial calculators we all wish we knew how to use. With a few simple keystrokes, *Financial Cookbook* lets you make more than 30 key time-value-of-money computations—just about all the ones you'd ever use for personal finances—like calculating mortgages with changing interest rates, compounding the interest on IRA and savings accounts, and buy-versus-lease comparisons for automobile purchases.

To find out more about these home management products and about what we have planned for the future, call or write: Electronic Arts, 2755 Campus Drive, San Mateo, CA 94403 (415) 571-7171.

first letter of the keyword and a SHIFTed second character. For instance, the abbreviation for POKE is P-SHIFT-O. (Note that the SHIFTed O appears on screen as a graphics character.)

You're still limited to typing 80 or 88 characters when using abbreviations. However, when you LIST a line with abbreviations, the abbreviations expand out to the full keywords, even if the resulting line exceeds the limit. The line appears illegal, but executes normally. Be aware that you cannot edit this line, however; if you want to make a change, you must retype the line from scratch.

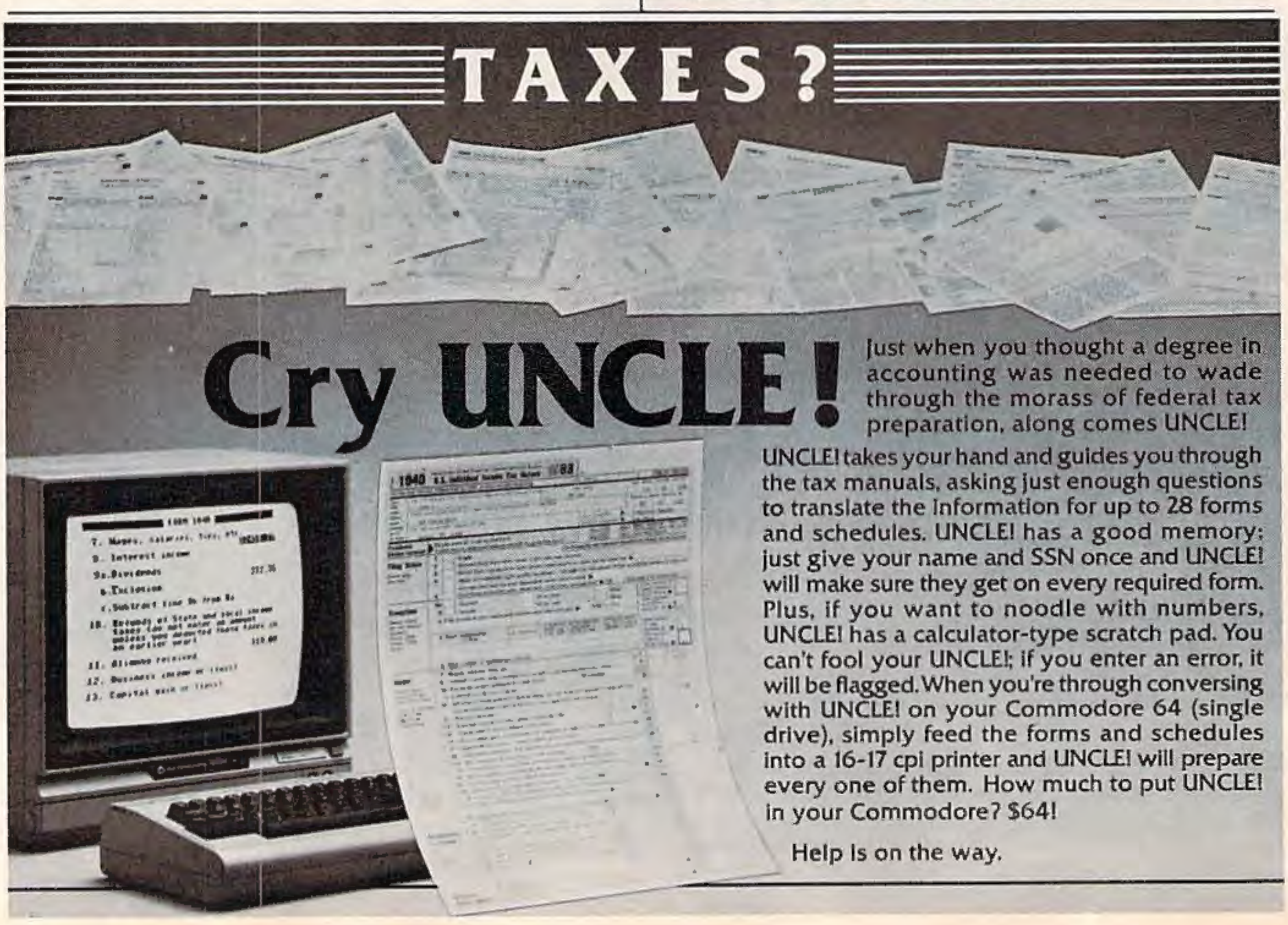
Another problem with abbreviations is that they confuse the "Automatic Proofreader." The checksum program cannot be used to spot typos in long lines.

Programmers use abbreviations to save typing and memory. Not that the abbreviations themselves save memory—BASIC stores all keywords as one-byte tokens, whether abbreviated or not. But abbreviations allow programmers to pack more statements into each line, and reducing the number of lines in a program *does* save a little memory.

Since these long lines execute normally, the programs work fine when we test them prior to publication. We list the program directly from disk and don't hear of a line-length problem until letters begin arriving three months later.

To correct this problem, our lister program now warns us whenever it detects a line exceeding 80 characters. We then try to break up the long line into two shorter ones. Unfortunately, some programs—especially on the unexpanded VIC—require every available byte of memory. Breaking up a line can ruin a working program. In these cases, we'll at least try to warn you of the long lines.

Readers can help, too. If you submit a program to COMPUTE!'s GAZETTE, do not exceed the line limits *unless absolutely necessary to conserve memory*. Also, please do not number BASIC lines by ones, so that our programmers have room to break up long lines if necessary.



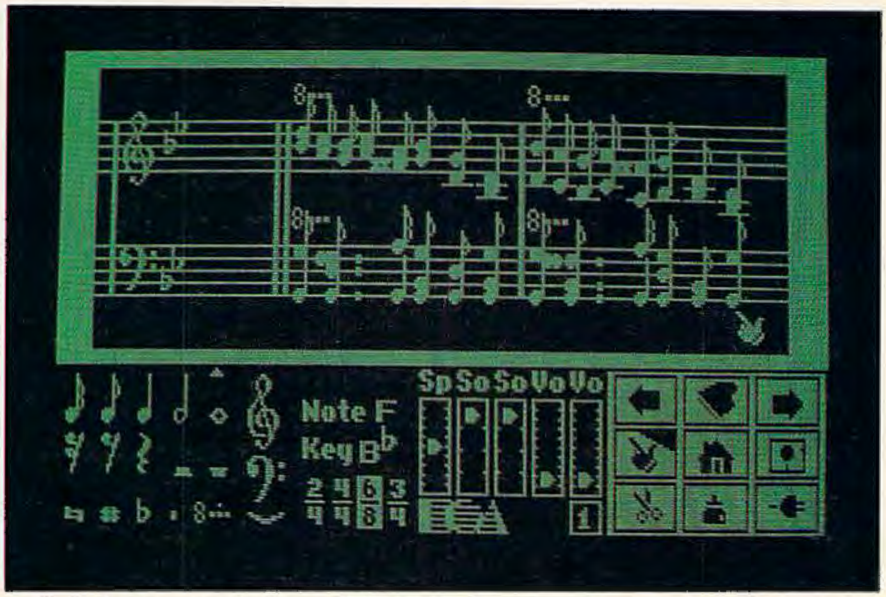
The END *of* DINKETY-DINK-DINK.

Announcing the first computer music program that actually sounds like music.

LET'S FACE IT. Up till now, music programs for your home computer have all sounded, well, pretty lame. There were the ones that resembled little electronic music boxes, remember? And then there were those that sounded like so many burps.

Enter Music Construction Set.™ It's the first music program that really makes use of the power of that machine you've got. If you're a serious student, this means you'll be able to work with an intricacy and range of sound quality you've never heard before on a computer. And if you know nothing about music, you'll find something even more important. Namely, that this thing is simple enough to be a lot of fun.

Take a good look at this screen because it, you, and a joystick are the whole story here.



That's you at the right end of the staff of notes — the little hand. Move the joystick, and you move the hand. Use it to carry notes up to the staff. Lay in rests, signatures, clefs, then point to the little piano in the lower right and listen, because you'll hear the whole thing played back.

Move those little scales in the middle up and down to vary the music's speed, sound quality, and volume. Use the scissors to cut out whole measures, then use the glue pot to paste them in somewhere else. Got a printer? Great. Print the score out and show it off to your friends.

But what if you're not up to writing your own stuff yet? No problem. There are twelve pieces of music already in here, from rock 'n roll to baroque. They're fun to listen to, and even more fun to change. (Apologies to Mozart.)

The point is, the possibilities are endless. But if you're still skeptical, visit your nearest Electronic Arts dealer and do the one thing guaranteed to send you home with a Music Construction Set in tow.

Boot one up. Point to the piano. And listen.

ELECTRONIC ARTS™

MUSIC CONSTRUCTION SET is now available for Apple II, II+, IIe, and Commodore 64 computers. The Apple version, with a Mockingboard,™ plays chords of up to six notes each. The Commodore version plays chords of up to three notes each. Apple is a registered trademark of Apple Computer. Commodore is a registered trademark of Commodore Business Machines, Inc. For more information about Electronic Arts, write us at 2755 Campus Drive, San Mateo, CA 94403 or call (415) 571-7171.

WORD PROCESSING
In The Home

Tom R. Halfhill, Editor

Word processors are displacing typewriters in offices as rapidly as ballpoints replaced fountain pens a few decades ago. But there are good reasons why many of today's households could use a word processor, too.

Word processors are probably the most popular inventions to hit the business world since electric typewriters in the 1960s. Every day, in somebody's office somewhere, a hapless typewriter with its rubber roller platen, pile of typing paper, and bottle of white correction fluid gives way to a gleaming new computer-age word processor. Secretaries are signing up for word processing courses to keep from becoming as obsolete as their traded-in typewriters. The quiet hum of video monitors and the whir of disk drives is replacing the percussion of mechanical striking arms slapping against paper.

It's called the "electronic office" or the "paperless office." At first there was resistance, but by now it's taken for granted that word processing (and computerization in general) is having a significant impact on the function of American business. The business of staying in business and competing for profits is too important for any business person to long ignore a new tool or method for achieving greater productivity.

But in the last five years something even more amazing has happened. This chic new business tool, the computer word processor, has started to find its way into the American home, too. The invention of the inexpensive microcomputer (and its clever packaging as the home computer) has made it possible for millions of people to afford a word processor as readily as most businesses. A \$50–\$100 word processing program running on a \$100–\$400 home computer with a printer can perform most of the major writing functions of a typical \$5000 or even \$10,000 dedicated business system.

However, just as many business users had to be sold on the advantages of word processing, so do many home users. After all, it's much easier to justify the expense of word processors in a business, where dozens or even hundreds of letters, memos, and reports are generated every day. But what good is word processing in the home? How much writing goes on in the average household?

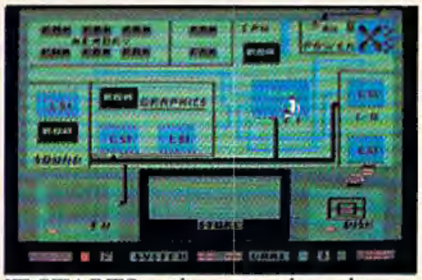
The answer varies, of course, but it can be argued that nearly any home with an adult working in a professional occupation, or with a student of almost any age, can probably benefit from an inexpensive home computer-based word processor. The key is to recognize what a powerful application word processing really is.

First of all, consider the precedence set by other business inventions which have moved into the average household: the calculator and the typewriter.

Mechanical adding machines were used by businesses for decades without making significant inroads into the home. There were several reasons for this: Adding machines were expensive, bulky, and nonportable. Also there was little use for a computing device, however primitive, in the typical home of the early- to mid-20th century. Household finances were generally pretty simple before the proliferation of credit cards, widespread con-

A GAME FOR KIDS. BUT NOT NECESSARILY.

What if you could get small enough to crawl inside your computer and see how all that stuff really works?



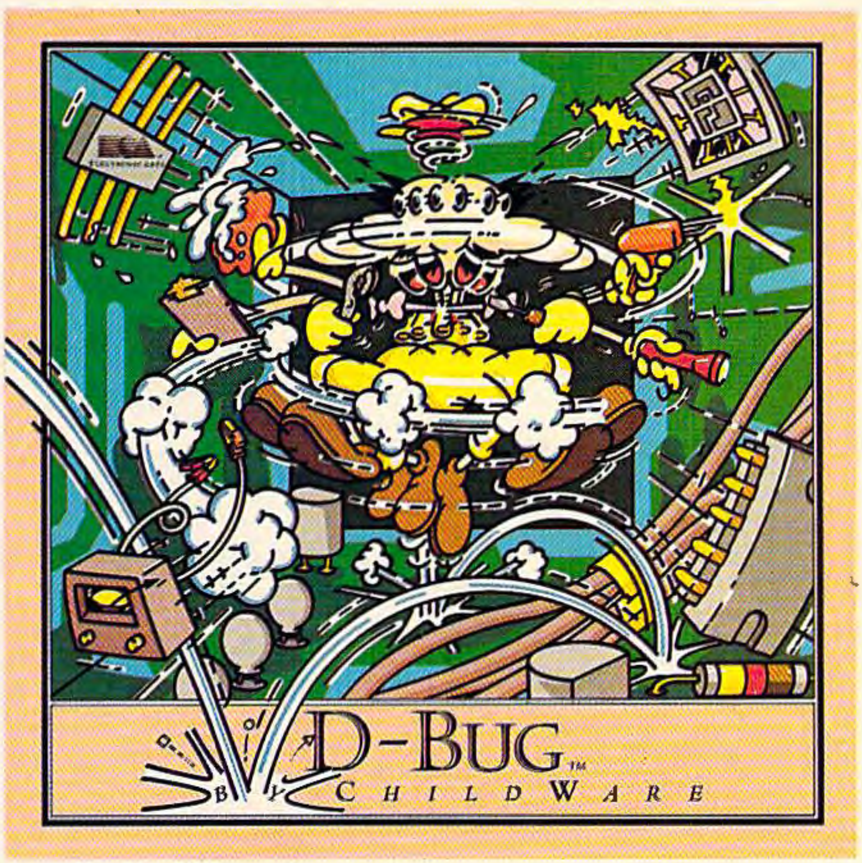
IT STARTS with an arcade-style game. You play it for a while and then something happens. The system goes down in a crash. And now your job is to find out why and make things right again.

Expert help is available in the form of a strange character named Charlie Fixit. He's got a way of making you small enough to get inside the machine. But being inside is yet another game. There are stray charges to duck, static to avoid, and all sorts of intriguing devices to explore before you can get everything back into working order.

The name of this unusual program is D-Bug,™ and it's a wonderful way to introduce your children (and maybe even yourself) to the terminology and basic workings of computers. But beyond this specific knowledge, you'll also learn some fairly subtle skills about how to link causes and effects, and how to develop creative strategies for solving problems.

D-Bug was designed, developed and programmed by ChildWare— pioneers in the field of computer literacy for children. It is just one of an entire line of programs we're publishing that deliberately blur the traditional distinctions between education and entertainment.

D-Bug is now available on diskette for Commodore 64 and Atari home computers and can be found at your favorite computer stores, software centers, and in fine department stores throughout the country.



ELECTRONIC ARTS™

To find out more about our company and our products, write or call 2755 Campus Dr., San Mateo, CA 94403 (415) 571-7171.

sumer loans, checking accounts, modern investment alternatives such as money market funds, and increasingly complicated income tax returns.

Luckily, the electronic calculator arrived at just the right time. Soon after they began replacing adding machines in businesses, they started showing up in homes. A whole new market was created. During the 1970s, calculators grew cheap, small, and powerful. Today almost nobody balances their checkbook or figures a tax return by hand. The very idea is becoming unthinkable.

The spread of typewriters from the office to the home is even more to the point since, like word processors, they are writing machines. Typewriters started appearing in American offices soon after their invention late in the 19th century. At first, their move to the home was held back by many of the same factors which discouraged the home use of adding machines: Early typewriters were expensive, bulky, nonportable, and not all that useful in the average household.

But sometime around World War II things started to change. New typewriters appeared on the market designed for personal use—relatively inexpensive, small, and portable. Colleges began requiring students to hand in typed term papers. Many people started using typewriters for personal business correspondence, and those with office jobs were taking work home. Soon typewriters became a common appliance in millions of households.

For many of the same reasons, word processors are spreading from offices to homes, too. Almost anything a typewriter can do, a word processor can do better. A home computer-based word processor is still more expensive than a cheap portable typewriter, but on the other hand, even the most advanced electronic "memory typewriter" is not as flexible as the typical home computer word processor. College, high school, and even younger students are using word processors for term papers and reports. Their parents can use the same word processor for personal business letters, or for work taken home from the office.

Best of all, the expense of a home word processor is minimal if the household already owns a home computer. Maybe the computer was originally purchased for running education/home application programs, or playing videogames, or for learning how to program. It can be transformed into a powerful word processor with the addition of the appropriate program and maybe a printer and disk drive.

Thinking of a word processor as a substitute for a typewriter is enough to justify its purchase for many people. But there are other reasons which may be even more compelling. A word processor is much more than just an electronic, computerized typewriter—it's a whole new way of writing.

In the beginning, typewriters were used simply to make neat copies of documents composed originally in longhand. Even today many people still use typewriters this way, preferring to write everything out before typing up the final draft.

But soon after their invention, typewriters were embraced by writers, especially journalists. For the first time since the clay tablet and stylus an entirely new writing implement had been invented. Newspapermen were among the first to make the transition from writing by hand to composing their thoughts directly on a keyboard. Pencils and pens—which in various forms had been the only writing instruments since writing itself was conceived—were relegated to notetaking and editing.

Why is this important? Because the transition to the keyboard is an important step in the development of many writers. The majority of professional writers do their writing on a keyboard, not in longhand.

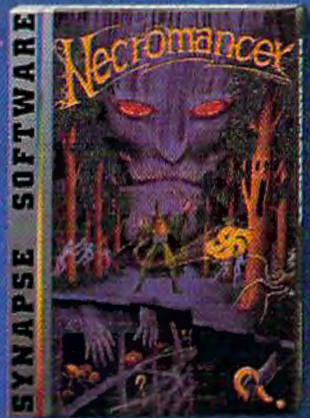
Some writers, including a few famous ones, still contend that writing in longhand with a pen is a more intimate way of committing thoughts to paper. We won't argue this point, because it's one of those philosophical questions that is rarely resolved. However, there is little doubt that writing in longhand is slow. When deadlines are not important, this may not matter. Indeed, many writers who always write in longhand are accustomed to pacing their thoughts accordingly, and argue convincingly that longhand doesn't slow them down.

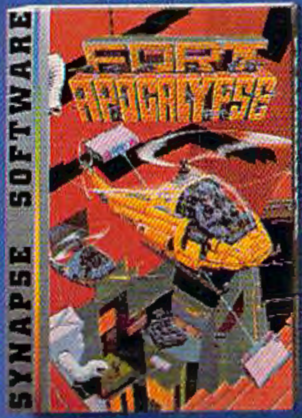
But many writers who switch to typewriters notice something strange and wonderful: apparently because they can put their thoughts to paper so much faster, the words start coming faster. At first the switch from longhand to typewriters is not always easy. It is forced on those writers who must consistently produce on deadline—such as journalists—and the transition can be traumatic. Before long, however, the old pen-and-paper method seems agonizingly slow, and they dread being out of reach of a keyboard.

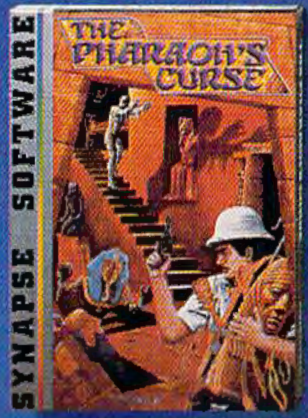
Unfortunately, efficient as they are, typewriters are far from the ultimate writing tools. Once a word is typed, for practical purposes it is committed as indelibly as a word penned in ink. It's possible to make minor corrections with erasable bond, correcting ribbons, chalk strikeover sheets, or white correction fluid. But major revisions mean extensive retyping. Longhand manuscripts aren't very flexible, either. Many drafts may be required before the final acceptable copy is ready.


FOR THE COMMODORE 64®
Pop open a box of
SYNAPSE SOFTWARE
SENTINEL
elite club
On guard!
The terrifying sights and sounds of deep space, as seen from the pilot's seat of a heavily armed star cruiser. Are you ready to meet the deadly Gorganitor? Sentinel.™
© 1983 Synapse Software

Synapse excitement!







Spell-binding

Only you can restore the forest with ancient spells. Then you must march your armies of enchanted trees into battle against the Troglodytes and the evil Necromancer. Who will emerge triumphant from the final conflict? Necromancer.™

Take the controls

Your helicopter mission—capture vital fuel and weapons, free the enslaved masses, and finally destroy the fortress itself. Will you triumph or be crushed by the fiendish Kraalthan lords? Fort Apocalypse.™

Ancient treasures

A fortune is yours for the taking. But can you avoid the ghost of Rama and the evil mummy? Are you nimble enough to leap the chasms and outsmart the deadly booby traps separating you from freedom? The Pharaoh's Curse.™

Join the Elite today!

A whole new dimension in gaming from the leader in home computer excitement! Elite Club membership entitles you to an ID card, certificate, a Synapse hat, patch and quarterly newsletter, all for one low membership fee.

You'll also receive advance information on new Synapse games and be able to buy special T-shirts and posters. Join the Elite today,

or send for more information and receive a FREE Synapse patch!

Special Offer: The first 1000 members also receive an incredible Blue Max™ poster!

synapse

synapse

☐ YES. Send me a FREE patch and more club information.

☐ YES, I want to join. I've enclosed $10 for one membership or $15 for my family. ($12/$17 in Canada). Cash, check, money order or VISA/MasterCard.

Name__________

Computer__________

Address__________

City__________

State______ Zip______

Please allow six to eight weeks for membership processing.

synapse

5221 Central Avenue, Richmond, CA 94804 · (415) 527-7751

That's where word processing comes in. You may have heard or read elsewhere about the advantages of word processing: Documents are typed not on paper, but on the video screen. Characters, words, phrases, sentences, paragraphs, or even large blocks of text can be modified, deleted, inserted, added to, moved, duplicated, and manipulated to your fingers' content. When everything is just right, you can print out as many perfect copies as you want. You can store the document on disk or tape for later use. You can merge documents saved on disk or tape to create a larger document, giving you the power to build anything from a form letter to a novel.

Most people these days are at least partly familiar with the advantages of word processors, even if they haven't actually used one themselves. But there's another bonus that is a bit more subtle—*a word processor can make you a better writer.*

This isn't just another outrageous claim of the "computers-will-save-the-world" ilk. This writer, and many others, is convinced that it's true.

Word processing makes writing so flexible, so fluid, that almost all the inhibitions are banished. Since anything you type can be changed in virtually any way, there is no reason to agonize over every word and phrase. If it doesn't "read" right, just back up and try again. Experimentation is easy. Even radical changes to your text are only a few keystrokes away. No other writing tool offers anything near this level of flexibility.

Let's face it—everything you write that is seen by other people is a reflection of not only your writing skill, but also your intelligence, style, and personality. These things show up between the lines, if not actually within them. If you are writing for publication, or sending a memo to your boss, or compiling a report to be seen by coworkers, or mailing a complaint letter to a company or a congressman, can you afford not to have every sentence as perfect as you can make it? How many times have you let a typo or awkward sentence slip by because it would mean retyping or rewriting an entire page or more?

Even more important, word processing should not be limited to adults. Children should be encouraged to write on a word processor as soon as they can handle the keyboard and the equipment (which these days is a pretty early age, it seems). Many school systems are beginning to realize the educational value of word processing. Word processors are becoming standard equipment in hundreds of schools, even at the elementary level. In fact, one commercial word processor now on the market for home computers (Brøderbund's *Bank Street Writer*) was specifically designed with young people in mind.

For years, standardized college-entry exams have revealed that the writing skills of American students are sadly deteriorating. Perhaps more emphasis on composition and the careful revision of one's own work—assisted by word processing—can help reverse the trend. Early results from classrooms using word processors are encouraging.

It certainly wouldn't hurt if children were started in this direction at home, using their family's home computer. They could be urged to use the computer for writing book reports, letters to grandparents, keeping a personal diary, composing their own stories, collecting jokes they hear, or just fooling around. In short, anything that gives them writing practice, whether they realize it or not.

More than a few adults have improved their writing with a word processor, too—including this writer. When I hear the cliché that computers (particularly home computers) are "a solution in search of a problem," I argue that even if computers were good for nothing else but word processing, it would be enough to justify their existence. Although the world got along fine for years with pencils and typewriters, sometimes a problem doesn't become obvious until a better solution is invented.

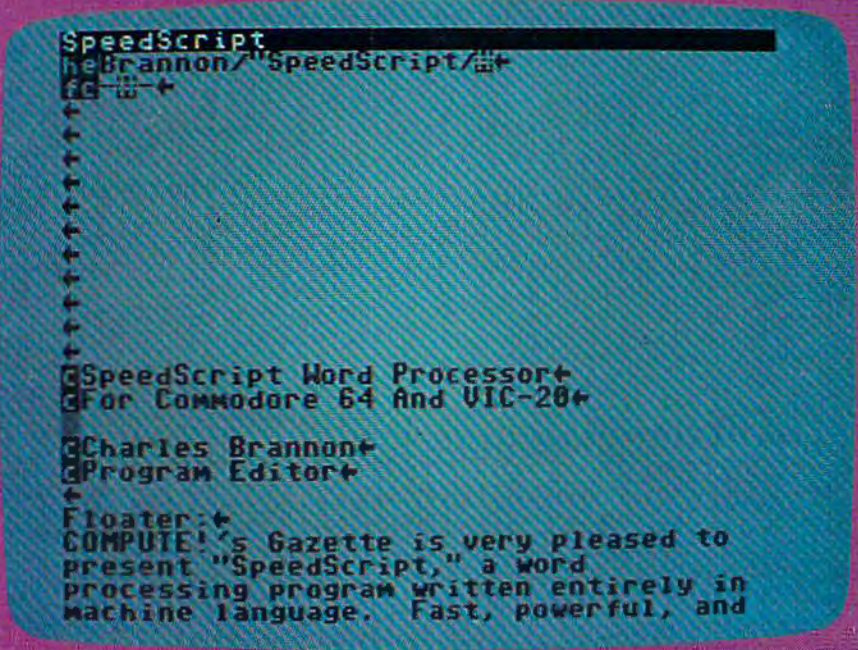
SpeedScript

Word Processor

For Commodore 64 And VIC-20

Charles Brannon, Program Editor

COMPUTE!'s GAZETTE is very pleased to present "SpeedScript," a word processing program written entirely in machine language. Fast, powerful, and easy to use, it includes almost all the major features found in professional word processor programs for personal computers. We believe it approaches commercial-quality programs costing $50 or more. It runs on the Commodore 64 (leaving a huge 45K free for text) and the VIC-20 with 8K or greater memory expansion. SpeedScript will considerably amplify the utility of your computer.



A current advertising campaign extols the virtues of a ballpoint pen that can erase like a pencil, dubbing it the "portable, personal word processor." It can even plot graphics. Like a word processor, the pen can edit, change, and erase. It can produce flawless hard copy. And, indeed, you can draw circles, squares, and bar graphs. But can the pen move paragraphs? Put a 100-page book on a 5¼" disk? Turn a rough draft into final copy with only a few changes? Can it truly edit without a trace of correction, and produce formatted, double-spaced, automatically page-numbered text?

Maybe we're not being fair to the erasable pen, but it should be made clear that word processing is more than just a computerized typewriter. Such a "word processor" would be a few lines long:

```
10 OPEN 1,4
20 INPUT A$
30 PRINT#1,A$
40 GOTO 20
```

When RUN, the program flashes the cursor and waits for a line to be typed. When you hit RETURN, the line is sent to the printer. You can move the cursor left and overstrike or use the DEL key to make changes to the line before you hit RETURN and print it out. But once it's on paper, it's committed. Too late to make any changes.

With a true word processor, you type everything in first, then print the whole thing out. Before you print, you can make as many changes as you want. A good word processor lets you change any line, swap paragraphs, and manipulate your text in numerous other ways. You can buy such a word processing program for your VIC or 64 for $40 to more than $100, depending on the features.

Or you can type in "SpeedScript." Even if you already own a commercial word processor for your VIC or 64, we think you'll be pleasantly surprised. SpeedScript offers all the standard features, plus others you may not have seen before. And there are nearly identical versions for both the 64 and VIC (with 8K or more expansion memory).

Entering SpeedScript

First, you'll need to type in SpeedScript. Programs 1 and 2 look long, but they are only about 4.5K, shorter than most BASIC games. The mass of numbers are machine language. Only with machine language do you get such power, speed, and compactness. Unfortunately, machine language isn't as easy to enter as a BASIC program. To aid with all the typing, we've developed MLX, the machine language editor. Be sure to read and understand the MLX article before you begin typing in SpeedScript.

Type in and SAVE the MLX program. The VIC version will require the 8K expander, both for MLX and SpeedScript. When you are ready to enter SpeedScript, turn your machine off and on (to clear it out), then enter one of these two lines before you load MLX:

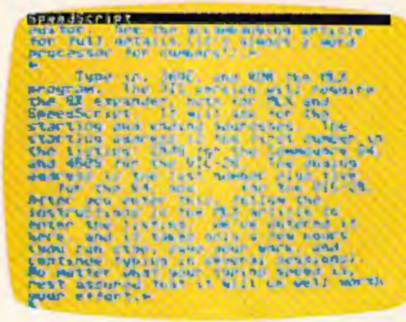
for the VIC:

```
POKE 44,37:POKE 9472,0:NEW
```

for the 64:

```
POKE 44,27:POKE 6912,0:NEW
```

You can then load MLX from tape or disk, and enter RUN. MLX will ask for the starting and ending addresses. The starting address is the first number in the listing: 2049 for the Commodore 64, and 4609 for the VIC-20. The ending address is the last number plus five: 6842 for the 64, and 9342 for the VIC-20. After you enter this, follow the instructions in the MLX article to enter the listing. We've entered it here, and it takes only a few hours (you can stop, save your work, and continue typing in several sessions). No matter what your typing speed is, rest assured that it will be well worth your effort.



Get Creative!

SAVE NEW YORK™
For the Commodore 64.

It was as peaceful a day as New York ever gets, when suddenly the sky went dark and a monstrous droning noise filled the air. Hordes of grotesque aliens were swooping down from all sides, biting into the Big Apple as if they hadn't eaten for days. They were laying eggs, too. Horrible slimy things that got down into the subway tunnels and began clawing their way up. If anyone was going to save the city, it would have to be me. I leapt into my rocket and began blasting away. I thought I stood a fighting chance, but fuel's running low... another wave of invaders on the horizon ...signing off...

S O F T W A R E

Award-Winning Hits for your Commodore

CHOPLIFTER*
For the Commodore VIC-20.

Those are our men they're holding hostage! We don't care how you do it, but you've got to shoot your way in there and bring 'em back alive. You've got three choppers, probably not enough but it's all we can spare. And the enemy camp is pretty heavily fortified. With tanks, jetfighters and truly nasty laser bombs. Okay, maybe it's a suicide mission, but somebody's got to do it. Dozens of innocent lives are at stake. We're counting on you . . . don't let them down!

Now you can play some of America's hottest computer games on your Commodore, and get a FREE introduction to Home Management Software. It's our way of showing you that action-packed gaming is only the beginning of your Commodore's capabilities.

C R E A T I V E

*SELECTED AS SOME OF THE "MOST INNOVATIVE COMPUTER PROGRAMS" 1983 CES SOFTWARE SHOWCASE AWARDS.

with a Free Software Bonus.

SERPENTINE*
For the Commodore VIC-20.

In the Kingdom of Serpents, the only rule is eat or be eaten. Three huge and evil red snakes are slithering through a complex series of mazes, closing in on your good blue serpent from all sides. Move fast and watch your tail! Try to survive long enough to let your eggs hatch into reinforcements. Swallow the magical frogs or your enemy's eggs and you can get the strength to go on... but look out to your left... and ahead of you! They've got you surrounded, and it looks like meal time.

It can teach you. Manage your family finances. Even help you buy a new car. And now, for a limited time only, when you buy one of our specially-marked games you'll receive a certificate good for one of our Home Management Programs absolutely free.

S O F T W A R E

"CHOPLIFTER" AND "SERPENTINE" ARE VIC-20 TRANSLATIONS OF ORIGINALS BY DAN GORLIN AND DAVID SNIDER, RESPECTIVELY. "CHOPLIFTER" AND "SERPENTINE" ARE LICENSED FROM BRODERBUND SOFTWARE, INC.

Getting Started

After you enter SpeedScript with MLX, you can just LOAD it like a BASIC program. As a matter of fact, you can make copies of it with the SAVE command, as usual (SAVE "SPEEDSCRIPT" or SAVE "SPEEDSCRIPT",8 for disk). After you LOAD, enter RUN.

The screen will be light gray or white with black (or dark gray) lettering. The top line of the screen is highlighted.

The blinking cursor shows you where text will appear when you begin typing. You cannot type on the top line of the screen. This is the command window, and is used by SpeedScript to ask questions and display messages. When a message is displayed, it will remain until you begin typing again.

To get started, just begin typing. If a word you're typing won't fit on the screen line, the word and the cursor are moved to the next line. This is called word wrap, or parsing. It makes your text much easier to read on the screen, as words are never split across the margin. Another thing to notice is that a back-arrow appears if you press RETURN. This marks the end of a paragraph or line. It is not necessary to press RETURN at the end of each screen line, as you must do when reaching the end of a line on a typewriter.

Most of us, being human, are not infallible, so you may need to correct your typing mistakes. This is a big advantage of a word processor. You fix your errors before you print, so there's no messy fluids or special ribbons (Did you ever have to manually erase on a typewriter?—ugh!)

If you want to backspace, press the INST/DEL key in the unSHIFTed position. The cursor backs up and erases the last letter you typed. You can press it as many times as necessary to back up to the error, then retype the rest of the sentence. This is clearly not the best way to do things. Instead, you can move the cursor nondestructively. The cursor control keys are in the lower-right corner of the keyboard (see Figure 1: Keyboard Map). The CRSR left/right key moves the cursor to the right, and when SHIFTed moves the cursor left. Before you can correct the error, you have to move the the cursor to the word in question. For example, to correct this line:

```
Now is the rime for
all good men█
```

The cursor is moved to the "r" (cursor-left 21 times):

```
Now is the rime for
all good men
```

The letter "t" is typed:

```
Now is the time for
all good men
```

And the cursor is moved to the end:

```
Now is the time for
all good men█
```

Resume typing:

```
Now is the time for
all good men to
come to the aid of
they're country.
```

Another error! We typed "they're" instead of "their." No problem.

In the above example, of course, you don't have to press the cursor-left key 21 times. You can just hold down the cursor-left key. It will repeat, and keep moving until you let go.

English Cursor Controls

You can also move the cursor in ways that make sense in plain English. For example, if you hold down SHIFT and press the f1 function key, (which is how you get f2), the cursor jumps back to the previous word. To correct the error in the example above, just press f2 five times. You can then press f1 five times to go back to the end of the sentence and resume typing. Here is a list of what the function keys do:

f1: Move cursor to next word.

f2: Move cursor to previous word.

f3: Move cursor to start of next sentence.

f4: Move cursor to start of previous sentence.

f5: Move cursor to start of next paragraph.

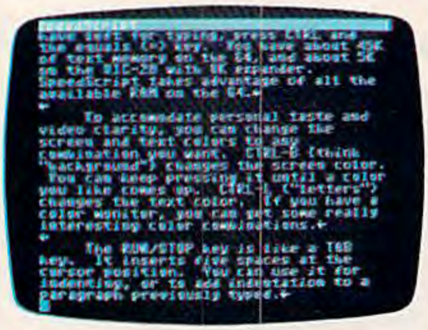
f6: Move cursor to start of previous paragraph.

SpeedScript recognizes a sentence by the ending punctuation (. or ? or !), or by a RETURN mark (back-arrow). A paragraph is any sequence of characters that ends in a RETURN mark (a RETURN mark by itself, which you can use to make blank lines, counts as a paragraph).

Since you're working with English, the cursor up-down keys do not move up or down exactly one screen line. Instead, they act like f3 and f4. Cursor-down moves to the next sentence, and cursor-up moves to the previous sentence. This is easier to understand for many people, but it takes some getting used to for others.

As you begin to move the cursor around, you'll notice that you cannot move the cursor past the end of text. There is an invisible marker, sometimes called End Of File (EOF) at the end of the document. You can add text to the end of your document, but you cannot move past it, since there's nothing there. In a very few cases, you may see some text past the end of file, but you can't move to it, so ignore it.

Many of the other keys behave predictably. The CLR/HOME key in the unSHIFTed position moves the cursor to the top of the screen. If you press it twice, it brings you to the top of your document (in case the document is longer than one screen). The insert key (SHIFT-INST/DEL) inserts a space at the cursor position. You can press it as many times as necessary to make space for inserting a word. You



can also go into insert mode, where every letter you type is automatically inserted. In insert mode, it is not possible to overstrike. You enter or leave insert mode by pressing CTRL-I.

Normally when you type a key, that letter or symbol appears. Certain keys, such as CLR/HOME, however, perform a function. SpeedScript extends this idea and places all the command keys in an easy-to-remember order. For example, insert mode is turned on or off by pressing CTRL-I. (To use a control key, hold down CTRL while you type the other key.)

When you enter insert mode, the command window changes color to remind you. If you press CTRL-I again, you're back in normal overstrike mode, and the command window reverts to its usual color.

CTRL-Z moves you to the bottom of your document (end of file). It's useful for adding text to the end. If you want to check how much memory you have left for typing, press CTRL and the equals (=) key. You have about 45K of text memory on the 64, and about 5K on the VIC-20 with 8K expander. SpeedScript takes advantage of all the available RAM on the 64.

To accommodate personal taste and video clarity, you can change the screen and text colors to any combination you want. CTRL-B (think "background") changes the screen color. You can keep pressing it until a color you like comes up. CTRL-L ("letters") changes the text color. If you have a color monitor, you can get some really interesting combinations.

The RUN/STOP key is like a TAB key. It inserts five spaces at the cursor position. You can use it for indenting, or to add indentation to a paragraph previously typed.

If you want to change the case of a letter or word, position the cursor on the letter and press CTRL-A. It will switch from lower- to uppercase or vice versa. CTRL-A moves the cursor to the right, so you can hold it down to change more than one letter. Another handy command is CTRL-X, or Transpose. It will switch two adjacent letters. My most common typing mistake is to wsitch (switch) two letters while I'm typing fast. With CTRL-X, it's easy to exchange the two letters without overstriking (which is useful in insert mode).

Text Deletion

With a typewriter, if you don't like what you've typed, you can tear the paper out, crumple it up, and dunk it into "file 13." With a word processor, this satisfying act is accomplished with but a few keystrokes.

With the DEL key, you can erase the last letter typed. If you're in the middle of text and press it, you'll notice that the character the cursor is sitting on is pulled on top of the previous character, and the rest of the text follows along. It sounds a little confusing, but it's easy:

```
The quick brown fox
juunmped over█
```

Cursor is moved to error:

```
The quick brown fox
juun█ped over
```

DEL is struck twice, deleting the erroneous characters:

```
The quick brown fox
juu█ped over
The quick brown fox
ju█ped over
```

If you don't want the text to be pulled back, you can press the back-arrow key. It will just backspace and blank out the previous character without pulling the adjacent characters backward. Another way to delete is with CTRL-back-arrow. The cursor does not move, but the following text is "sucked into" the cursor. It is like a tiny black hole.

If you want to strike out a whole word, sentence, or para-

graph, it's time for a more drastic command: CTRL-E. When you press CTRL-E, the command window turns red (to instill fear and awe). You see the message:

```
Erase (S,W,P):
RETURN to exit
```

Each time you press one of the three keys, a sentence, word, or paragraph is pulled toward the cursor and deleted. You can keep pressing S, W, or P until all the text you want to remove is gone. Then press RETURN to exit the Erase function and resume writing. Erase will remove text to the right of the cursor. If you are at the end of a sentence, word, or paragraph, you can use Delete (CTRL-D) to erase backward. CTRL-D displays:

```
Delete (S,W,P)
```

and immediately returns to the normal mode after its work is done. As an analogy, CTRL-Delete is like the DEL key, and CTRL-Erase is like CTRL-backarrow.

What if you pressed one key too many in the Erase command? What if you change your mind? Oh, no! What if you accidentally erased the wrong paragraph? On most word processors, you're out of luck. But with SpeedScript, you can retrieve the crumpled-up piece of paper and "uncrumple" it. Within certain limitations, SpeedScript remembers and stores the text you Erase or Delete. If you change your mind, just press CTRL-R.

Here's how it works. When you Erase text, the text is moved from the main screen into a *failsafe buffer*, a reserved area of memory. The Commodore 64 version of SpeedScript reserves 12K for the failsafe buffer and the VIC-20 version has 1K.

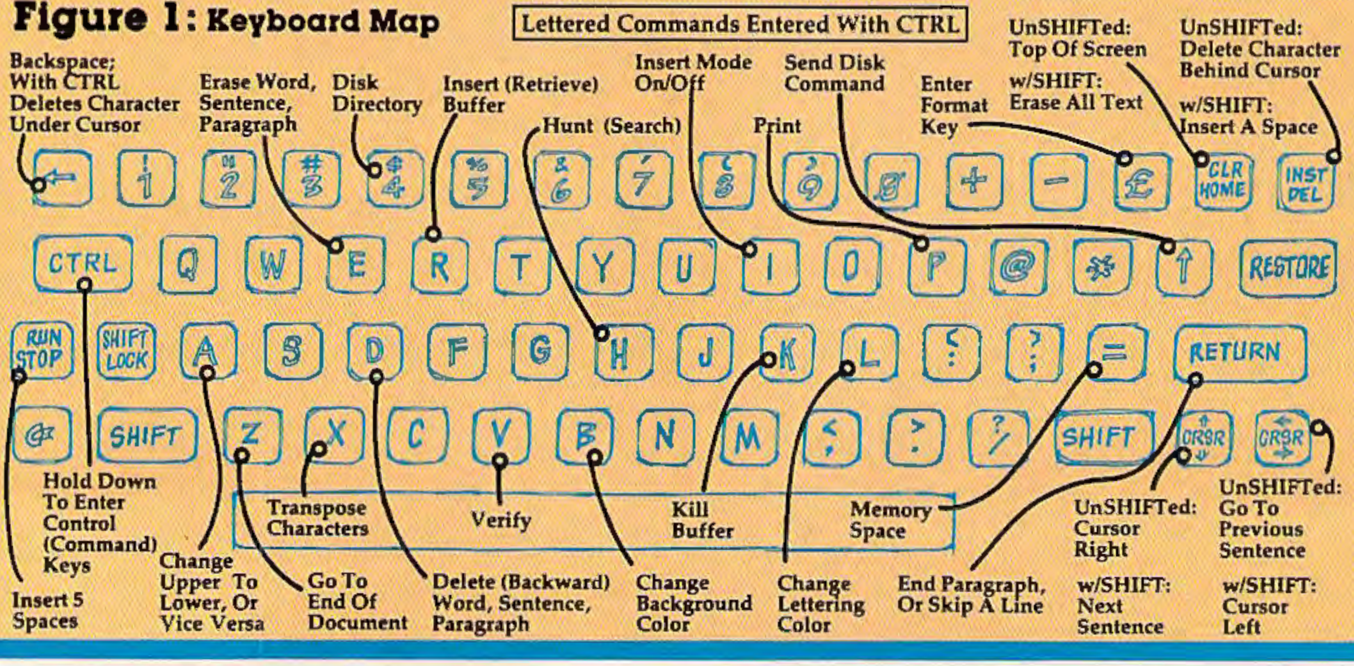
There's another valuable use for the buffer, too. You can move text by putting it in the buffer and recalling it at the destination. Just Erase the paragraphs, words, or sentences you want to move, then place the cursor where you want to insert the text and press CTRL-R (think "Restore," "Retrieve," or "Recall"). In a flash, the text is inserted. If you want to copy (rather than move) a word, sentence, or paragraph, you can restore the deleted text with CTRL-R, then move the cursor and press CTRL-R to insert the deleted text again. You can retrieve the buffer contents as often as you like. For example, if you use a long word or phrase often, just type it once, Erase it, then use CTRL-R to have the computer type it out for you.

You should be aware that CTRL-E and CTRL-D will clear the previous buffer contents. When you move one paragraph, then go back to move another, you don't want to have both paragraphs merged together the second time. Additionally, if CTRL-Delete added text to the buffer instead of replacing the buffer, CTRL-R would insert the text entries out of order, since CTRL-D deletes "backward."

If you want to move two paragraphs at the same time instead of separately, you can override the replacement and cause CTRL-Erase to add to the end of the buffer. Just hold down SHIFT with CTRL as you press E. If you want to force the buffer to be cleared, you can use CTRL-K (Kill) to clear the buffer. If you try to delete more than the length of the buffer (12K on the 64, 1K on the VIC), you'll see "Buffer Full". Stop and move the text, or use CTRL-K to clear the buffer to erase some more.

Finally, if you really want to wipe out all your text, there is a way. (Beware: You cannot re-

Figure 1: Keyboard Map



Tax Pack™

I designed Taxpack so you could do something really practical with your Commodore VIC 20.

Peter Lambert, MBA
Vice-President,
Product Development
Cosmopolitan Software

Taxpack
Powerful income tax computing software specially designed for the Commodore VIC 20.

Now you can use your Commodore VIC 20 to perform all the calculations on your Canadian T1 general tax form. *Taxpack* guides you easily through every aspect of the form with friendly prompts and a comprehensive instruction manual. This new software is available on cassette tape and will run on the standard 3.5k memory in your Commodore VIC 20 home computer.*

Taxpack lets you tackle your income tax form at your own pace. A convenient save-and-restore function lets you record and review historical results. Professional editing features assure easy and accurate data entry. *Taxpack* puts the power of tax modelling and planning for subsequent years in your hands, today.

Many happy returns

Because you can calculate and preview more tax scenarios with *Taxpack* than you'd have the patience or the time to do manually, this software can help you save tax dollars. Custom-tailored to the Canadian T1 general form, *Taxpack* will be updated every year to reflect changes in the government's income tax regulations. Innovative program design allows us to update *Taxpack* within days of the new T1's availability.

So easy to order

To use your *Visa* or *Mastercard*, phone us toll free; or, send your cheque or money order with the handy mail-order form attached. We'll confirm your order by return mail. Your up-to-date *Taxpack* cassette and manual will be shipped within 15 days of the release of the 1983 T1 general form.

To order with Visa or Mastercard call us toll-free:

1-800-268-6364

(from B.C., call 112-800-268-6364)

Satisfaction Guaranteed

We guarantee that you will find *Taxpack* an excellent software value. If you are not totally satisfied, drop us a note to say why, and return the product post paid to us within 10 days for a full refund of the purchase price.

I Want Taxpack!

Please send me ________ Taxpacks @ $29.95	$________
Nova Scotia residents only, add 10% Sales Tax	+$________
Add $2 per unit shipping and handling charges	+$________
Total	$________

My Name ____________________

Address ____________________

City ____________________ Province ____________________

Postal Code ____________________ Telephone ____________________

Make Cheque or Money Order payable to:
Cosmopolitan Software Services Limited
and mail with this order form to:
Box 953, Dartmouth, Nova Scotia B2Y 3Z6 **Attn: Order Desk**

*The cassette also includes an expanded version of Taxpack with enhanced display features, for the Commodore VIC 20's with 8k+ memory expansion.

cover from a total clear.) Press SHIFT-CLR/HOME. You will see:

```
ERASE ALL TEXT: Are
you sure? (Y/N):
```

If you really want to erase all the text, press Y. Any other key, including N, will return you to your text unharmed. You should use this command only when you want to start a new document, as it is one of the few ways to lose text beyond recovery.

Search Feature

When you are lost in the middle of a big document and want to find a particular word or phrase, the Hunt command comes in handy. Press CTRL-H and you'll see:

```
Hunt for:█
```

Enter the word or phrase you want to find, then press RETURN. SpeedScript will locate the word and place the cursor on it, scrolling if necessary. If the phrase is not found, you'll see a "Not Found" message in the command window.

The first time you use Hunt, SpeedScript will search for the phrase from the top of the document. Pressing CTRL-H again will find the next occurrence of the search phrase after the cursor position. You can search for a new phrase without waiting to get "Not Found" for the previous phrase by holding down SHIFT while you press CTRL-H.

There are some tricks to using Hunt. For example, if you search for the word "if," SpeedScript will match it with the embedded "if" in a word like "specific." Should you just want to find the word "if," search for "if" followed by a space. Also, searching for "if" will not match with the capitalized "If."

Saving And Loading

What makes a word processor truly great is that you can save your typing to tape or disk. Say you're writing a term paper. You type it in and save it to disk. Your teacher returns the rough draft with suggested corrections. Without retyping the entire paper, you just load the original, make some changes, and print it out. A 5¼" disk can hold more writing than a briefcase! You can also write in stages: save your work as you go along, then come back to it at another time. Saving and loading alone elevates word processing far above any other means of writing.

To save your work, press f8 (SHIFT-f7). You will see:

```
Save:█
```

Enter the name you want to use for the document. Follow the standard Commodore filename rules, such as keeping the name to 16 characters or less. Press RETURN, then press either T or D, answering the prompt TAPE OR DISK?.

After the Save is completed, you'll see NO ERRORS (hopefully). If there was an error during the save, such as no disk in the drive, or a disk full error, SpeedScript will read the error channel and display the error message. You'll get the error "file exists" if you try to save using a name that's already on the disk. If you want to replace the file, prefix the name with the characters "@:", such as "@:Document". This is called "Save with Replace." You can also press CTRL-↑ (up arrow, explained below) and scratch the file before you save.

Press f7 to load a file. You may want to use SHIFT-CLR/HOME to erase the current text first. The Load feature will append text starting wherever the cursor is positioned. This lets you merge several files from tape or disk into memory. If the cursor is not at the top of the file, the command window will change color to warn you that you are performing an append. You should add text only to the end of the file, as the end-of-file marker is put wherever the load stops. Also, beware that you can crash SpeedScript if you try to load a file and don't have enough room (a file longer than available memory).

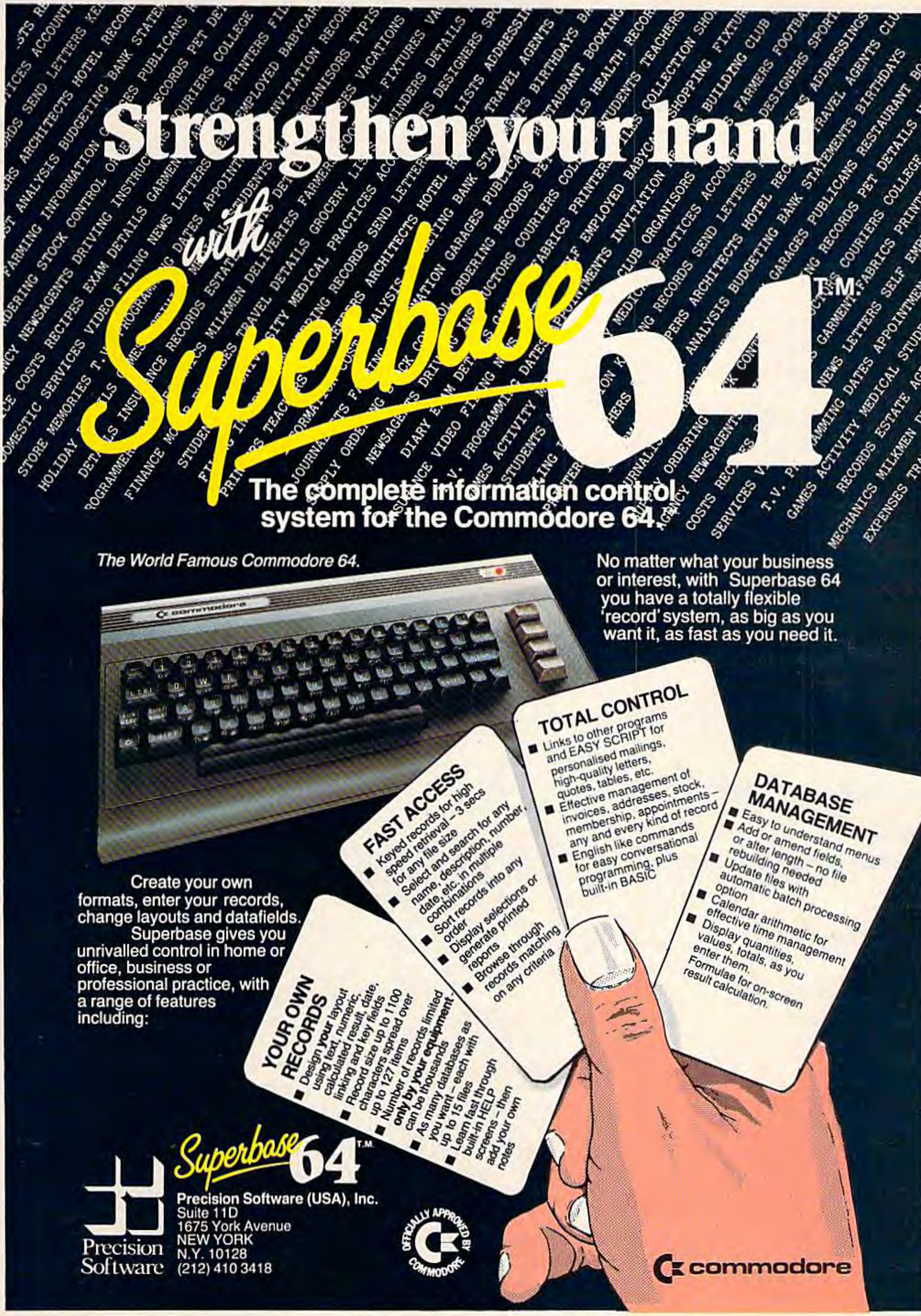
You can use CTRL-V to Verify a saved file. Verify works like Load, but compares the file with what's in memory. It's most useful with tape, but you can use it with disk files, too.

SpeedScript files appear on the directory as PRG, program files. The documents certainly aren't programs, but since the operating system has convenient Save and Load routines, the text files are just dumped from memory. This is also more reliable for tape. You can load files created on some other word processors, such as *WordPro* or *PaperClip*, but you may have to do some reformatting. If the upper- and lowercase come out reversed, you can hold down CTRL-A to transform the entire file.

Other Disk Commands

Use CTRL-4 (think CTRL-$, as in LOAD"$",8 from BASIC) to look at the disk directory. You will not lose whatever text you have in memory. While the directory is being printed on the screen, you can press CTRL to slow down the printing, or the space bar to freeze the listing (press the space bar again to continue).

You can send any other disk command with CTRL-↑ (up-arrow). It may not seem easy to remember, but I think of the arrow as pointing to the disk drive. The command window shows a greater-than sign (>). Type in the disk command and press RETURN. By referring to your disk drive manual, you can do anything the commands permit, such as Initialize, New, Copy, Rename, Scratch, etc. (also see "Getting Started With A Disk Drive," a continuing series in COMPUTE!'s GAZETTE). If you press RETURN without entering a disk command,



Strengthen your hand
with
Superbase 64
T.M.
The complete information control system for the Commodore 64.
The World Famous Commodore 64.
No matter what your business or interest, with Superbase 64 you have a totally flexible 'record' system, as big as you want it, as fast as you need it.
Create your own formats, enter your records, change layouts and datafields. Superbase gives you unrivalled control in home or office, business or professional practice, with a range of features including:
TOTAL CONTROL
■ Links to other programs and EASY SCRIPT for personalised mailings, high-quality letters, quotes, tables, etc.
■ Effective management of invoices, addresses, stock, membership, appointments – any and every kind of record
■ English like commands for easy conversational programming, plus built-in BASIC
FAST ACCESS
■ Keyed records for high speed retrieval – 3 secs for any file size
■ Select and search for any name, description, number, date, etc. in multiple combinations
■ Sort records into any order
■ Display selections or generate printed reports
■ Browse through records matching on any criteria
DATABASE MANAGEMENT
■ Easy to understand menus
■ Add or amend fields, or alter length – no file rebuilding needed
■ Update files with automatic batch processing option
■ Calendar arithmetic for effective time management
■ Display quantities, values, totals, as you enter them. Formulae for on-screen result calculation.
YOUR OWN RECORDS
■ Design your layout using text, numeric, calculated result, date, linking and key fields
■ Record size up to 1100 characters spread over up to 127 items
■ Number of records limited only by your equipment – can be thousands
■ As many databases as you want – each with up to 15 files
■ Learn fast through built-in HELP screens – then add your own notes
Superbase 64 T.M.
Precision Software
Precision Software (USA), Inc.
Suite 11D
1675 York Avenue
NEW YORK
N.Y. 10128
(212) 410 3418
OFFICIALLY APPROVED BY COMMODORE
commodore

Table 1:
Clip-Out Quick Reference Card—Editing Commands

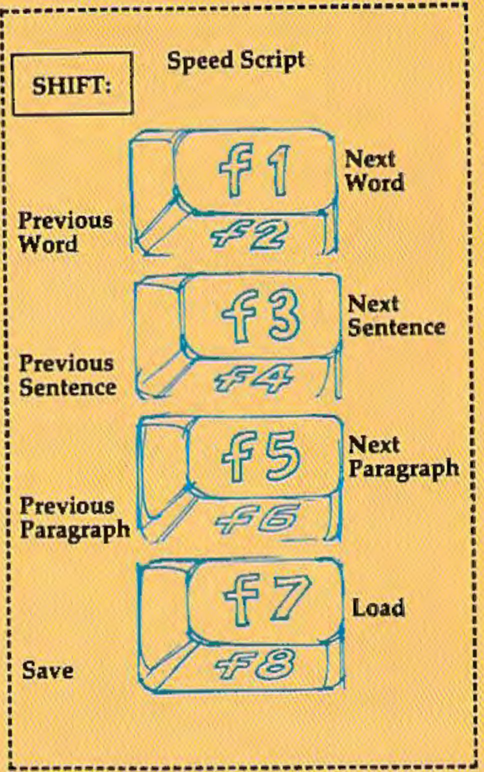
CTRL-A: Change case
CTRL-B: Change background color
CTRL-D: Delete
CTRL-E: Erase
CTRL-H: Hunt
CTRL-I: Insert Mode
CTRL-K: Clear buffer
CTRL-L: Change lettering color
CTRL-P: Print
CTRL-R: Recall buffer
CTRL-V: Verify
CTRL-X: Transpose characters
CTRL-Z: End of document
CTRL-4: Disk directory
CTRL-↑: Send DOS command
CTRL-£: Enter format key
f1: Next word
f2: Previous word
f3: Next sentence
f4: Previous sentence
f5: Next paragraph
f6: Previous paragraph
f7: Load
f8: Save
Cursor Up: Previous sentence
Cursor Down: Next sentence
Cursor Left/Right: As implied
CLR/HOME: Erase All
Back-arrow: Backspace
CTRL-Back-arrow: Delete character
RUN/STOP: Insert 5 spaces

Table 2:
Clip-Out Quick Reference Card—Format Commands

Format commands in column one are entered with CTRL-£.

Cmd	Description	Default
l	left margin	5
r	right margin	75
t	top margin	5
b	bottom margin	58
h	define header	none
f	define footer	none
w	wait for next sheet	no wait
a	true ASCII	
u	underline toggle	
c	center line	
e	edge right	
#	page number	
1–9	see text	

Figure 2:
Clip-Out Function Key Overlay



SpeedScript displays the disk error message (if any). It may be obvious by now that CTRL-↑ is much like the DOS wedge.

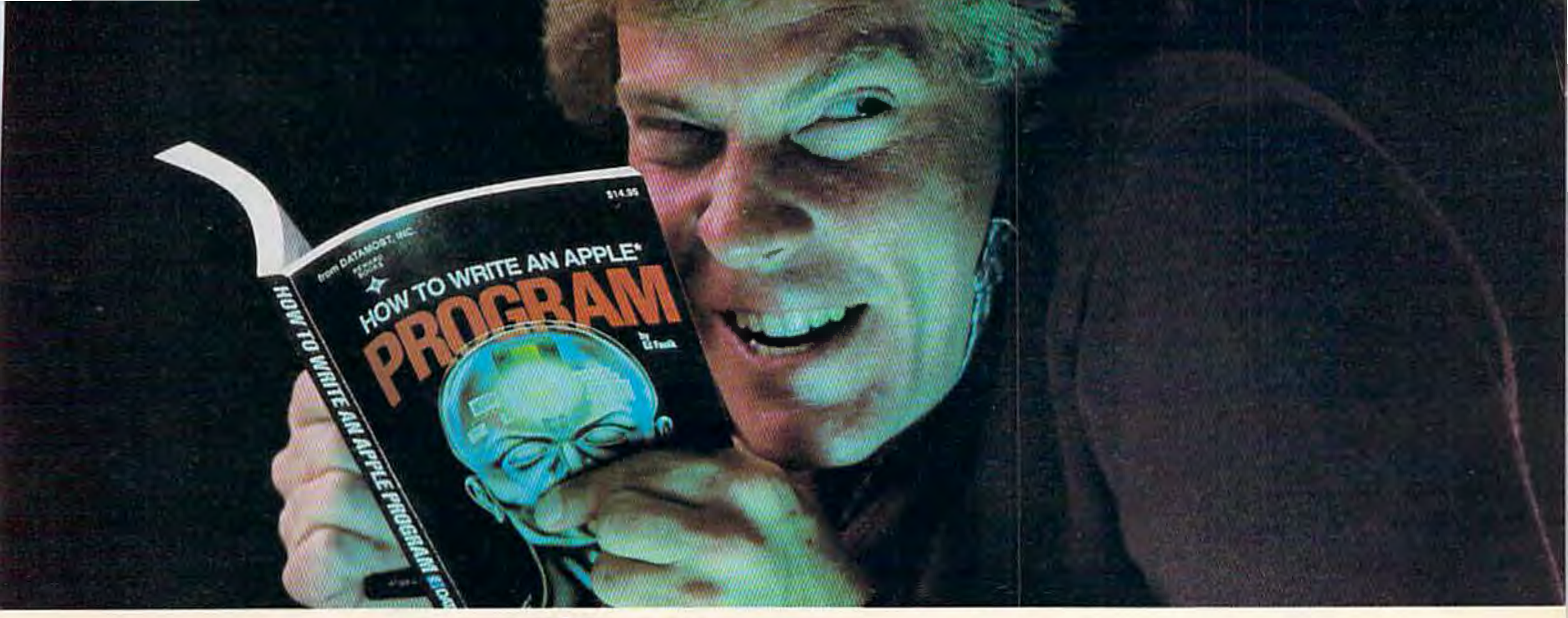
PRINT!

At last, we get to the whole point of word processing—the printout. Actually, you can use SpeedScript without a printer. If you and a friend each have a copy of SpeedScript, you can exchange letters on tape or disk, ready to load and view. You can get a lot of text on one tape or disk. And if you have a friend with a printer and a VIC or 64, you can bring SpeedScript and your files.

Before your text can be printed, it must be formatted. The text must be broken into lines with margins, and there has to be a way to divide the output into pages. For those with pinfeed paper, we also need to skip over the perforation. Of course, it would be nice to be able to automatically number all pages. And why not let the computer center lines for you, or block them edge right? You should be able to change the left and right margin anytime, as well as line spacing. Headers and footers at the top and bottom of each page would add a really nice touch.

Well, SpeedScript does all that and more. But with that power comes the responsibility to learn more commands. These commands do not act directly on the text, but control how the text is printed out. Some commands do things like change the left margin, while others let you do things with the text like centering or underlining. Remember, the formatting commands will not change how the text on the screen looks. They affect only the hardcopy (what's on paper).

Thanks to several default settings, you can print right away without using any printer commands. If you press CTRL-P, SpeedScript will make several assumptions and begin to print. A few of these assumptions are: left margin of five spaces, right margin at 75 (meaning a line



Outsmart your computer.

Show your computer who's boss. Earn its respect. With a Datamost book.

No matter what age or ability level you're at, we have the right book that talks just to you. And your computer. Whether you own an Apple.* An Atari.* Or just about any brand.

All our books are incredibly easy to understand.

Which will make it incredibly easy to understand your computer.

What If You're Scared Of Books About Computers?

Don't be.

Our books are written in friendly, familiar American English. Highlighted with cartoons. And illustrations. So they're fun to read. As well as educational.

And there's over 30 books to choose from. Basic computer learning books to programming books to coloring books.

You won't be bombarded with complicated programming routines until you're ready for complicated programming routines. And no funny technical talk until you've reached the level where you don't think it's funny.

We'll tell you what you need to know. And then, when that's understood, you can take the next step toward outsmarting your computer.

Before You Spend Big Bucks On A Computer, Spend Little Bucks.

Once you've decided, kind of, almost, nearly, what type of computer you think you'd like to buy, buy a Datamost book.

It'll help you understand the Atari or the Commodore* or the Apple of your eye.

Before you spend a lot of money. Before you take your computer home.

You'll make a better purchase decision. Because you'll understand what you're doing. And what you and your computer can do together.

So there won't be any misunderstandings to ruin your new relationship.

How To Get The Most Out Of Your Computer.

Get the most out of our minds.

Datamost books for every computer. Everybody. Every level. Pretty smart, huh?

DATAMOST™ INC
The most out of our minds.™

Datamost, Inc., 8943 Fullbright Ave., Chatsworth, CA 91311, (213) 709-1202
*Atari is a trademark of Atari Computer. *Apple is a trademark of Apple Computer.
*Commodore 64 is a trademark of Commodore Business Machines, Inc.

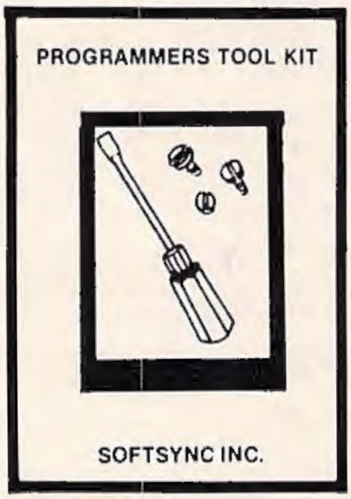
Commodore 64

HOTWARE sale for grownups

LUXURY, FACILITY OR NECESSITY?

COMPUTER MECHANIC • Tune up your car with computer diagnosis • Teaches auto maintainance • Diagnostic guide • Data base for car records • Prevents car failure!
List $\$49^{00}$. **Sale $\$29^{00}$**. (Tape or Disk)

PERSONAL ACCOUNTANT • Easy to use • Double entry accounting system • Home and business • 3 integrated programs • Simultaneously post entries • Compiles financial reports • Large data base • Breaks down expenses • Amortization and payment schedules!
List $\$49^{00}$. **Sale $\$29^{00}$**. (Tape or Disk)



PROGRAMMERS TOOL KIT • Access hi-res graphics • Draws and fills in rectangles, ovals, circles for bar charts, pie charts, and line graphs • Creates single-multi color sprites easily moved • Double width characters • Load and save all memory blocks • Extends and enhances basic • A must for serious programmers!
List $\$49^{00}$. **Sale $\$29^{00}$**. (Tape or Disk)

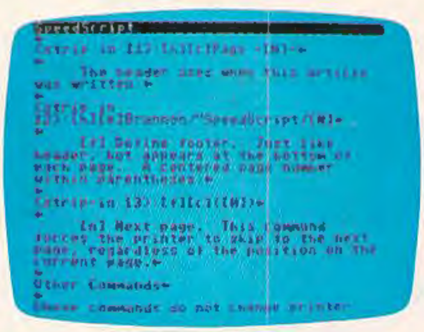
MODEL DIET • Lose weight • Analyze your diet • Based on government research • Tracks calorie intake • Personalizes your diet • Keep fit and healthy!
List $\$49^{00}$. **Sale $\$29^{00}$**. (Tape or Disk)

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

SOFTSYNC SOFTWARE

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order


DR. DATA
Prescribes ABI Software
COMMODORE 64
CASTLES AND CREATURES
DR. DATA's role playing game in which you fantasize yourself as a knight, or otherwise brave person looking for adventure. You journey through the Castle and explore its TOWERS and DUNGEONS, slaying CREATURES as you go.
A top selling Canadian graphics and text adventure game.
DISC
LIST: $29.95 PROTECTO PRICE: $22.95
P. MAN
DR. DATA's prescription: Have your 64, eat every apple to the core, then return to the store to eat the core, but don't get caught by the apple salesmen.
Another fast action, arcade quality, high-resolution game exclusively yours from ABI SOFTWARE.
DISC
LIST: $21.95 PROTECTO PRICE: $17.95
ROBOT RUN
DR. DATA'S PATIENT is being held captive in an installation with a multi-room maze-like security system. You must blast your way through enemy droids, your weapon is a laser, but take heed, for they band together and fight back.
Another fast action, arcade quality, high-resolu
game exclusively yours from ABI SOFTWARE
DISC
LIST: $21.95 PROTECTO
17.95
• THE ASSEMBLER – Assembler language monitor. (VIC 20 & 64).
• INNER SPACE PLANT ATTACK – Save the world's plants.
• RESCUE FROM MARS – Rescue the landing party.
• THE DESIGNER – Sprite and character generator.
• SONIC BOOM – Stop the sound – 20 tunes.
• JET DUEL – Two-player air battle.
• BREAKIN – Arcade game.
PROTECTO ENTERPRIZES
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order



length of 70 characters), and double spacing. If you want to change these settings, you'll need to use the formatting commands.

Entering Format Commands

The format commands are single letters or characters that appear on the screen in reverse video. To get a reverse video letter, press CTRL and the English pound sign (next to the CLR/HOME key). The command window will prompt "Key:". Now press one of the format letters, such as "r" for right margin, or "c" for center. That letter will appear in reverse video (within a "box," with its colors switched). SpeedScript recognizes only lowercase letters and some symbols as commands.

Changing Printer Variables

The printer variables are values such as left margin, right margin, line spacing, top and bottom margins, etc. They are called variables because they can change. For example, to quote a passage within your text, you may indent it by increasing the left margin, and also change to single spacing to set it apart. You would then want to switch back to normal margins and double spacing for the rest of the paper.

To change a printer variable, just follow the reverse video letter with a number. Do not leave a space between a letter and a number. You can put the format commands anywhere in text, though I prefer to group them together on a line of their own. Here is an example setting:

```
[l]10[r]60[s]1[t]10[b]50←
```

To set off these format commands, I'll show here that they are in reverse video by enclosing them in brackets. You'll enter them with CTRL-English pound sign.

[l] Left margin, default 5. The left margin is the number of spaces to indent for each line.

[r] Right margin, default 75. This must be a number less than 80, which is the number of characters that can fit on a line. Add the line length you want to the left margin to get the right margin.

[t] Top margin, default 5. How many blank lines to skip from the top of the page to the first line of printing. Should be at least 5.

[b] Bottom margin, default 58. A number less than 66, which is the number of lines on an 8½" × 11" sheet of paper or pinfeed paper. Do not use a bottom margin more than 58.

[h] Define header. The header is printed at the top of each page, if you specify one. To define the header, begin a line with [h], enter the header text, then press RETURN. Example:

```
[h]Accounting
Procedures←
```

You can embed a format [c] after the [h] to center the header, a format [e] to block the header edge right, and a format [#] any place you want a page number to appear. Examples:

A centered page title with a dash on each side:

```
[h][c]Page -[#]-←
```

The header used when this article was written:

```
[h][e]Brannon/
"SpeedScript/[#]←
```

[f] Define footer. Just like header, but appears at the bottom of each page. A centered page number within parentheses:

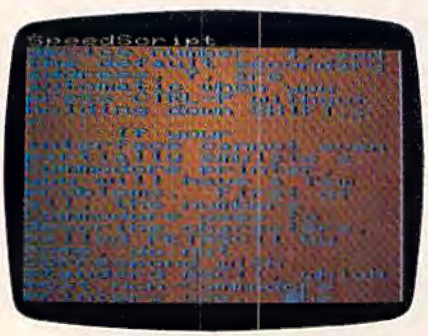
```
[f][c]([#])←
```

[n] Next page. This command forces the printer to skip to the next page, regardless of the position on the current page.

Other Commands

These commands do not change printer variables, so they are usually embedded within a line.

[u] Underline—place on each side of a word or phrase to under-



line. It works by backspacing and overstriking an underline symbol on top of each character. Some printers, including the VIC 1525, do not support the backspace command, so underlining will not work on these printers.

[c] Center—place this at the start of a line you wish to center. Remember to end the line with RETURN.

[e] Edge right—like center, but will block the line to the edge of the right margin.

[#] Page number—When SpeedScript encounters this symbol, it prints the current page number.

User-Definable Codes

Many printers use special so-called escape sequences to control printer functions such as automatic underlining, boldface, italics, super/subscripting, elongated, condensed, etc. These codes are either ASCII numbers less than 32 (control codes) or are triggered by an ESCape character, CHR$(27), followed by a letter or symbol. For example, for the Epson MX-80 with Graftrax, italics is turned on with ESC 4. You should study your manuals to learn how to use these codes. Since most of the control codes and the escape character are not available from the keyboard, SpeedScript lets you define the format commands 1–9.

If you enter [1]=65, then every time the reverse video [1] is encountered during printing, that character (65 is the letter A in ASCII) is sent to the printer. For example, SpeedScript uses the back-arrow for a carriage return mark, so you can't directly cause a back-arrow to print on the printer. Instead, you can look up the ASCII value of the back-arrow, which is 95. You would enter [1]=95, say, at the top of your document. Then, any place you want to print a back-arrow, just embed a [1] in your text. The first four numbers are predefined so that you don't have to set them, but you can change their definition:

[1]=27 (escape), [2]=14 (elongated, most printers), [3]=15 (elongated off), [4]=18 (condensed).

A fascinating possibility is to trigger the bit graphics capability of your printer. For example, you could define special characters. On the VIC 1525, you could send a graphic box (for a checklist perhaps) with:

```
1=82=153=2554=193
134444432 Toothpaste
```

This would appear on the printer as:

```
☐ Toothpaste
```

Printer Compatibility

SpeedScript works best, of course, with a standard Commodore printer. However, we have used it with several other printers such as the Epson MX-80, an Okidata Microline 82A, and the Leading Edge Prowriter (NEC 8023), via an appropriate interface. The interfaces I've used are the Cardco Card/Print and the Tymac Connection. Any interface that works through the Commodore serial port should be fine. SpeedScript will probably not work with an RS-232 printer attached to the modem/user port. SpeedScript may operate with some interfaces which emulate a Centronics port on the user port via software, as long as the software does not conflict with SpeedScript. If you can get your printer to work fine with CTRL-P, skip the next few paragraphs to avoid confusion.

The Commodore printers and most interfaces use a device number of 4. (Other device numbers are 1 for the tape drive and 8 for the disk drive). If you have more than one printer attached with different device numbers, you can enter this number by holding down SHIFT while you press CTRL-P. You'll be asked to enter the device number and the secondary address. Incidentally, you can get a rough idea of page breaks before printing by using a device number of 3, which causes output to go to the screen.

The secondary address is a command number for the printer. For Commodore printers or interfaces which emulate the Commodore printer, the secondary address should be 7, which signifies lowercase mode. The default device number, 4, and the default secondary address, 7, are automatic when you press CTRL-P without holding down SHIFT.

If your interface cannot even partially emulate a Commodore printer, you will have a few problems. First of all, the numbers Commodore uses to describe characters, called PETASCII by some, do not correspond with standard ASCII, which most non-Commodore printers use. The result is usually that upper- and lowercase come out switched. SpeedScript lets you get around this if you place a format [a] at the top of your file.

You also need to use the [a] if you want to bypass the emulation offered by the interface. You may do this to be able to activate your printer's special function codes which are often intercepted and interpreted by the interface. You will also have to use a different secondary address. I'll have to bow out and suggest you scrutinize both your printer's manual and that of the interface.

Pinfeed Versus Single Sheet

The pinfeed or tractor feed is the cheapest and most common paper delivery system for printers. Some printers, however, have a platen like a typewriter and can accept single sheets of paper, such as stationery or company letterhead paper. Normally, SpeedScript prints continuously, skipping over the perforation that divides continuous pinfeed paper.

If you are using single sheets of paper, you need SpeedScript to stop at the end of each page, tell you to insert a new sheet, then continue. If you place a reverse video [w] (for Wait) at the top of your file (again, use CTRL-English pound sign to do this), SpeedScript will do just that. When you get to the end of the page, insert a new sheet, then press RETURN to continue printing.

As you can tell, SpeedScript is a truly comprehensive word processor. I used it to write this article, and it is becoming popular here at COMPUTE! Publications, where writing is a main activity. Although SpeedScript is ultimately easy to use, it may take you a while to master all the features and variations. I hope your adventure will prove to be fascinating and fruitful.

See program listings on page 172.

The Inner World Of Computers

Part 3: How A Computer Remembers

Tom Prendergast

This month's installment examines how computers store information in memory, how you can manipulate that information with PEEK and POKE commands, and how a computer stores a BASIC program.

There have been lots of fan letters (at least two) wanting to learn more about the ELFS (ELectronic FingerS) that work the microswitches inside your computer. Keep those letters coming, folks.

There's one thing we'd like to clear up at the beginning, though. Some people thought it was cruel to shrink programmers down to ELF-size so they could be squeezed into a computer. That's not what we said! We said some genius noticed that the ON/OFF pattern of the front-panel switches on the early mainframes looked like binary and began to program the switches in binary. Then, since hand-setting was no longer necessary, the switches were moved inside. So it was the program and not the programmer that was put inside the computer—there's a big difference.

The earliest computers, full of vacuum tubes and wires, were called "giant brains." They were as big as the side of a barn, but they weren't as brainy as an unexpanded VIC of today because they were four-bit computers, with *nybble*-sized (four-bit-wide) memory cells. You might call them the "four-fathers" (Ouch!) of the VIC and 64, which have eight-bit (*byte*-sized) memory cells.

Four bits limit you to 16 possible binary ON/OFF switch-patterns—0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1111—if you remember our "15-cent computer" of two months ago. You can crowd a heckuva lot more information into an eight-bit byte, because the powers of two *double* the possibilities with every bit you add.

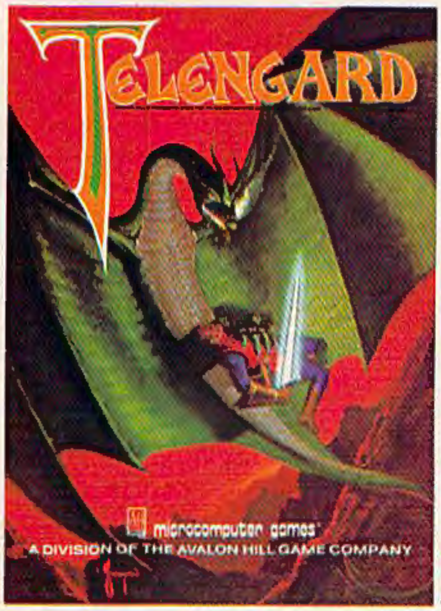
Even so, you can do a lot with four-bit nybbles. Hexadecimal is read in nybbles, and the VIC-20 uses nybble chips in color memory.

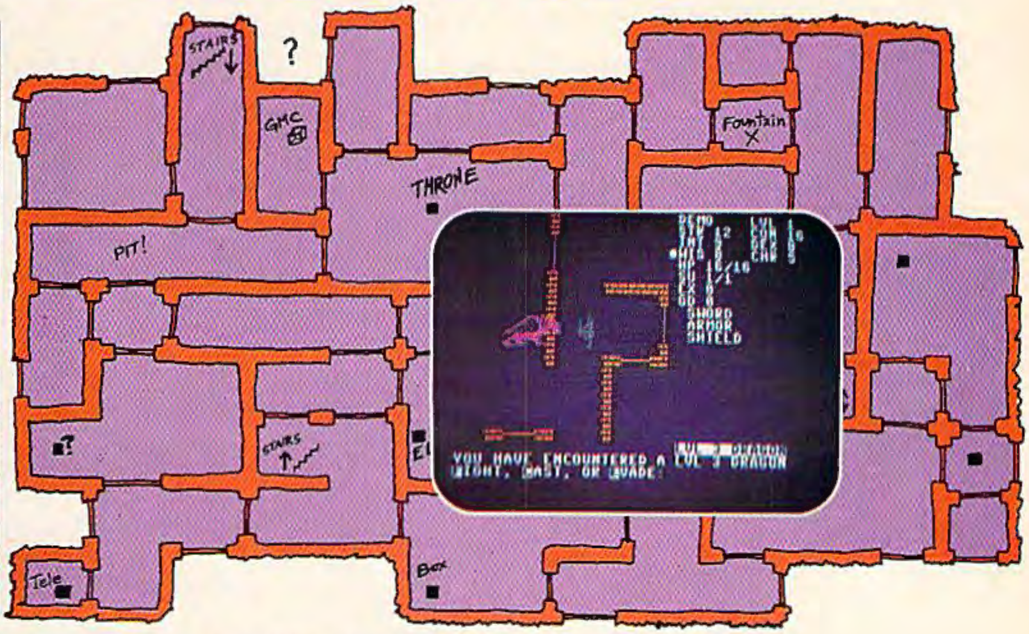
ELF joke: How many ELFS does it take to change a light bulb? Eight if it's a light bulb, but only four if it's a color bulb.

A computer's brain is a lot like ours, although it's a lot smaller, because it's divided into different sections that remember different things. There's a section that remembers what color it was using (color memory), a section that remembers where it put certain things (the *stack*), sections that remember how to do arithmetic and what the letters of the alphabet are.

A PEEK is like reading a computer's mind because it tells you the different kinds of information stored in a memory cell.

Be Amazed!





Telengard: How low can you go?

We've created a subterranean monster. Fifty stories low.

That's the number of levels in the TELENGARD dungeon.

Each labyrinthine level holds hundreds of dark chambers and tomb-like corridors for the mighty adventurer to explore. It goes without saying that a shifting collection of hideous monsters with unpredictable behavior patterns can make life in the TELENGARD maze quite interesting—and frequently quite short!

Using wits, magic and true grit, your character delves deeper and deeper into the depths of TELENGARD in this realtime fantasy role-playing game. Available on cassette for Commodore® 64, Atari® Home Computers (40K), TRS-80® Mods. I/III (32K) and PET® 2001 (32K) for a ghoulish $23.00. 48K diskettes for Apple® II, Atari®, Commodore® 64 and TRS-80® available also, for $28.00.

AND FOR THOSE WHO DON'T DIG UNDERGROUND GAMES . . . There's **B-1 NUCLEAR BOMBER**, a nail-biting solitaire simulation of a manned B-1 on a mission over the Soviet Union. Your plane is equipped with six Phoenix Missiles, a one megaton warhead and orders to retaliate! Cassette for Commodore 64, Atari Home Computers (32K), TI99/4 & 4A (16K), VIC-20 (16K), Timex/Sinclair 1000 (16K), and TRS-80 Mods. I/III (16K) are available for an explosive $16.00. Diskette versions for Apple (48K), TRS-80 (32K), Atari (24K) and IBM (48K) just $21.00.

NUKEWAR: Defend your country by massive espionage efforts, or by building jet fighter bombers, missiles, submarines and ABM's. Your cold and calculating computer will choose its own strategy! Cassette for Commodore 64, VIC-20 (16K), TRS-80 Mods. I/III (16K) and Atari Home Computers just $16.00.

FOOTBALL STRATEGY: Animated action on a scrolling field. A head-to-head challenge or solitaire as you select the best offensive or defensive plays in response to your opponent. Commodore 64, Atari Home Computers (32K) and TRS-80 Models I/III/IV (16K) cassette for $16.00. Atari (32K), IBM (64K) and TRS-80 Models I/III/IV (32K) diskettes available at $21.00.

T.G.I.F.: Thank Goodness It's Friday! Avalon Hill's new party game for one to four players recreating an often-not-so-typical week in the lives of the working class. Half the fun is just making it from Monday to Sunday. Commodore 64, Atari Home Computers (40K) cassette for a meager $20.00. Atari diskette (48K) for $25.00.

AVAILABLE WHEREVER GOOD COMPUTER GAMES ARE SOLD or call Toll-Free: 1 (800) 638-9292 for the name of a store near you. Ask for Operator C.

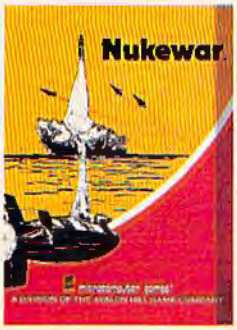
microcomputer games®

a Division of the

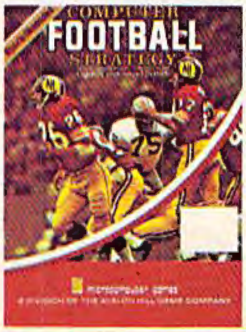
Avalon Hill Game Company

4517 Harford Road, Baltimore, MD 21214 U.S.A. Earth

® Trademarks of Tandy Corp., Apple Computers, Commodore Business Machines, Warner Communications and International Business Machines.



AH
QUALITY



If you count the ASCII and CHR$ codes (see the charts in the back of the VIC and 64 manuals), you'll find there are 256 of them. You have a code for every letter of the alphabet, the decimal numbers from 0 to 9, punctuation marks, graphics—plus codes that call up functions, such as CHR$(147), which clears the screen.

A fully expanded VIC, PET, Apple, Atari, or Commodore 64 has 65536 memory cells, each of which can remember up to 256 different switching-patterns (thought patterns.) Some cells are "hard-wired"—the Read Only Memory (ROM) cells—and can't be changed. But you can change anything in Random Access Memory (RAM) with a POKE.

You can POKE any number up to 255, but when you get to the limit of ON-bits a byte can hold—11111111 (255)—that's it! The next number would be 256 (100000000), and that's 9 bits—too many bits for an eight-bit byte. If you POKE 256 or higher, you'll get an ?ILLEGAL QUANTITY ERROR.

By the way, don't you just love those error messages? What's so illegal about asking for something that isn't there? And the question mark before ILLEGAL is a dead giveaway that they're not quite certain it is illegal. But that's not the ELFS' fault, it's a canned message in ROM memory. When you get a little deeper into machine language, you'll be able to change error messages to anything you want. Like, SORRY, SWEETHEART—TRY AGAIN. You do this by changing the "pointer" (sort of like an ELF bird dog) to point to a different block of memory cells where your new message is stored.

Here's a short program to show you what we're talking about. It POKEs different characters into the RAM cells that are "mapped" to the screen:

```
10 SC=7680:PRINT CHR$(147):POKE 36879,7
20 FOR CELLS=0 TO 505:POKE SC+CELLS,79:NE
   XT
30 PRINT"{15 DOWN}{RVS}{4 SPACES}PRESS SP
   ACEBAR{4 SPACES}"
40 GET SPACEBAR$:IF SP$=""THEN 40
50 POKE SC+CH,CH:PRINT CHR$(19);:PRINT"
   {14 DOWN}{RVS} CODE NUMBER";CH"
60 CH=CH+1:GOTO 40
```

Note: For the Commodore 64, change lines 10 and 20 as below:

```
10 SC=1024:PRINT CHR$(147):POKE 53281,7
20 FOR CELLS=0 TO 999:POKE SC+CELLS,79:NE
   XT
```

When you RUN the program, the screen divides itself into little cells. Now watch the top left HOME cell and press the space bar. @ appears in that memory position, with the CODE NUMBER 0 in reverse below.

Press the space bar again and the letter A appears in the second cell. The code number changes to 1 (its POKE value). Keep pressing the space bar until you've filled up the first 256 screen cells with all of the characters and graphics in character ROM.

Don't press the space bar to POKE beyond code 255, though. You did? You overloaded the byte and got an ?ILLEGAL QUANTITY ERROR IN 50, right? Don't say we didn't warn you!

OK. So we've learned how to POKE things into RAM. Now let's take a PEEK to see how BASIC stored our program in memory. Clear the screen because we're going to see a lot of numbers and you don't want them scrolling off the screen.

Now list line 10 (type LIST 10 and RETURN), and directly below—with no line number—type this:

```
FOR I=0TO31:PRINT PEEK(4096+I);:NEXT
```

For the 64, use PEEK(2048+I)

Make sure you've included the semicolon after the second parenthesis, then type RETURN.

This is what you should see for the VIC (the 64 display will be slightly different):

```
READY.
LIST 10
10 SC=7680:PRINT CHR$(
147):POKE 36879,7
READY.
FOR I=0TO31:PRINT PEEK
(4096+I);:NEXT
 0  32  16  10  0  83
 67  178  55  54  56
48  58  153  32  199
40  49  52  55  41  58
  151  32  51  54  56
 55  57 44 55  0
READY.
```

What do all these numbers mean? Each one represents the byte stored in the 32 memory cells storing line 10. VIC program storage starts at 4096 (2048 is the starting address for the 64), so that accounts for the first zero. That zero is a "null byte"—sort of a place marker—and so is the zero at the very end marking the end of line 10 in memory.

The next two numbers are actually one two-byte number because it's a *pointer* pointing to the memory address where the NEXT program line is stored. (Line 20 has a pointer in front of it pointing to where line 30 is stored, and so on, for every line to the end of the program.) The VIC, 64, Apple, and Atari hitch two bytes together to form an address. This means you can have an address as high as 65535 (1111111111111111 in binary), but figuring out addresses gets really complicated because the bytes are hitched together backwards and the *high byte* follows the *low byte*.

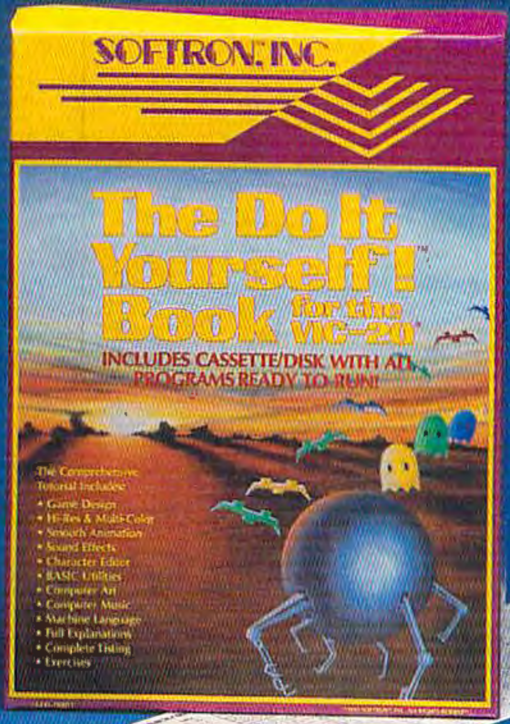
SOFTWARE BREAKTHROUGH
from SOFTRON, INC.

The Do It Yourself! Book™
For the Vic 20®
Comprehensive tutorial with
26 programs ready to load and run!

- Game Design
- Hi-Res & Multi-Color
- Smooth Animation
- Sound Effects
- Character Editor
- BASIC Utilities
- Computer Art
- Computer Music
- Machine Language
- Full Explanations
- Complete Listing
- Exercises

Available soon for the Commodore 64®.
$27.95 Book and Cassette

PIZZA TIME! ™
For the VIC-20®
20 levels of progressive,
fast action maze game!
All machine code, multi-color
arcade action for the VIC-20.
Two players, joystick,
8K expan., $22.95 cassette



Keyboard Soft/Lay ™
For the VIC 20®, Commodore 64®
No more endless paging through
manuals and books! All you need is
immediately at your fingertips!
BASIC commands, memory locations,
memory map, printer codes, music notes,
and more... $8.95

Ask for these Softron™ products at
your favorite computer store,
or directly from us:
150 Nassau St. Suite 2024
New York, N.Y. 10038
212-608-2922
Orders only 800-328-8029
Dealer inquiries invited.

Ez/Disk™
For the VIC 20,® Commodore 64®
Now even you can use a disk drive! No more lengthy
BASIC commands...so easy to use...

Press one key for:
- Disk directory
- Read errors
- Format new disk
- Compress files

And enter only program name for:
- Load, save, verify
- Load, save machine language files
- Back-up files
- Rename
- Delete
- More utilities

Call the menu anytime...
All without losing your BASIC program in memory!
EZ to understand—unnecessary manual included!
[redacted] $29.95

Send Check, M/O plus $3 S/H. $2 extra for disk. (Except Ez/Disk™) N.Y. Residents, add 8¼% tax. Add $3 for COD.

Fortunately, there's a little ELF in there that does all the figuring when a program is running, but if we want to figure it out, we have to multiply the high byte (16 for the VIC, 8 for the 64) by 256 and add the low byte (32) to it. Quick now, what's the starting address for line 20?

Hang in there—we're coming out of the darkness into the light.

The number 10 looks familiar. What do you suppose it represents? It's the "10" of line 10! The zero following the 10 is the high byte of the line number. Like addresses, line numbers are kept in low byte/high byte form. The 83 and 67 are the ASCII coding for S (83) and C (67) of our SCreen variable SC.

Now we're going to throw you a curve. You might expect that the equal sign would be ASCII coded, too, but it's not. In this particular case, the equal sign is an *operator*, and the token code for = is 178. All BASIC operators are tokenized—squeezed into a byte. PRINT, for instance, which has five letters and would need a location for each letter in ASCII, when tokenized to 189 requires only a single cell. This saves a lot of memory space. You don't save any memory by tokenizing a one-character operator like = but CHR$ and POKE are operators and use only one cell.

If you count the number of characters in the listed version of line 10, then count the numbers, you'll find that tokenizing saves you eight bytes: 39 versus 31. There is another reason for tokenizing besides saving memory. The BASIC interpreter, which converts your BASIC programs into machine language (which can be executed by the computer), can only understand instructions in tokenized form. That is, when the interpreter sees the number 153 it knows you want to print, but it does not understand the letters PRINT.

We're going to leave the rest of the numbers up to you to figure out. One trick is to use the operator tokens as landmarks (see the list below for the tokens used in the program), so that any numbers in between must be ASCII. The ASCII code for the digit 0 is 48, for instance; 1 is 49, and so on, in sequence up to the 9, which is 57.... The left parenthesis "(" is 40 in ASCII, the right parenthesis ")" is 41, and a space is 32. You'll find the rest of the ASCII code on page 145 of the VIC manual and on page 136 of the 64 manual.

So that's how an ELF remembers. Some of this may have seemed complicated and roundabout—all the different codes, numbers that aren't numbers, binary, hex—but it's something that's been worked out over the years, and it works!

That's not to say that someone won't think of an easier and quicker way of doing things. A few years from now, we'll be heehawing at today's computers as hard as we heehaw at the big monsters of just a few years ago. You can bet that the computers of the future will be as different from today's machines as rockets from the high-wheeled bike.

Next month we'll take you inside a computer for a guided tour of ELFland. And we'll also show you an easy way to convert decimal to binary that's so simple you can do it in your head. (Who said you need a computer to compute?) Until then, may the ELFs be with you.

SOME BASIC TOKENS

OPERATOR	TOKEN NUMBER
FOR	129
NEXT	130
GOTO	137
IF	139
POKE	151
PRINT	153
GET	161
TO	164
THEN	167
PEEK	194
CHR$	199

Getting Started With A Disk Drive
Part 3: More Disk Commands

Charles Brannon, Program Editor

More on the disk commands, and simplifying them with the DOS wedge.

Last month, we covered many aspects of disk use, from formatting a disk to LOADing, SAVEing, and VERIFYing BASIC programs. I suggest you get that back issue if you haven't seen it yet.

The disk drive, like your computer, has its own microprocessor and memory, which makes it a computer in its own right. This intelligence lets it perform many of the tasks that the computer itself performs on other personal computer systems.

This saves computer memory, since no program is required for essential operations (called the Disk Operating System, DOS). Also, since the disk drive has some independence, it can execute the command you send it, then let the host computer go on to some other task. This is *multitasking*: two microprocessors working together to perform separate tasks simultaneously.

All your VIC or 64 has to do is send an "English-like" command to the drive. As discussed last month, you first have to open up the lines of communication (a channel). This line:

```
OPEN 15,8,15
```

does that trick. The first number can be almost anything. It is just a *code number* that subsequent commands will use to identify this particular channel. The second number, 8, signifies the disk drive. Here is a list of device numbers for Commodore devices:

1 = Cassette
2 = RS-232 (modem)
3 = Screen
4 = Printer
5–7 = Expansion (other printers)
8 = Disk drive
9 = Another optional disk drive

The last number, 15, is the *secondary address*, also known as the command number. In our case, this number tells the disk drive that all input/output through this channel will be communication with the drive's command channel, rather than data to be read or written. We'll cover other uses of the secondary address when we get into reading and writing our own data files.

After we've OPENed our channel, we can send commands in BASIC with PRINT# (say, PRINT-file), or request information from the drive with INPUT# (you got it, INPUT-file). Last month, we tried out the NEW command that formats a disk (prepares it for storage). The form of NEW is:

```
PRINT#15,"N:disk name,ID"
```

Remember that the PRINT#15 will not work

SCRIPT 64™ Makes Your World Letter Perfect

SCRIPT 64 moves you into the exclusive neighborhood of sophisticated word processing for an amazing $99.95.

In addition to all the standard text editing features most higher priced systems offer, SCRIPT 64 also gives you an 80 column format and a built-in spelling dictionary...all in one program for the Commodore 64™.

To make your world letter perfect, ask your dealer for a demonstration of SCRIPT 64 today.

Developed by:

RTC

Richvale Telecommunications, LTD.

Blue Sky Software
Ashland Office Center
Evesham & Alpha Avenues
Voorhees, NJ 08043
609-795-4025

Distributed by:

Warehouse 1, Inc.	Computer Marketing Services, Inc.	DES-Data Equipment Supply
Eastern U.S. 800-253-5330 Western 800-255-0056	800-222-0585	213-923-9361

Also available for the IBM-PC.

unless we've first OPENed file 15. If you don't OPEN a file, yet try to access it, you'll get the obvious:

?FILE NOT OPEN ERROR

Another possible error is:

?DEVICE NOT PRESENT ERROR

You'll usually get this if you don't have the device (disk drive, printer) turned on, attached, or "ready" (some printers have a "local" mode where you control it from its console rather than from the computer).

Here's another command that you'll use a lot. Everyone has files on his disk that he no longer needs. The files may be temporary files, obsolete, or even incorrect. You may also need to remove files from a disk to free up some room on the disk for a new file. The SCRATCH command allows this. Its format is:

```
PRINT#15,"S:filename"
```

Again, the command itself is inside the quotes. The PRINT#15 is just BASIC's way of sending a command. We'll cover another way to send commands with the DOS wedge a little later.

After you send the SCRATCH command, the drive goes to work and BASIC instantly comes back with READY, even though the disk is still spinning. This can be a little misleading. You cannot remove the disk yet—not while the red "busy light" shines and the disk motor is on. But you are free to use your computer for other tasks, such as LISTing a program on the screen.

If you try to send another disk command before the drive has finished SCRATCHing the file, the computer will "hang" while it waits for the drive to finish, then sends the command and returns READY. This process is known as "pipelining."

You can use the asterisk (*) wildcard with SCRATCH, but do be careful. For example,

```
PRINT#15,"S:ENERG*"
```

erases all files on the disk beginning with "ENERG" such as "ENERGY FILE", "ENERGY BASE", "ENERGIZE", etc. It will not remove a file called "SOLAR ENERGY". As you can tell, the asterisk is powerful but dangerous. If you sent:

```
PRINT#15,"S:*"
```

every single file on your disk would be SCRATCHed, quite a catastrophe if done by mistake. I generally do not use the asterisk with SCRATCH, just to be safe. If you're not sure what a file's name is, you can always LIST the directory with LOAD "$",8.

After you SCRATCH a file, it leaves a "hole" behind. If you had three files on a directory:

```
Ø "DEMO DISK          " QZ 2A
2     "TINSELTOES"        PRG
3     "SPACEFACE"         PRG
1     "SMELDGEOIDS"       PRG
658 BLOCKS FREE.
```

and SCRATCHed the middle one:

```
Ø "DEMO DISK          " QZ 2A
2     "TINSELTOES"        PRG
1     "SMELDGEOIDS"       PRG
661 BLOCKS FREE.
```

then there is an invisible gap left between what is now the first and second files. This can be confirmed by writing another file to the disk. Let's say you write a program:

```
10 INPUT "AMOUNT";A
20 PRINT "4% SALES TAX:";A*1.04
```

then SAVE it as "TAXCOMP". The directory would then look like this:

```
Ø "DEMO DISK          " QZ 2A
2     "TINSELTOES"        PRG
1     "TAXCOMP"           PRG
1     "SMELDGEOIDS"       PRG
66Ø BLOCKS FREE.
```

It's sometimes necessary to change the name of a file. Perhaps you've merely changed your mind, don't like the existing name, or want to use an existing filename for another file. The disk drive lets you RENAME a file.

```
PRINT#15,"R:new name=old name"
```

For example, to change the nondescript "GAME1" into "SPACE THIEF", use:

```
PRINT#15,"R:SPACE THIEF=GAME1"
```

This is one of the few commands that readily makes sense.

Another disk command is COPY. It lets you copy a file onto the same disk with a different filename. It can be used in this manner to make convenient backup copies of a file on the same disk. Another use is to move files. If you want to place another program at the top of the disk, for example, COPY the existing program to the disk with a different name, SCRATCH it (which leaves behind a "hole,") then RENAME it. Now you can SAVE the file you want at the top of the disk since it will fill the hole left by the SCRATCHed file.

COPY has a really strange syntax:

```
PRINT#15,"C0:new file=0:other file"
```

I've found you can shorten it to:

```
PRINT#15,"C:new file=other file"
```

In fact, RENAME was also shortened from "R0:NEWNAME=0:OLDNAME". The shorter

form, with the drive number (0) deleted, works just fine:

PRINT#15,"R:NEWNAME=OLDNAME".

COPY cannot copy a file to a different disk or disk drive. It does have another use. You can use COPY to "glue" several files together under a different name. This merge operation is useful for combining two or more data files into one. Again, all the files have to be on the same disk. You can combine up to four files. The syntax here is trickier than ever:

PRINT#15,"C0:newfile=0:file1,0:file2,0:file3,0:file4"

The filename "newfile" (or whatever you call it) will be a merge of file1, file2, file3, and file4. Fortunately, you can shorten this command, too:

PRINT#15,"C:newfile=file1,:file2,:file3,:file4"

The drive number, again, was left out, since there is only one drive in the 1541 (as opposed to the earlier dual-drive 2040 and 4040 for CBM computers). If you only want to chain two files together:

PRINT#15,"C:newfile=file1,:file2"

Notice that the last file in the command need not have ",0" or "," added to the end. Fortunately, few people will ever need to use this variant of the COPY command.

By the way, some of you may be thinking that COPY would be a convenient way of merging two programs, such as a main program and a subroutine. Indeed, Commodore Disk BASIC 4.0 uses COPY for its APPEND command. But since COPY just tacks the files together, it leaves the "end of program" marker between the two files. When you LOAD the combined program and LIST or RUN, the computer sees only the first program, even though the second one is there, using memory. It is possible to remove the end of program marker, but the technique is not brief enough to include here (cheer up, Disk BASIC 4.0 can't do it either).

There are many other disk commands, but most of the rest will be useful only to programmers. We'll cover two of the more arcane ones, though: Validate and Initialize. The form of both commands is simple:

PRINT#15,"V" for validate
PRINT#15,"I" for initialize

What do these do? Initialize causes the disk light to shine, and the disk whirs, spins a bit, then quits. Validate will take quite awhile to finish, then will seemingly have done nothing when you look at the directory. To understand these two commands, we'll have to take a look at the BAM—the Bit Access Map (or Block Availability Map).

There are 683 blocks on one disk. Each block holds 256 bytes, giving you a potential 174,848 bytes of space.

(By the way, a sector size of 256 bytes would seem to indicate double density, since single-density drives use only 128 bytes per sector, so maybe you should buy double-density grade disks. On the other hand, the classification is usually reserved for drives with more than 35 tracks. Try several brands and grades of disks and see which works best for you.)

Somehow, the disk drive has to keep track of which blocks have been used for files, and which are available for future use. Were it not for this housekeeping, a new file could overwrite a previous one. The BAM is stored on the disk as a block of bits, where each bit (on/off, 1 or 0) specifies whether the sector in the corresponding position as the bit is allocated or not (the twelfth bit denotes sector 12). When a file is written, the sectors used are noted in the BAM. In fact, the last line of the directory: xxx BLOCKS FREE, is computed from the BAM.

DOS does not read the BAM every time it needs the information. Usually, DOS reads the BAM once, and stores it in its own memory. It will then update the BAM on the disk when it's done. If you change disks, however, the drive may get confused. It may try to write new files with the old BAM, then write the old BAM to the new disk. Scramble city! The old BAM may say that certain sectors are available, but they might not be on the new disk.

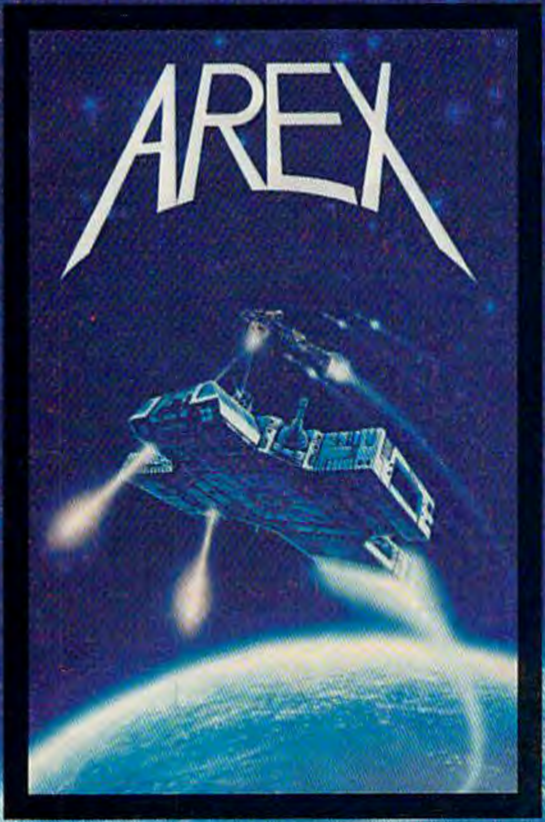
Fortunately, DOS checks the disk's ID before it tries to write a file, or change the BAM. The drive is helpless, however, if you have NEWed (formatted) both disks with the same ID. This is why it is vital that every disk have a unique ID number.

You can prevent this possible catastrophe with Initialize. Initialize forces the drive to read the BAM from the diskette. It also resets some other minor DOS variables. Some people revere Initialize with religious fanaticism, refusing to write to a disk without the ritual of OPEN 1,8,15,"I" (yes, you can send a command via the filename in OPEN).

In practice, it can't hurt. In fact, sometimes the disk head will find itself in an intermediate position between tracks, usually when jostled. The disk can't figure out where it is, since it can't read its signposts which were put on the disk during formatting. You can set the disk straight with an Initialize, which tells it to "go home." (Home is track 18, where the BAM and directory are stored.)

Validate is more useful. It reconstructs the entire BAM by tracing each file on the directory, noting which sectors are used. After it's traced

BECOME AN INTREPID SPACE ADVENTURER . . .



by William Muk
CoCo version by Roger Schrag
Apple version by Gordon Eastman
Atari and Commodore 64 version by John Anderson

Far beyond the known galaxies, you venture deep into the vast reaches of outer space. But you are not alone! In a flash, without so much as a how-do-ya-do, they're in hot pursuit and you're left to do before you're done unto. **Can** you elude your pursuers? **Will** you elude your pursuers? And who ***are*** these guys anyway? Find the answers to these and other compelling questions in AREX. See your dealer today!

AREX . . . Coin-op arcade realism at home for 1 to 2 players.

AREX		
APPLE 48K DISK	042-0172	$34.95
ATARI 16K TAPE	160-0172	$34.95
ATARI 16K DISK	162-0172	$34.95
CoCo 16K TAPE	060-0172	$34.95
COMMODORE 64	160-0172	$34.95
COMMODORE 64	162-0172	$34.95
IBM 64K DISK	132-0172	$34.95
TRS-80 16K TAPE Model 1 & 3	010-0172	$34.95
TRS-80 32K DISK Model 1 & 3	012-0172	$34.95

. . . OR FLY HIGH IN THE WORLD OF HIGH FINANCE

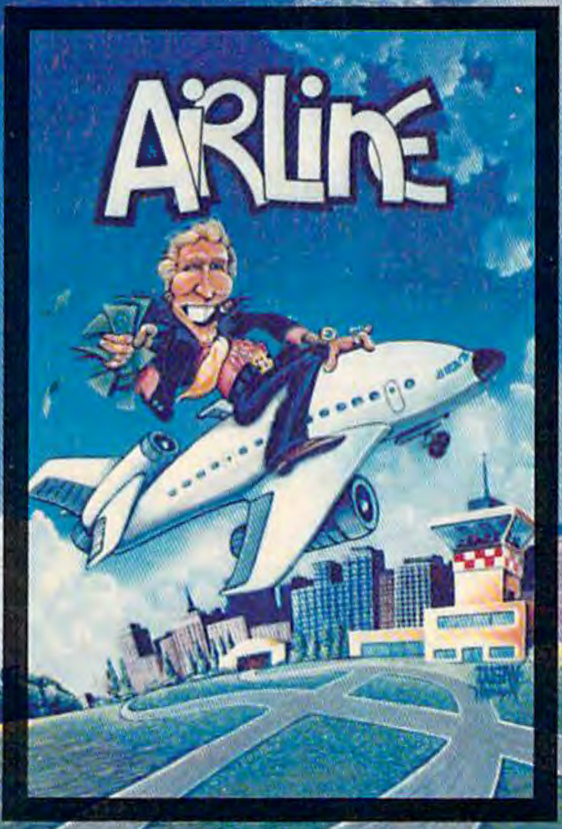
by George Schwenk
TRS-80 version by Dave Simmons
CoCo version by Roger Schrag

"Yas, after purchasing diamond mines in South Africa, oil wells in Saudi, and rare beer cans in Walla Walla, Washington, I had begun to wonder what other trendy commodities remained to be added to my swelling portfolio. Then a snip of a ticket girl dared to tell me (ME, Hartley J. Wormsflather III!) that my flight was overbooked. To avoid future misunderstandings, I bought the airline."

"I think I'm on to something profitable here." Hartley J. Wormsflather III

AIRLINE . . . A no-holds-barred strategy game for 1 to 4 players.

AIRLINE		
ATARI 400 & 800 / CoCo / Model 1 & 3 16K TAPE	140-0169	$24.95



Published by Adventure INTERNATIONAL

a subsidiary of Scott Adams, Inc.
BOX 3435 • LONGWOOD, FL 32750 • (305) 862-6917
Prices Subject To Change Without Notice

To order, see your local dealer. If he does not have the program, then call 1-800-327-7172 (orders only please) or write for our free catalog. DEALER INQUIRIES ARE INVITED!

through all the files, it can then rewrite the newly created BAM to the disk. This can sometimes give you more blocks free on the directory.

Some error or discrepancy in DOS occasionally causes it to misallocate sectors. It may fail to free up blocks, or, rarely, fail to allocate them. These accumulated bit errors can add up to a lot of wasted disk space over time, since the disk won't write to an allocated sector. Validate finds out the truth, so you can sometimes free up disk space unexpectedly.

It's worrisome when this happens, though, since it proves that DOS has made minor errors. If DOS failed to allocate a sector, then that sector could be used by another file, destroying the original file.

Another bug seems to be related to BAM. When you load one program yet get another, the pointers on the disk which identify the starting sector of each file have become switched or garbled. This problem is also accompanied by sectors of the original file which have become overwritten, so there is no way to recover. This problem happens most often with frequently used disks or those which are full. It can also happen when you forget to use the "0:" prefix when SAVEing to disk (SAVE "0:program",8).

Validate can sometimes clear up these confused disks. It can also aggravate it, since the directory from which Validate computes the BAM may be incorrect itself.

You may be tired of always having to OPEN 15,8,15 to send a command. And no one likes having to SAVE your program before you LOAD "$",8 to LIST the directory. Well, Commodore hears you. It has thoughtfully provided a convenient shortcut for using the disk drive from BASIC. Just insert your TEST/DEMO disk and LOAD "C-64 WEDGE",8 or LOAD "VIC-20 WEDGE",8 and RUN. The program will then LOAD the actual wedge program (which is in machine language) and execute it.

The wedge adds a few single-key commands to BASIC. You can use these commands only in the immediate (READY) mode, not in a program. First, let's display the directory. Enter:

```
@$
```

Magically, the directory scrolls by on the screen. You can hold down CTRL to slow it down, or press SPACE to freeze it. Press SPACE again to continue. And when it's finished, you still have your program in memory. Most useful.

You can also send any of the disk commands we've mentioned. Just replace the PRINT#15, with @. For example:

```
PRINT#15,"R:newname=oldname"
```

would be:

```
@R:newname=oldname
```

Remember the small one-line program from last month that will read the error message if the red light is blinking?

```
10 OPEN 15,8,15:INPUT#15,EN,EM$:PRINT
   EN;EM$:CLOSE15:END
```

Quite a lot just to read the error message. The wedge makes this trivial. Just enter the @ and hit RETURN, without sending a command. If there is no error, you'll see:

```
00, OK,00,00
```

Otherwise, you'll see something such as:

```
63, FILE EXISTS,00,00
```

For a complete list and description of DOS error messages, see Appendix B in your disk drive manual.

With the wedge, you should never have to remember to add ",8" to the end of a LOAD or SAVE. Instead, two single-key commands, / (divide-by, on ? key) and the back-arrow (upper-left corner), give you single-key LOAD and SAVE. To LOAD a program, enter:

```
/program
```

If you would like to LOAD and RUN in one step, use the up-arrow:

```
↑program
```

SAVEing is easy with:

```
←0:program
```

If the file exists on the disk already, you may want to SCRATCH it first, or use @0: in place of 0: (called Save with Replace).

There's another convenience, too: You don't have to enter the filename. Just list the directory with @$, then stop it (RUN/STOP) when you see the name you want. Move the cursor up to the directory and just type the / or ↑ in the first column, and hit RETURN. The wedge will ignore the quotes, spaces, and extraneous "PRG" business, and go to work.

One more wedge command: the % replaces LOAD "name",8,1. This is known as a *nonrelocatable* load. You would use the % key to LOAD machine language programs:

```
%UNNEW
```

It has an advantage over using BASIC's LOAD command. It will not change the end-of-variables pointer. What this means is that you won't get an ?OUT OF MEMORY ERROR after you use it. You can therefore use it to LOAD machine language without disturbing a BASIC program in memory.

Since the DOS wedge "wedges" into BASIC,

it can make some programs RUN more slowly. If you want the wedge out of your way, just enter @Q (for Quit).

It's not easy to make a copy of the 64 version of the wedge, since it is in machine language. You should first SAVE the boot program "C-64 WEDGE" found on the demo disk, then type in and RUN the program accompanying this article, "Wedgemaker." It will SAVE the wedge from memory, so be sure you've already LOADed in the wedge from your demo/utility disk.

We've covered just about all the essential information this month. Remember that you can use many of these commands from applications such as word processors, too. Next month, we'll further our exploration by reading and writing our own data files. Until then, try out all the commands with a scratch disk until you get the hang of them.

Wedgemaker

```
10 REM 64 WEDGEMAKER                  :rem 139
20 OPEN1,8,1,"0:DOS 5.1"              :rem 218
30 POKE780,253:POKE253,0:POKE254,192:POKE
   781,90:POKE782,207:SYS65496       :rem 214
40 CLOSE1:END                         :rem 28
```

SHOOT THE WORKS with an ARROW™

Have you been spending too much time with your VIC? If it's because you are always waiting for your programs to load, Skyles Electric Works can help.

Enter the Arrow.

This cartridge can load programs off cassette at about eight times the normal speed. It can re-record a program *once* at up to eight times the density on tape. This means that you can get 200 kbytes of programs on a single C-20 tape. In the long run, you'll be able to put 10 tapes into one.

The Arrow cartridge will pay for itself in the tape and the vast amount of leading time it saves.

And it's easy to use.

ARROW ...... adds 4 commands: ←S, ←L, ←V, and ←A that are used to SAVE, LOAD, VERIFY AND APPEND at a *high speed* (3600 baud).

ARROW ...... leaves the usual messages that appear on the screen.

ARROW ...... adds a message telling you the length of the program that has been loaded (or saved).

ARROW ...... keeps all of the usual shorthand fea'ures of program naming.

ARROW ...... starts automatically when you turn on your VIC-20 or CBM-64.

It gives the length of the program that it has loaded or saved. It can handle hexadecimal/decimal calculations: it even has a machine language monitor in it. Bull's-eye!

And advanced programmers can get an optional machine language assembler in the same cartridge. Even though it speeds up program loading it does not effect the tape recorder. It's an *extremely* reliable device.

If you want lightning fast cassette operations, shoot for the Arrow.

But this great device does more than cut down on loading time and tape space. It is a fine programmer's aid, too. With some sharp but simple program text editing commands, the Arrow might be the best bargain around in personal software for your VIC. The Arrow is always on target with auto line numbering and line deleting features.

ARROW ...... adds the following programming aids:

AUTOLINE numbering
RENUMBERING of lines
DELETE a line or group of lines
SEARCH with optional replace
MOVE a block of memory
COMPARE two blocks of memory
HEXADECIMAL/DECIMAL calculator converter
XIT two machine language monitor with: Display, Go, Register, Save, Load and Exit commands

Only $49.95 for the VIC or Commodore 64 version. The high-speed cartridge with machine language assemble costs just $89.95. And the easy-to-use Arrow comes with the clear Skyles manual.

AR20 (VIC-20), AR64 (CBM-64) $49.95
ARAS20 (VIC-20), ARAS64 (CBM-64) 89.95

For CBM-64 and VIC owners only:

This is just 1 of 20 pages of the newest and biggest Skyles catalog, hot off the press.

We know you'll want this page, in its full 7¼ x 9 splendor, and another 19 pages of peripherals, software and books that will make your CBM-64 or VIC computer even nicer to live with. So, if we missed sending you your very own copy within the last few weeks, call us at **(800) 227-9998** **(unless you live in California, in which case call (415) 965-1735.**

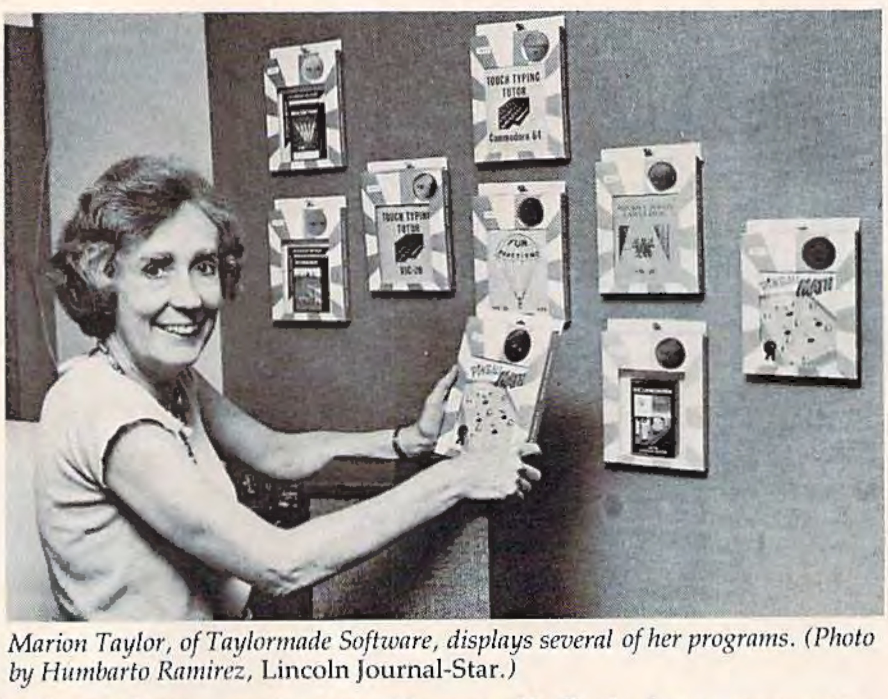
From Skyles Electric Works, the oldest and largest professional specialists in the business.

Skyles Electric Works
231-E S. Whisman Road
Mountain View, California 94041

Marion Taylor
The Programmer Behind *Touch Typing Tutor*

Kathy Yakal, Editorial Assistant

It's much harder to use a computer if you don't know how to type. Judging from the number of typing tutorials available these days, and the success of many of them, lots of people are learning to type for the first time on their home computer keyboards. Here's a look at the programmer behind one of the best-selling typing programs available for the Commodore 64 and VIC-20.



Marion Taylor, of Taylormade Software, displays several of her programs. (Photo by Humbarto Ramirez, Lincoln Journal-Star.*)*

The programmer is a woman. And she's been gainfully employed as a computer programmer for 28 years.

Those two facts make this month's subject of "Inside View" a bit unusual. "I'm old enough to be the mother of lots of these people who are programming best sellers, and the grandmother of some of the kids that are using the programs," says Marion Taylor of Taylormade Software, the programmer behind *Touch Typing Tutor*.

That's not the only thing that sets Taylor apart from the usual software author, who is typically a male under 30 years old. She's also a one-woman show. She works out of her home in Lincoln, Nebraska, and runs all aspects of her company: product development, marketing, and, of course, programming all of the products herself. That's becoming very unique in these days of rapidly expanding software companies and increasingly divided labor in the software industry.

Your ship has just come in.

Aboard the S.S. Slipshod, things are getting a little out of hand.

It seems that the navigator mistook an iceberg for Pittsburgh. With predictable results.

Soon, every stateroom on the ship will have an ocean view.

But never fear, 'cause Captain Clutch is here.

He'll stoke the boilers to keep the game afloat, lead the passengers to the lifeboats and rescue a few bars of gold along the way.

Why, you ask, doesn't the courageous Captain simply pocket the gold and row off into the sunset?

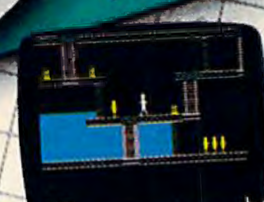
Because our Captain is above that sort of thing, that's why.

And besides, the game won't let him.

By the way, how long can you tread water?

Developed by Steve Morris.
Available on diskette
for your Commodore 64™

WATERLINE!



TRONIX™

8295 South La Cienega Blvd.
Inglewood, CA 90301 (213) 215-0529

Commodore 64 is a trademark of Commodore Electronics, Ltd.

Scientific Background

Taylor was graduated with a mathematics degree from Pomona College in California and started her programming career on the first-generation vacuum tube computers. "That was before the days of computer classes," she says. "The company that hired you also trained you.

"Those first computers were not able to perform both scientific and business functions. The next generation was able to, but that was still prior to the days of high-level languages [such as BASIC]. Basically, you could only perform one run a day of a given program."

Due to her husband's job with the military and her own changing career, Taylor moved around the country quite a bit, living and working in a total of six states. She worked for places like Westinghouse, the University of Wisconsin, and the Livermore Radiation Lab.

Shifting Gears

Then came a move to Lincoln, Nebraska, a job at the University of Nebraska, and the introduction of microcomputers. Taylor bought an Apple in 1979 and started writing programs on it in her leisure time.

But what started as a leisure-time hobby turned into a full-time obsession. "I found I couldn't work eight or nine hours a day, raise a family, and do all the programming I wanted to do on micros," Taylor says. "I decided to devote my work time to micros. That's where all the fun and creativity is."

Taylor didn't start out by programming games, as many programmers do. "I wanted to write programs that had lasting value. That led me to educational programming."

When the VIC-20 was introduced in 1981, Taylor switched her focus from Apple to Commodore because she liked its features, and because its low price made it very accessible to people. Then came the Commodore 64 and even more programming capability.

More Than Fun

So what makes a good educational program?

"First," says Taylor, "it has to be educationally sound. It has to have lasting value—it can't be so simple that it can be done in 15 minutes. The writer of the program has to be familiar enough with school methods so the children aren't confused." This, she says, can be something as simple as using an asterisk in place of the multiplication sign in a mathematical program. If children are not familiar with BASIC programming, they don't know what that means.

An educational program must also go beyond what a child can read in a book. "Micros have the potential to teach concepts in a more concrete way than books," she says.

Marion Taylor's Touch Typing Tutor.

Realizing that her sons had learned fractions in grade school but still didn't really understand them was an eyeopener for Taylor, and it led her to write another of her many programs, *Fun Fractions*. "I try to achieve a good balance between drill and instruction in what I write. That way, children can better understand what they're learning."

The graphics and sound capabilities of micros greatly enhance the educational process, says Taylor. She cites turtle graphics as an example of good educational software. "Children enjoy moving the turtle around the screen to build things, but they're really learning about loops, arrays, and other higher-level programming tools.

"But the fun aspect of a program is only one element of educational programs. Some of the programs on the market today deal only with that aspect and slight the educational side."

What's Next?

The volatile nature of the microcomputer industry today makes it difficult to do long-range planning, Taylor believes. "I'm always planning new programs, because that's what I like to do. But it's difficult to predict very far into the future as far as what specific programs I'll be doing."

And though she admits that right now she's a bit of a shock to people, being an old hand at programming and a woman, Taylor expects to see more of that in the future. "People have always said that boys are better at math and science than girls. They learn better hand-eye coordination from an early age because of the types of things that they are encouraged to do. Having computer training in the schools from an early age will change that."

This Christmas, Atari and Commodore* owners will go out of their minds.

It'll take about 30 seconds.

Because once you boot a Datamost game on your Atari home computer, or your Commodore 64, you'll come face to screen with the most mind-blasting games ever.

And what better time to go out of your mind than Christmas?

Our Music Will Have You Hearing Things.

Going out of your mind never sounded so good.

Because now our games have music. You heard right. Music. Original Music. Throughout. And scored just for our newest releases. You'll go nuts over it.

Our Graphics Will Have You Seeing Things.

You and everybody else.

Because our games are so great–how great are they?–they're so great you'll want to play them again and again and again. And then your friends will go bonkers over them.

And they'll want to play. And then your family will want to play. And then total strangers off the street will want to play and...

Mind-blowing arcade-quality action like this makes it one mad world, pal.

We'll Torment You Right From The Start.

No time to settle down and get comfy.

The tension begins building the moment you boot the game.

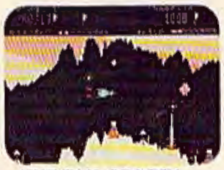
Terrific screen titles tease you with the game's objective, scenario, characters.

And while you wait, hand twitching over the Joystick, you'll find out what planet you're on.

What the fuss is all about.

Why you're involved. And perhaps of singular importance to you personally, how to keep from being obliterated. So get ready to get the most out of your mind.

By getting the most out of ours.



THE TAIL OF BETA LYRAE.™ Changes as you play. Unpredictable. Impossible to master. (No one has!)

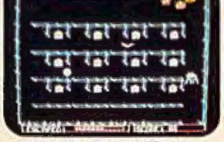
DATAMOST™ INC

The most out of our minds.™

Datamost, Inc., 8943 Fullbright Ave., Chatsworth, CA 91311, (213) 709-1202

Atari is a trademark of Atari Computer. *Commodore 64 is a trademark of Commodore Business Machines, Inc. TM Registered Trademark of Datamost.

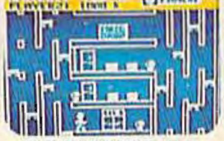
COSMIC TUNNELS.™ Four games in one. Four times the challenge! Incredible graphics.



MONSTER SMASH.™ Deathly strategy. Mash the monsters! Let the visitors live.

NIGHTRAIDERS.™ Strafe a city under siege with 3D angled selective firing!

ROUNDABOUT.™ Sharpens your shoot 'em up skills. 24 different game screens. Habit-forming!



COHEN'S TOWERS.™ You're the mailboy in a big city skyscraper. Work your way to the top.

MR. ROBOT.™ Screens scream with color, action and sound! Design your own screens, too.

Cave-In For VIC

Paul L. Bupp and Stephen P. Drop

"Cave-In" is an excellent three-dimensional maze game which uses a screen-flipping technique to swap screen displays. The game requires a joystick and runs on the unexpanded VIC-20.

When you play "Cave-In," you become the newly appointed foreman of a mining operation. After completing your initial inspection, you believe that a cave-in is imminent. You realize that you must explore every tunnel to find and rescue all of the miners.

Taking into consideration your unfamiliarity with the mine, you decide to make a map of the shafts as you travel.

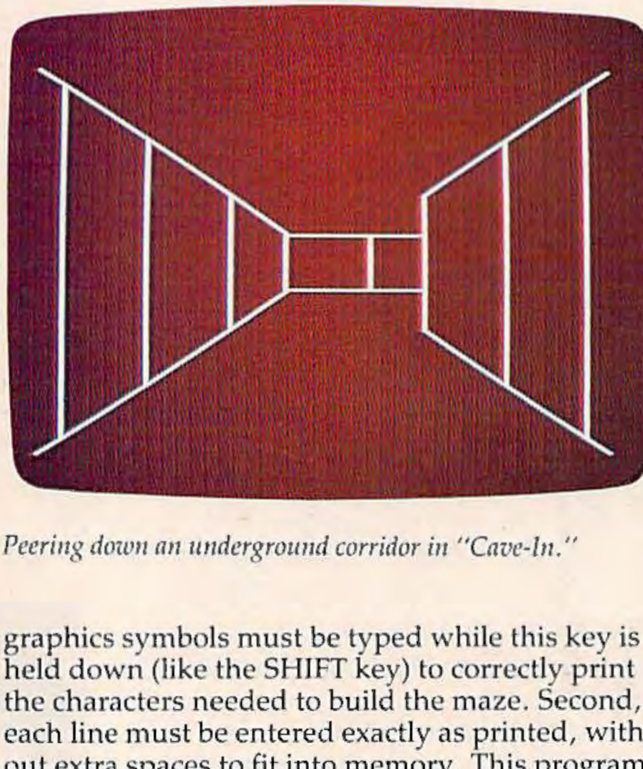
To refer to your map, push the fire button on the joystick. Push it again and you return to the mine. The dark circle on the map is where you started and must return to escape the mine safely.

Just as you expected, no sooner do you find the last miner and warn him of the danger than the cave-in begins. Now you have to get out before the falling rock traps you. Aren't you glad you made the map? (In the advanced game, however, you lose the map after the cave-in starts, so you must rely on your memory to recall the maze-like passages.)

Other Game Controls

You may view instructions at any time by pressing the f1 special function key. However, once you see the instructions, you face a fresh maze upon returning to the game. To travel through the tunnels, change directions by moving the joystick right or left, and then move forward by pushing the stick forward.

Observe some precautions when typing this program. First, it requires using the Commodore key at the lower left of the keyboard. Some of the



Peering down an underground corridor in "Cave-In."

graphics symbols must be typed while this key is held down (like the SHIFT key) to correctly print the characters needed to build the maze. Second, each line must be entered exactly as printed, without extra spaces to fit into memory. This program uses all but about 15 of the 3583 available memory locations. The program will *not* run correctly with any memory expansion boards. Third, you will probably need to abbreviate some keywords to enter a few of the lines in the program, such as line 52. See your manual for legal keyword abbreviations.

If you have difficulty entering the game successfully, or if you prefer not to type the program, just send a blank cassette, self-addressed stamped mailer, and $3 to:

Paul L. Bupp
21724 124th Ave. SE
Kent, WA 98031

See program listing on page 198.

ZAP

D'FUSE

TIC ATTACK

BOMBER WORD

RES Q

KEY QUEST

DOT-A-LOT

SPACE BATS

THE HOTTEST NEW GAMES FOR YOUR VIC 20® ARE FROM TYMAC®

Light up your VIC 20 with these hot new games. Arcade style action right in your living room. Exciting, challenging ways to get more fun out of your computer. Buy these cartridges now from your favorite computer products retailer.

Limited Time Offer: Send proof-of-purchase of any two of these games and $49.95 and receive TYMAC's EXPAND-O-RAM, a $119.00 retail value. EXPAND-O-RAM gives you two switchable cartridge slots and 16K of additional memory. See details inside each package.

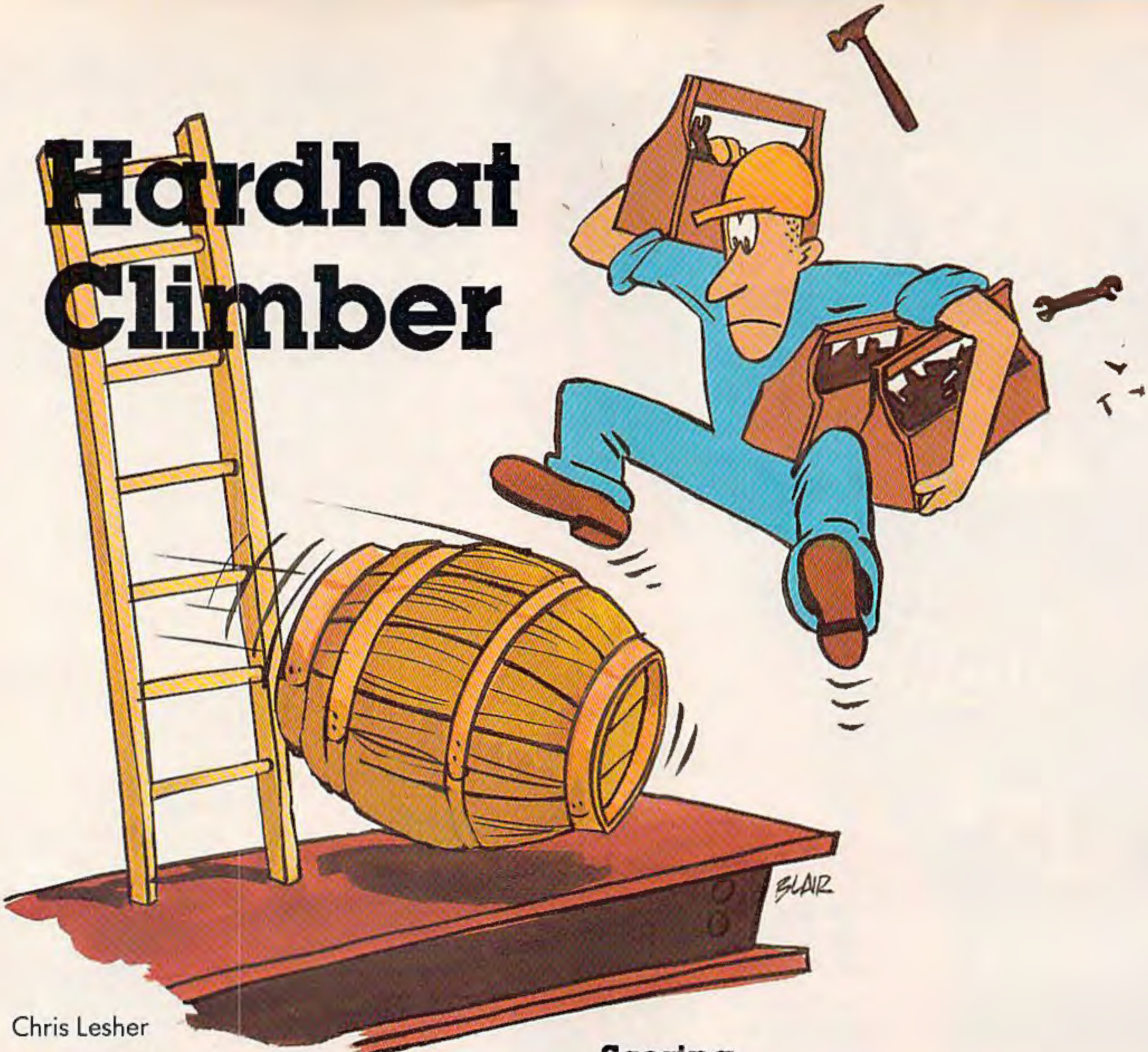
TYMAC

TYMAC INCORPORATED
127 MAIN STREET
FRANKLIN, NJ 07416
201 827-4050

TYMAC is a registered trademark of Tymac Controls Corporation

VIC 20 is a trademark of Commodore Electronics Ltd

Hardhat Climber



Chris Lesher

"Hardhat Climber" is one of the best games we've seen for the unexpanded VIC-20 and is an excellent example of what can be accomplished with BASIC. We've included an adaptation for the Commodore 64.

You are standing at the bottom of four levels of girders, connected by ladders. At the top is a pile of 12 barrels and scattered along the girders are toolboxes. The object of "Hardhat Climber" is to walk around the girders and pick up every toolbox while avoiding the barrels that roll down at you. If you pick up all of the toolboxes you are rewarded bonus points, and you move on to a more difficult screen.

I wrote the VIC-20 version of Hardhat Climber almost entirely in BASIC, with only a short machine language routine to check the joystick. Using the stick, you can move the climber up, down, left, and right along the girders and ladders. Pressing the fire button makes your climber jump in the direction he was last moving. He can jump over barrels and holes in the girders.

Scoring

You score 150 points for every toolbox you pick up, 1000 points for jumping over a barrel, and 100 points for each barrel remaining after you have picked up all the toolboxes. The score is displayed in the upper-left corner of the screen. The number of the screen is displayed in the upper-right corner. The number of climbers you have left is displayed between the score and screen number.

You begin the game with three climbers and earn an extra one every 10,000 points. A climber is lost if he is hit by a barrel, walks off a girder, or has not picked up all the toolboxes by the time all 12 barrels have rolled off the pile. The game ends when you lose your last climber.

Many program lines in the VIC version are longer than the maximum limit of 88 characters. They must be entered by abbreviating the keywords and omitting the space between the line number and first keyword. The abbreviations may be found in the manual that came with the computer. If there is an error in any of these lines, the entire line must be retyped using the abbreviations again. Also be sure to save the program

If you've ever dreamed of playing the pro-tour golf circuit on the world's finest courses...

Or fantasized about getting ringside seats to all the championship fights in history...

Or drooled over the prospect of managing the best pro baseball teams of all time...

We've just answered all your hopes and wishes!

And SSI's 14-day "satisfaction or your money back" guarantee makes sure our games will help make your dreams of sports glory come true. In fact, the only way they won't is if you don't hurry to your nearest computer/software or game store and get these fine strategy sports games today.

For the APPLE® and COMMODORE 64™:

These games ($39.95 each) come on mini floppy disk for the APPLE® II with Applesoft ROM Card, Apple II Plus, Apple IIe, or Apple III.

Also on disk for the Commodore 64.™

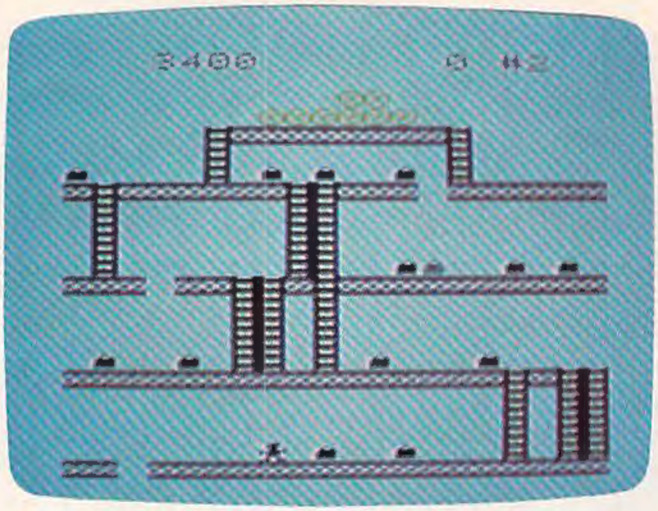
Apple is a registered trademark of Apple Computer Inc.

Commodore 64 is a trademark of Commodore Electronics, Ltd.

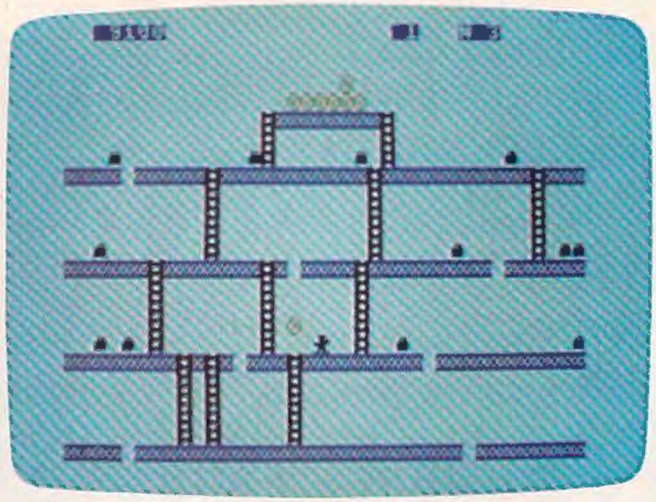
STRATEGIC SIMULATIONS INC

If there are no convenient stores near you, VISA & Mastercard holders can order direct by calling **800-227-1617, ext. 335 (toll free).** In California, call 800-772-3545, ext. 335. To order by mail, send your check to: Strategic Simulations Inc, 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043. (California residents, add 6.5% sales tax.) Please specify computer format.

WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES.



Collecting toolboxes on the first floor in a VIC game of "Hardhat Climber."



A falling barrel narrowly misses the player on the second floor (64 version).

before running it in case there is a mistake in the machine language in lines 106-108. If any of the numbers in these lines are mistyped, you could lose the program.

The VIC version lines, which are especially long, include lines 37, 56, 71, 73, 77, 81, 101, 102, 106, 107, and 108. Remember when you enter these lines with abbreviations while using the "Automatic Proofreader," the checksum number will not match up. (See "Simple Answers To Common Questions" in this issue.)

See program listings on page 195.

AARDVARK *Action Software*

PRESENTS. . .

ZEUS

WITHOUT QUESTION, ABSOLUTELY —

The Fastest

ARCADE STYLE GAME

AVAILABLE, THAT WE KNOW OF.

FAST AND FURIOUS. You become the wizard fighting off the thunderbolts of an angry Zeus. Multiple action. Are you good enough to keep up as the difficulty increases wave after wave. Hours of challenging fun. This is one game you may never completely master.

AVAILABLE ON: TRS-80C(16K), CMD 64, VIC 20(13K)
STOCK #1007 TAPE:$24.95 DISK: $29.95

PYRAMID... Our toughest and most intrigueing adventure. Average time though the pyramid is 50-70 hours. Clues everywhere and some very ingenious problems. This one is popular around the world.

AVAILABLE ON: TRS-80C (16K) CMD 64 VIC 20 (13K) TIMEX/SINCLAIR TI/99-4A
STOCK #5002 TAPE: $19.95 DISK $24.95

DUNGEONS OF DEATH. . . A serious D & D type game for you up to 6 players at a time. You get a choice of race and character, on screen mapping of the maze, a chance to grow from game to game, and a 15 page manual. Requires extended basic on TRS-80C.

AVAILABLE ON: TRS-80C(16K), CMD 64, VIC 20(13K), TRS-80C 32K DISC
STOCK #7001 TAPE: $19.95 DISK: $24.95

QUEST. . . A different kind of adventure. Played on a computer generated map of Alesia. You lead a small band of exployers on a mission to conquer the Citadel of Moorlock. Takes 2-5 hours to play and is completely different each time.

AVAILABLE ON: TRS-80C(16K) CMD 64 VIC 20(13K) TIMEX/SINCLAIR TI99/4A, EXTENDED BASIC REQUIRED ON TI
STOCK # 7007 TAPE: $19.95 DISK: $24.95

Quest is also available on TRS-80C as a 32K extra memory, extra fun adventure. It's a little more difficult, and will test the most experienced explorer of the kingdom.

STOCK #7006 TAPE: $24.95 DISK: $29.95

AARDVARK *Action Software*

IS AVAILABLE AT SOFTWARE RETAILERS EVERYWHERE. IF YOUR LOCAL COMPUTER OR SOFTWARE STORE DOES NOT CARRY IT — ASK THEM WHY NOT. SEND ONE DOLLAR FOR ILLUSTRATED CATALOG. DEALER INQUIRES INVITED.

HOURS OF FUN AND ENJOYMENT FROM THE "ADVENTURE PLACE"

TO ORDER: SEND AMOUNT INDICATED PLUS $2.00 SHIPPING, PER ORDER. INCLUDE STOCK NUMBER, QUANTITY DESIRED AND YOUR PREFERENCE OF TAPE OR DISK. BE SURE TO INDICATE TYPE OF SYSTEM, AMOUNT OF MEMORY, ETC. WHEN USING CHARGE CARD TO ORDER BY MAIL, BE SURE TO INCLUDE EXPIRATION DATE.

CHARGE CARDS WELCOME VISA

1-800-624-4327

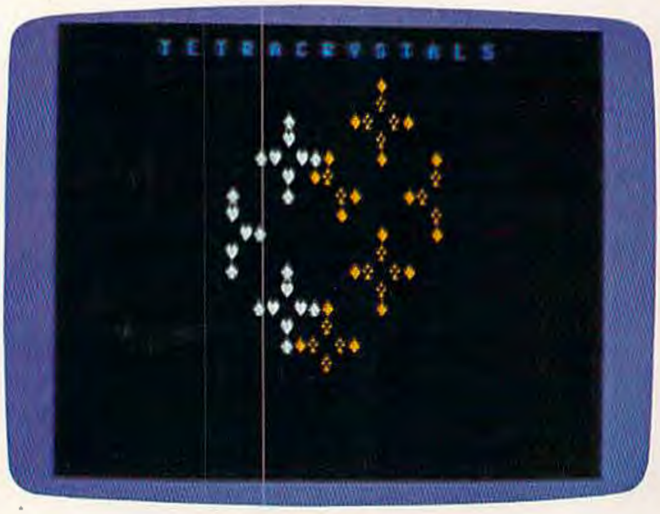
PHONE ORDERS ACCEPTED
8:00 A.M. TO 8:00 P.M. E.S.T. MON-FRI

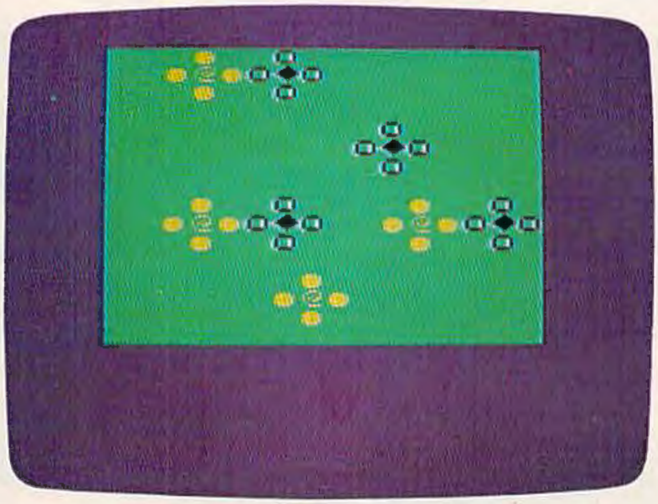
AARDVARK *Action Software*

2352 S. COMMERCE RD. WALLED LAKE, MI 48088 313/669-3110

Tetracrystals Of Veluria

Todd Heimarck





This nonviolent, noncompetitive game for the unexpanded VIC-20 and Commodore 64 produces fascinating patterns of colorful crystals. It also incorporates some advanced programming techniques, including page-flipping, a very smooth method of animation.

The Story

The prospectors have failed. They searched every inch of the Velurian asteroid belt and discovered no minerals of commercial value, except for a strange type of ice crystal. These "tetracrystals" grow into large crystals when dropped in water and exposed to sunlight.

Crystal Growth

There are two ways the tetracrystals can grow. First, each one goes through four stages of growth: seed, monad, tetrad, and shell. After the fourth stage, the shell collapses and melts into plain water.

Second, a tetrad always tries to sprout four new seed-crystals at right angles to the tetrad. These seed-crystals draw energy from the tetrad (which is why it becomes a shell), and then go through the four stages of growth. When the seeds grow into tetrads, they sprout new seeds, and so on.

Limits To Growth

The new seeds will grow only if they have space, sunlight, and water. That means there are three limits to growth. First, a seed needs space to grow, so it cannot be put into a space that is already occupied. If a tetrad is right next to another crystal (in any stage), it will not plant a seed in that space. The other three seeds can still grow, unless they are affected by the limits on growth.

Second, a seed needs energy (sunlight) to grow. Tetrads and shells cast shadows that block

NEW EXCITEMENT FOR COMMODORE 64*

J-Bird™

Hop into excitement with J-Bird! As you rush to change the color of your 3-D pyramid of blocks you have to avoid the critters that are out to make a meal of you. You're going to love the exciting sound and animation of J-BIRD. There are magic balls, jumping frogs, frisky cats, coily snakes and a host of other obstacles out to do you in.

SPECIAL FEATURES INCLUDE: incredible graphics and sound; sound on/off option; two player option; joy- stick or keyboard operation; multiple difficulty levels; high scores saved on disk and cassette. These features make J-BIRD enjoyable for everyone.

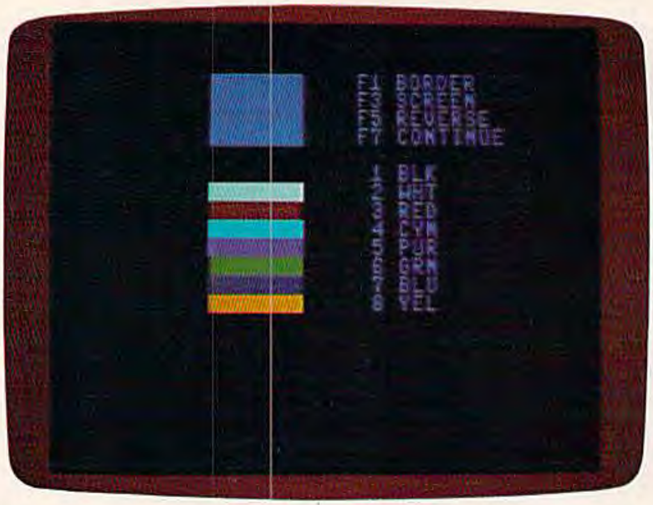
Ask for J-BIRD at your local computer store.

ORION SOFTWARE

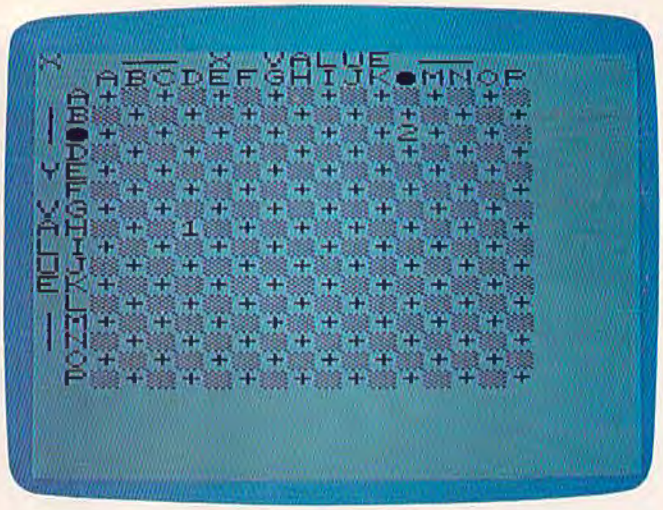
For your nearest dealer, or to order direct call: **1-800-821-8088.**

To order by mail send $29.95 for J-BIRD to: ORION SOFTWARE, P.O. BOX 2488, Dept. 1283, Auburn, Alabama 36831.

MasterCard VISA



The opening screen in "Tetracrystals" lets you choose your own screen/border color combination (Commodore 64 version; VIC similar).

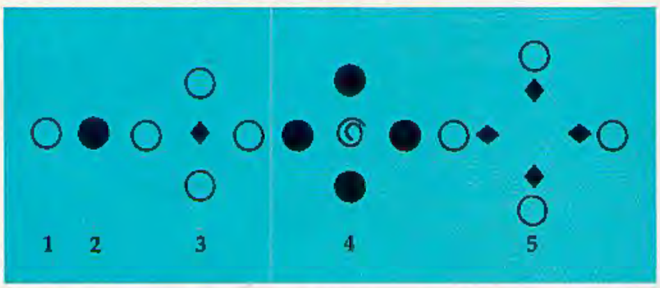


After selecting the speed and character set, you plant different kinds of crystals on a grid (VIC version; 64 similar).

sunlight from new seeds. A seed will not grow in a space right next to a tetrad or shell. This rule takes care of the problem of two tetrads trying to put seeds into the same space.

Third, a seed will grow only in water. A tetrad on the edge of the 16 × 16 grid cannot put seeds outside of the border (because there is no water there).

How A Crystal Grows



The seed-crystal in stage 1 becomes a monad in stage 2 and a tetrad in stage 3. In stage 3 it sprouts four new seeds, which then go through the stages. Note that in stage 5, the shell has disappeared and only four new seeds were generated by the four tetrads. That is because of the limits on growth. They are not allowed to put seeds in a space that is next to another tetrad or shell. All four of the tetrads tried to put a new seed in the center and all failed.

Tetracrystals is simple enough to play with paper and pencil, but you would have to erase and draw over and over again. Using the computer is quicker and easier.

How To Play

When the game first starts, you see eight color bars. You can change the screen color with the function keys:

f1 changes the color of the border.

f3 changes the color of the screen.

f5 switches "reverse" (inverse video) on and off.

f7 starts the game, after you have chosen the colors.

Before you press f7, notice which colors show up on the screen you pick. If you plant crystals that are the same color as the screen, they will grow, but they will be invisible.

Next you pick the speed. Zero, the lowest number, yields the fastest game. Then choose a character set (1, 2, 3, or 4); I prefer number 1.

Finally, you plant the crystals. In the upper-left corner of the screen you will see a prompt (X, Y, P, or C). Choose an X-coordinate (A through P), a Y-coordinate (A through P), plant the crystal (1 for a seed, 2 for a monad), and choose the color (type a number between 1 and 8; the color is the same as the color printed on the key).

Up until the point when you choose the color, you can cancel your choices by pressing f1. When you are ready to start the game, press f7.

Options During The Game

Seven of the function keys allow you to control growth during the game. The f1, f3, f5, and f7 keys (unSHIFTed) control the amount of sunlight that reaches the crystals. Press f7 twice to reduce the light (all growth will stop). After you freeze the picture (with f7), you can watch the growth step by step by pressing f7. If you want to go back to continuous growth, f1 restores the game to normal. The f3 key slows the growth and f5 speeds it up. If the growth is very fast, press f3 and a number from 1 to 9. The higher the number, the slower the growth. Press f5 (plus a number) to speed up the growth.

The f2 and f4 keys throw more seeds onto the field. The f2 key gives you a random-colored seed

at a random location. Press f4 (and a number from 1 to 8) and you will get a seed the same color as what is printed on the number key, planted randomly on the screen.

The f6 key stops the game and returns you to the beginning. "Tetracrystals of Veluria" uses part of memory for machine language routines, so if you want to stop playing altogether, it is a good idea to turn your computer off and then on again before you load another program.

To recap:

f1 continues the game (after f7 step by step).

f3 slows the growth (type 1–9 to continue).

f5 speeds up growth (type 1–9 to continue).

f7 allows step-by-step growth.

f2 plants a random-colored seed at a random location.

f4 plants a seed at a random location (choose the color with 1–8).

f6 starts a new game.

Strategies

Tetracrystals is a simple game. You can plant two types of crystals, up to eight different colors. There are two rules for growth and three limits on growth. Most children will understand how it works.

But like Reversi (also known as Othello), simple rules hide the many subtleties of play. The more you play it, the more interesting variations you discover.

There are no rules for winning or losing. I originally wrote Tetracrystals as a nonviolent, noncompetitive game.

If you don't like games without competition or winners and losers, you can make up variations. You and a friend can choose two different colors and plant crystals around the screen. If one color takes over the screen, that player wins. In some cases—if you start with symmetrical positions, for example—neither color will take over the screen and you would have to call it a tie game.

If you plant just one seed, it will spread to take over about half the screen, and then (because of the limits on growth) it will disappear. It is a good idea to start with at least two crystals.

In the reverse video option, the crystals leave behind traces of where they have been.

For some reason, crystals that start near the edges have a slightly better chance of surviving (remember this if you decide to play competitively). And usually, if you plant seeds *and* monads, one or the other will take over the screen; they don't coexist very well. Imagine the 16×16 grid as a chessboard, with alternating black and white squares, because it will make a difference if your starting positions are all on the same color squares or on opposite colors.

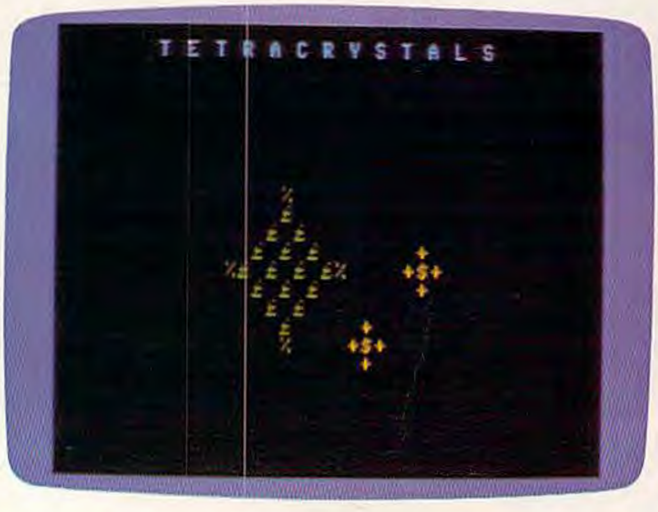
You can also try the sandwich maneuver. If you plant a seed somewhere on the grid and then plant seeds of different colors, one right above and one right below, I think you'll be surprised at the results. The crystal in the middle will usually crowd out the other two crystals.

Special Typing Instructions

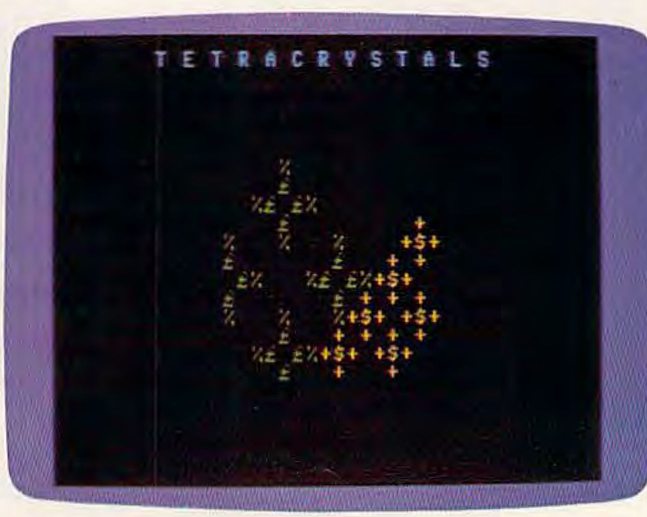
The VIC version of Tetracrystals needs two programs to run, and you must enter them in the correct order. Program 2 (the one with all the DATA statements) contains the machine language routines. Program 1 will not run without them.

Follow this procedure for the VIC version (note the minor program changes for disk below):

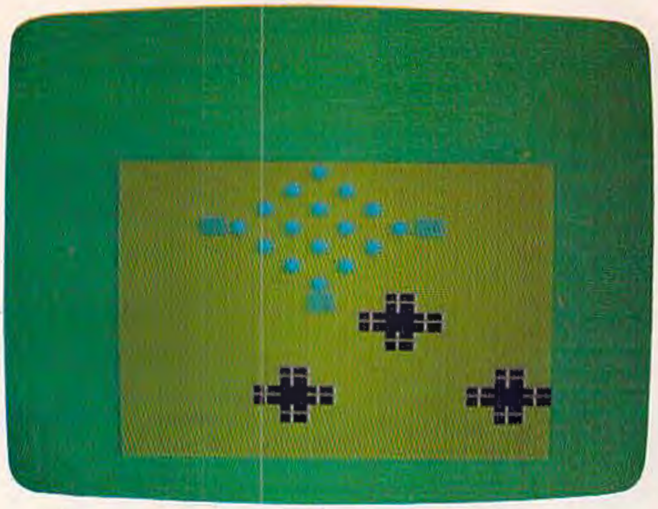
1. Type in Program 1 first, *but do not RUN.* Instead, SAVE it on tape or disk. If you're using tape, I recommend saving Program 1 as the first program on a fresh cassette. For tape, use the filename "CRYSTALS T1/JAN". For disk, use the



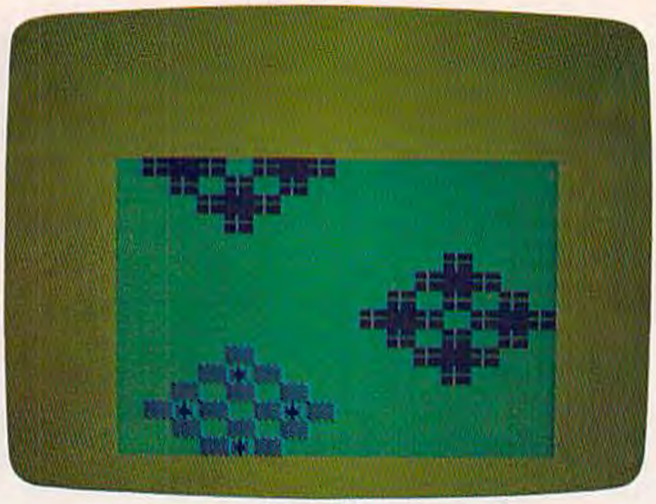
Crystals of various colors begin growing and interacting, as seen in these Commodore 64 screens...



...sometimes overlapping and crowding out other crystals.



Crystals divide and multiply in these VIC screens...



...forming larger crystals, and occasionally dominating all the space around them.

filename "CRYSTALS D1/JAN". Do not rewind the tape after saving.

2. Type NEW and press RETURN.

3. Type in Program 2. *Do not SAVE it yet.* Contrary to standard procedures, Program 2 should be RUN before it is SAVEd. When you type RUN, Program 2 first checks itself for typing errors and warns you of any mistyped DATA statements. If the DATA statements are entered correctly, the program waits for you to press a key to continue. Before continuing, make sure the tape or disk with Program 1 is in the cassette recorder or disk drive. With cassette, make sure the tape is positioned just past Program 1 (which is where it will be if you left it alone after SAVEing Program 1). Now, when you press a key to continue, Program 2 will begin creating a data file on your tape or disk. Program 2 automatically names the data file "CRYSTALS T3/JAN" for tape or "CRYSTALS D3/JAN" for disk. If you're using cassette, you will notice the tape stopping and starting by itself as the data file is created. This is normal. Do not press the STOP button on the recorder until the data file is finished and the screen says READY.

4. When the screen says READY, the data file is created. Now is the time to SAVE Program 2. Use the filename "CRYSTALS T2/JAN" for tape or "CRYSTALS D2/JAN" for disk. You won't need Program 2 again unless you want to create another data file, perhaps for backup.

5. Finally the game is prepared. To play, LOAD and RUN Program 1 (filename "CRYSTALS T1/JAN" or "CRYSTALS D1/JAN"). When you type RUN and press RETURN, Program 1 automatically begins reading the data file created by Program 2. (That's why it's so important to make sure the data file immediately follows Program 1 if you're using tape.) As the data file loads, you'll see numbers appearing on the screen. This is normal. When all the data is read by Program 1, the game begins.

If you press the RUN/STOP key while playing and want to restart the game, don't type RUN, because the program will look for the data file again. Instead, type RUN 13 (which starts running the program at line 13).

One more caution: Tetracrystals takes up almost all of the available memory in the VIC; there will be only a few bytes left. *It is vital to type in the programs exactly as listed.* Don't add any extra spaces. If your VIC gives you an ?OUT OF MEMORY error after the data file is read by Program 1, you can safely delete line 10 and lines 5000–5100 to play the game. Once these lines put the machine language into memory, you don't need them anymore (but don't SAVE the program after doing this).

Modifications For Disk

The 64 version works on either tape or disk. But the VIC programs, as listed, are designed for tape. A few minor changes need to be made for disk.

In Program 1, change line 5000 to:

```
5000 OPEN1,8,0,"CRYSTALS D3/JAN"
```

(This change tells the computer to read the data file from disk instead of tape.)

In Program 2, change these lines:

```
35 PRINT"GET DATA DISK READY[3 SPACES]HIT
   A KEY TO CONTINUE
40 OPEN1,8,1,"0:CRYSTALS D3/JAN"
```

(Line 35 alters the prompt to read "DATA DISK" instead of "DATA TAPE". Notice there is no closing quote on the PRINT statement; closing quotes are optional in Commodore BASIC, and leaving it off saves one byte of memory. Line 40 tells the computer to create a data file on disk

TAKE A TIP FROM SANTA
TRI MICRO SOFTWARE
belongs on your holiday shopping list. Providing a wide range of gift selections, the choices can satisfy a variety of tastes from wise men to elves -- word processing, file management, spread sheet analysis, home and entertainment packages, or complete collections.
Choose Tri Micro... Santa's Choice.
INTERNATIONAL TRI MICRO
4122 E. Chapman Ste 30
Orange, CA 92669
(714) 771-4038

instead of tape, with the appropriate filename expected by line 5000 in Program 1.)

Page-Flipping

The animation here is not the usual erase-and-draw method.

In most games, when you want a character to move, you tell the computer where the character currently is and where it should be next. The computer erases the old character and then draws it at the new location. If the screen is full, you can see the changes being made, from the top-left corner to the bottom right.

Page-flipping is a much smoother method of animation.

I got the idea from "Alternate Screens" by Jim Butterfield in *COMPUTE!'s First Book Of VIC*.

Normally an unexpanded VIC uses two pages of memory (7680 to 8191) for screen memory. Memory location 648 is a pointer that tells the VIC where to find the screen. By POKEing 648 with a different number, you can change the location of screen memory.

In Tetracrystals, one screen is visible, the other hidden. While you are watching one screen, the computer is drawing the next picture on the hidden screen. When the new picture is ready, a couple of POKEs (lines 1100 and 1120) make the new picture visible. Then, while that screen is up, the next one is being drawn on the hidden screen.

How The Program Works—VIC Version

Line 10 protects the memory for the second screen and the machine language instructions. It jumps to subroutine 5000, which reads the ML instructions from tape.

Lines 13–20 set variables.

Line 60 jumps to subroutine 1100, which restores the screen to the normal location (beginning at 7680).

Line 100 jumps to subroutine 8200 (which sets the screen to the usual colors and restores it to a 22 × 23 size), then subroutine 6000 (which sets up the new screen color).

Lines 160–197 set up the speed and character set.

Lines 255–395 plant the crystals on the screen.

Lines 510–795 are the heart of the program—where the crystals grow. First there are three SYSes to ML routines. Since the visible screen has been cut down to 16 × 16 (256 bytes) and there are 512 available for each screen, that means there are 256 bytes below each screen. This is what I call the "shadow screen." The first SYS clears the shadow screen of the hidden screen. The second SYS controls the direct growth (seed to monad, monad to tetrad, and so on). The third SYS grows the brand-new seeds. Then BASIC takes over. The program goes through a delay loop (with the counter HF, for "How Fast") and checks for keyboard input. If one of the function keys has been pressed, it takes care of what needs to be done. Then, if the CP flag has been set, it stops until it gets the "step-by-step" instruction. Subroutine 1000 changes the random (but melodic) music. The last two SYSes translate from the shadow screen to the not-now-visible screen and to color memory. Finally, subroutine 1100 flips the screen memory to the other screen.

See program listings on page 191.

Look at these Features

- Fully screen-oriented
- Horizontal and vertical scrolling
- Terminal mode — never seen before on a wordprocessor
- Supports Commodore disk and cassette handling
- Imbedded commands

Bliztext

Wordprocessor for Commodore 64

BLIZTEXT is a trademark of ELCOMP PUBLISHING, INC.

Commodore-64 and VIC-20 are trademarks of Commodore Business Machines.

Dealer and Distributor inquiries are invited.

BLIZTEXT — SUPER WORDPROCESSOR for the Commodore-64

— ON SALE NOW! —

- Fully screen-oriented, up/down, left and right scrolling — Upper and lower case
- More than 70 commands
- Full I/O compatibility with Commodore peripherals Upper and lower case
- Works with practically every printer on the market, user definable printer control commands
- INCLUDE command allows handling large files on up to 4 diskettes or on cassette.
- Build in terminal software for electronic mail and networking. Telecommunications mode, upload and download, save on disk or cassette.
- Dynamic formatting, Imbedded commands
- Single keystroke for disk directory and error channel
- Program comes on disk or cassette
- Double line spacing, left and right margin justification, centering, page numbering, and practically everything one expects from a good wordprocessor.

AVAILABLE NOW!

Order # 4965	$89.00
Manual only (62 pages)	$29.95

MACROFIRE - Editor/Assembler for the Commodore-64

ON SALE NOW

AVAILABLE IMMEDIATELY

One outstanding tool, consisting of 3 powerful elements combined into one efficient program!

1.) Fully screen-oriented Editor (more than 70 commands)
2.) Very fast assembler with macro capability
3.) Machine Language Monitor

Assembly can be started from the editor. Translates in 3 passes. More than 1,000 lables, screen oriented/no line numbers, scrolling, includes disk files.

Practically everything the serious machine language programmer needs everyday!

Manual only	$19.95
Order # 4963	$89.00

THE GREAT BOOK OF GAMES, VOL.I, by Franz Ende

46 programs for the Commodore 64

Introduction to graphics and sound. How to program your own games. Walking pictures, animation, high resolution graphics, programming tips and tricks, hints and useful subroutines for the beginner and advanced programmer. This book is a MUST for every C-64 owner.

Come and get it — It's yours for only $ 9.95

Order # 182 — 128 pages — $9.95

Programs from the book on disk.

Order # 4988 — $19.95

MORE ON THE SIXTYFOUR, by H.-C. Wagner

How to get the most out of your powerful Commodore 64. Very important subroutines, tricks and hints in machine language for your C-64. How to modify DOS. How to connect a parallel and serial printer. How to design your own terminal program for communication and networking. Dig into I/O for cassette and disk.

Order # 183 — $9.95

Programs from the book on disk

Order # 4989 — $19.95

NEW PRODUCTS

Watch out for our new books, software and add-ons to come soon. ON SALE NOW! -- ORDER TODAY!

How to program in 6502 Machine Language on your C-64, by S. Roberts (Introduction)

Order-# 184 — $12.95

Commodore-64 Tune-up, Vol. I, by S. Roberts

How to expand and customize your C-64.

Order # 185 — $12.95

Small Business Programs for the Commodore-64 by S. Roberts

How to make money using your C-64. Mailing list, invoice writing, inventory, simple wordprocessing and much more.

Order # 186 — $12.95

Hardware Add-Ons:

Parallel printer interface KIT	Order # 4990 $ 19.95
Direct Connect Modem KIT	Order # 4991 Ask f.price
Universal Experimenter Board	Order # 4970 $ 9.95
Expansion Board, space for four experimenter boards(board only)	Order # 4992 $ 29.95

For your VIC-20

Tricks for VICs — $ 9.95

Universal Experimenter board

HOFACKER

PAYMENT: check, money order, VISA, MASTER CARD, Eurocheck, ACCESS, Interbank
Prepaid orders add $3.50 for shipping (USA)
$5.00 handling for C.O.D.
All orders outside USA: add 15 % shipping, California residents add 6.5 % sales tax.

ELCOMP PUBLISHING, INC
53 Redrock Lane
Pomona, CA 91766
Phone: (714) 623 8314
Telex: 29 81 91

ELCOMP PUBLISHING, INC.
53 Redrock Lane
Pomona, CA 91766
USA
Phone: (714) 623-8314
Telex: 29 81 91

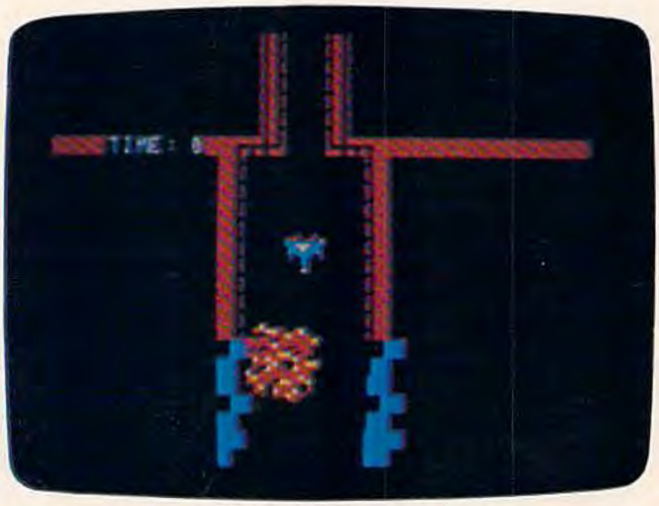
ELCOMP Computer (S) Pte. Ltd.
89 Short Street
Unit 03-07, Golden Wall Auto Centre
SINGAPORE 0718
Phone: 3382623, 3388228
Telex: 56516

Ing.W. Hofacker GmbH
Tegernseerstr. 18
D-8150 Holzkirchen
West-Germany
Phone: 08024 / 73 31
Telex: 52 69 73

Canyon Cruiser

Thomas Catsburg

Originally written for the Commodore 64, "Canyon Cruiser" has been adapted for the unexpanded VIC-20 as well. The game works with either keyboard controls or a joystick.



A multicolored asteroid approaches the player's spaceship in "Canyon Cruiser" (64 version).

Commodore 64 owners are usually hungry for games. For a long time there was not much software to choose from, although the situation has improved considerably in recent months.

There are, of course, hundreds of games written for the older Commodore PET computers available. Using one of the PET emulators on the market, many of these games will run with little or no modification on the Commodore 64.

Unfortunately, these games do not take advantage of the Commodore 64's advanced features—such as sprites, custom characters, and synthesized sound. These features just weren't available on the PETs.

Updating An Old Favorite

"Canyon Cruiser" is an updated version of an old favorite on the PET. The idea is to guide your spaceship through a narrow canyon. The walls keep getting closer, naturally, so the game keeps getting harder.

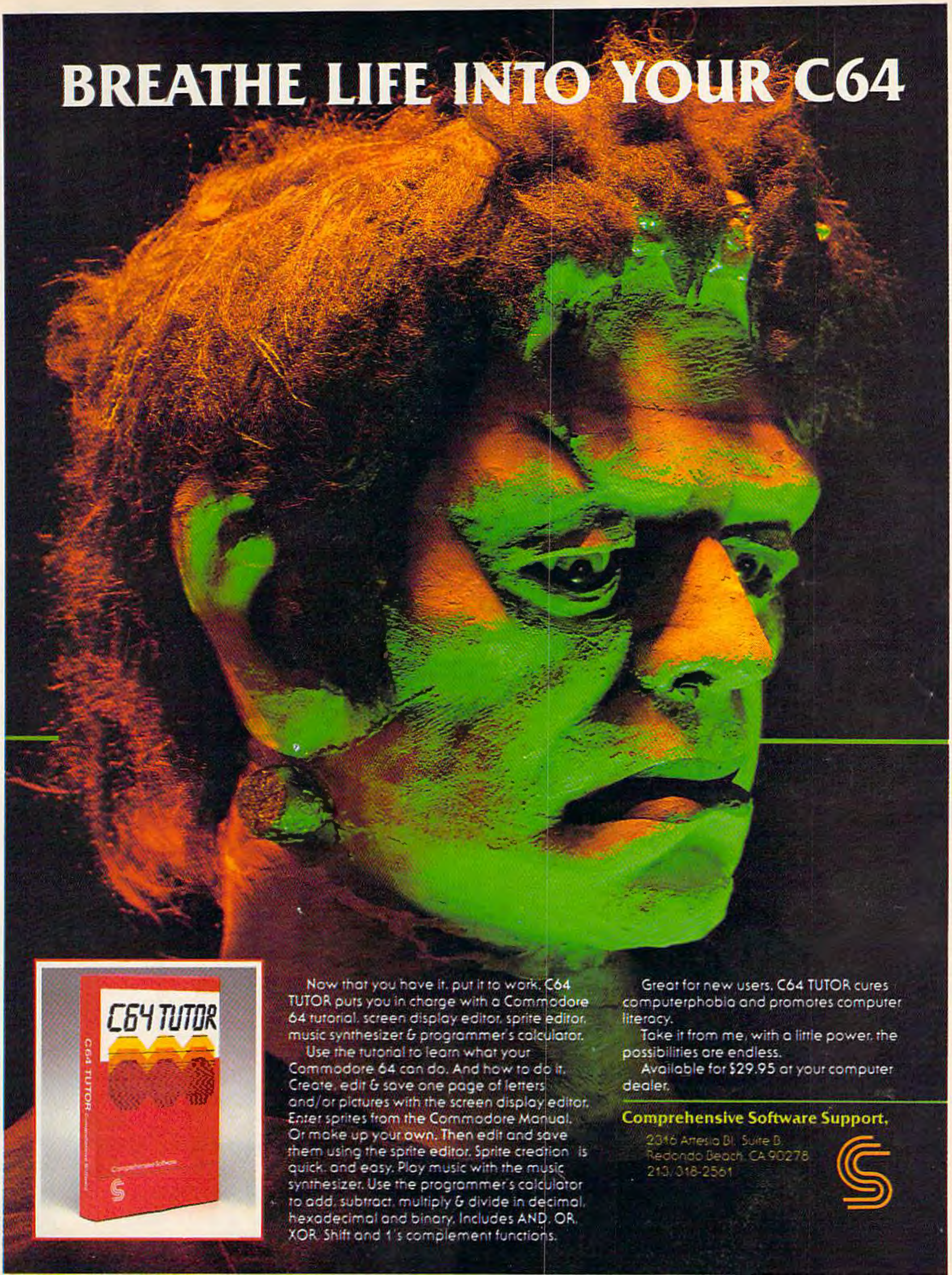
Starting with this basic concept, I improved the Commodore 64 adaptation by making the spaceship a multicolored sprite and by adding a new twist—wandering asteroids. The asteroids, also sprites, cannot destroy your ship. But they do add to the visual confusion while passing by. In that sense they can be considered an additional hazard.

The VIC-20 lacks sprites, so all the shapes in the VIC version are created with custom characters.

Cruisin' For A Bruisin'

Canyon Cruiser transforms you into the pilot of a new spaceship. Your goal is to test the craft to its limits by flying it down the funnel-shaped canyon without crashing into the unyielding walls.

To control the spaceship from the keyboard, steer it left or right with the colon (:) and semicolon (;) keys. You can also use a joystick if you prefer (on the Commodore 64, plug the stick into port 2).


BREATHE LIFE INTO YOUR C64
C64 TUTOR
Now that you have it, put it to work. C64 TUTOR puts you in charge with a Commodore 64 tutorial, screen display editor, sprite editor, music synthesizer & programmer's calculator.
Use the tutorial to learn what your Commodore 64 can do. And how to do it. Create, edit & save one page of letters and/or pictures with the screen display editor. Enter sprites from the Commodore Manual. Or make up your own. Then edit and save them using the sprite editor. Sprite creation is quick and easy. Play music with the music synthesizer. Use the programmer's calculator to add, subtract, multiply & divide in decimal, hexadecimal and binary. Includes AND, OR, XOR, Shift and 1's complement functions.
Great for new users. C64 TUTOR cures computerphobia and promotes computer literacy.
Take it from me, with a little power, the possibilities are endless.
Available for $29.95 at your computer dealer.
Comprehensive Software Support,
2316 Artesia Bl. Suite B
Redondo Beach, CA 90278
213/318-2561

cardco, inc.

It's time for your computer to grow up.

VIC-20 EXPANDER SALE!

• 8K RAM MEMORY EXPANDER (Free $16.95 Game)	$ 69.95	**$39.00**
• 16K RAM MEMORY EXPANDER "CARDCO" (Free $29.95 Adventure Game)	$ 99.00	**$55.00**
• 24K RAM MEMORY EXPANDER (Free $29.95 Adventure Game)	$159.00	**$79.00**
• 32K RAM MEMORY EXPANDER (Expands to total memory to 57K (57,000 bytes)	$199.00	**$95.00**
• 3 SLOT "CARDCO" SWITCHABLE EXPANDER BOARD	$ 39.95	**$29.00**
• 6 SLOT "CARDCO" SWITCHABLE **EXPANDER BOARD** RIBBON CABLE EXPANDER BOARD (Lifetime warranty)	$ 99.00	**$49.00**
• 40-80 COLUMN BOARD	$129.00	**$69.00**

VIC 20

40-80 COLUMN BOARD

SALE **only $69⁰⁰** SALE

Now you can get 40 or 80 Columns on your T.V. or monitor at one time! No more running out of line space for programming and making columns. Just plug in this board and you immediately convert your VIC-20 computer to 40 or 80 columns! Comes in an attractive molded case with instructions!
List $129.00 **Sale $69.00**

FOR ONLY $19.95 you can get a 40-80 Column Board "WORD PROCESSOR" with mail merge and terminal emulator PLUS! AN ELECTRONIC SPREAD SHEET (like Visicalc) the word processor requires 8K—mail merge 16K! List $49.90. **Sale $39.90.** *If purchased with board only $19.95. (Tape or Disk.)

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and Master-Card. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

SANYO MONITOR SALE!!

9" Data Monitor

- 80 Columns × 24 lines
- Green text display
- East to read - no eye strain
- Up front brightness control
- High resolution graphics
- Quick start - no preheating
- Regulated power supply
- Attractive metal cabinet
- UL and FCC approved

- *15 Day Free Trial - 90 Day Immediate Replacement Warranty*

9" Screen - Green Text Display	**$ 79.00**
12" Screen - Green Text Display (anti-reflective screen)	**$ 99.00**
12" Screen - Amber Text Display (anti-reflective screen)	**$119.00**
14" Screen - Color Monitor (national brand)	**$249.00**

Display Monitors From Sanyo

With the need for computing power growing every day, Sanyo has stepped in to meet the demand with a whole new line of low cost, high quality data monitors. Designed for commercial and personal computer use. All models come with an array of features, including up-front brightness and contrast controls. The capacity 5 × 7 dot characters as the input is 24 lines of characters with up to 80 characters per line.

Equally important, all are built with Sanyo's commitment to technological excellence. In the world of Audio/Video, Sanyo is synonymous with reliability and performance. And Sanyo quality is reflected in our reputation. Unlike some suppliers, Sanyo designs, manufactures and tests virtually all the parts that go into our products, from cameras to stereos. That's an assurance not everybody can give you!

SANYO

Official Video Products of the Los Angeles 1984 Olympics

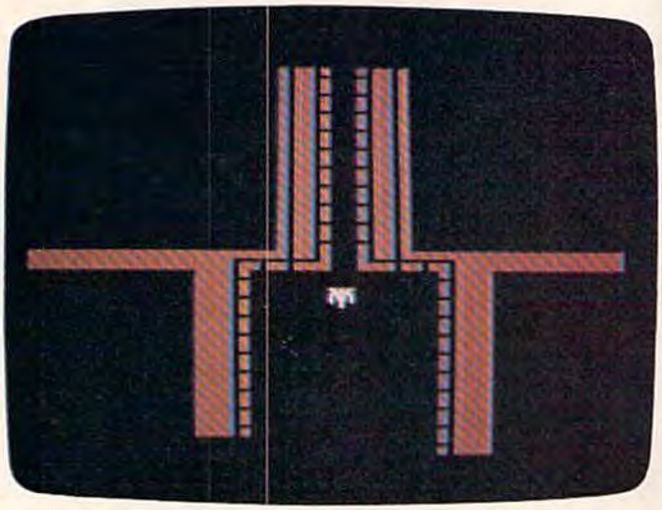
• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY • BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. Visa - MasterCard - C.O.D.

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order



Leaving the base at the beginning of a game of "Canyon Cruiser" (VIC version).

The spaceship changes color to warn you that the canyon is narrowing.

At the end of the flight you will be ranked according to your skill. There are 11 ranks ranging from "Starter" to "Han Solo." If you want, you can change these ranks to anything you want by locating these lines in the program and modifying them. (But be sure to get the program working as published before tinkering with it.)

See program listings on page 188.

VICTORY SOFTWARE
INTRODUCES
THE EARTH WARRIOR SERIES

METAMORPHOSIS $19.95
You stumbled into the nest of the Cyglorx and find yourself fighting off robot tanks guarding the Cyglorx eggs. You think you have everything under control and then the eggs start hatching. Commodore 64 version features 4 different screens.
Available for COMMODORE 64 and VIC-20. Played with JOYSTICK.

CREATOR'S REVENGE $19.95
The creator assembled a massive army of robots and insects to take revenge on the earth. Destroy insects, get treasures, and get the neutron bomb deactivator. Battle robots and destroy the neutron bomb before it annihalites your city. Miss and you must face the mutants. Features 4 different screens.
Available for COMMODORE 64. Played with JOYSTICK.

LABYRINTH OF THE CREATOR $19.95
Journey into the most complex and dangerous fortress ever build by the creator. You will encounter deadly robots, skulls, lakes, avalanches, false creators, and a creature who roams 256 rooms relentlessly pursuing you.
Available for COMMODORE 64. Played with JOYSTICK.

BOUNTY HUNTER $19.95
An adventure in the Old West. Journey back with us into the days of Jessie James and Billy the Kid where the only form of justice was a loaded revolver and a hangman's noose.

In this full-length text adventure, you play the role of Bounty Hunter, battling against ruthless outlaws, hostile Indians, wild animals and the elements of the wilderness with only your wits and your six gun. Average solving time: 20-30 hours. If you love adventures, this one is a real treat.
Available for COMMODORE 64 and VIC-20 (with 8K or 16K expander). Played with Keyboard.

CAVE KOOKS $14.95
Earth's surface is threatened by collapse from a strange group of creatures who bore out the earth's crust to make their dens. Your objective is to enter the creature's habitat and spin the invaders to death.
Available for COMMODORE 64. Played with JOYSTICK.

Programs for the VIC-20 and the COMMODORE 64.

All games available on TAPE or DISK.

All Arcade-Style games are packed full of MACHINE CODE.

KONGO KONG $19.95
Climb ladders, avoid the barrels the crazy ape is rolling at you, and rescue the damsel. Commodore 64 version features 4 different screens!

Available for COMMODORE 64 and VIC-20. Played with JOYSTICK or KEYBOARD.

GRAVE ROBBERS $19.95
Introducing the first GRAPHIC ADVENTURE ever available for the VIC-20 or COMMODORE 64! With realistic audio-visual effects, you explore an old deserted graveyard and actually see the perils that lie beyond.
Available for COMMODORE 64 and VIC-20. Played with KEYBOARD.

CHOMPER MAN $19.95
Don't let the bullies catch you as you gobble the goodies! This program has 8 screens and still fits in the standard memory.
Available for COMMODORE 64 and VIC-20. Played with JOYSTICK or KEYBOARD.

Illustrations: Elizabeth Hauck

Check your LOCAL DEALER or order directly.
ORDERING: We accept personal checks, money orders, VISA, and MASTERCHARGE. Charge orders please include number and expiration date. Overseas orders please use charge or have check payable through a U.S. bank. Add $1.50 postage and handling per order. PA residents please add 6% sales tax.

VISA

VICTORY SOFTWARE

VICTORY SOFTWARE INC.
7 Valley Brook Road, Paoli, PA 19301
(215) 296-3787

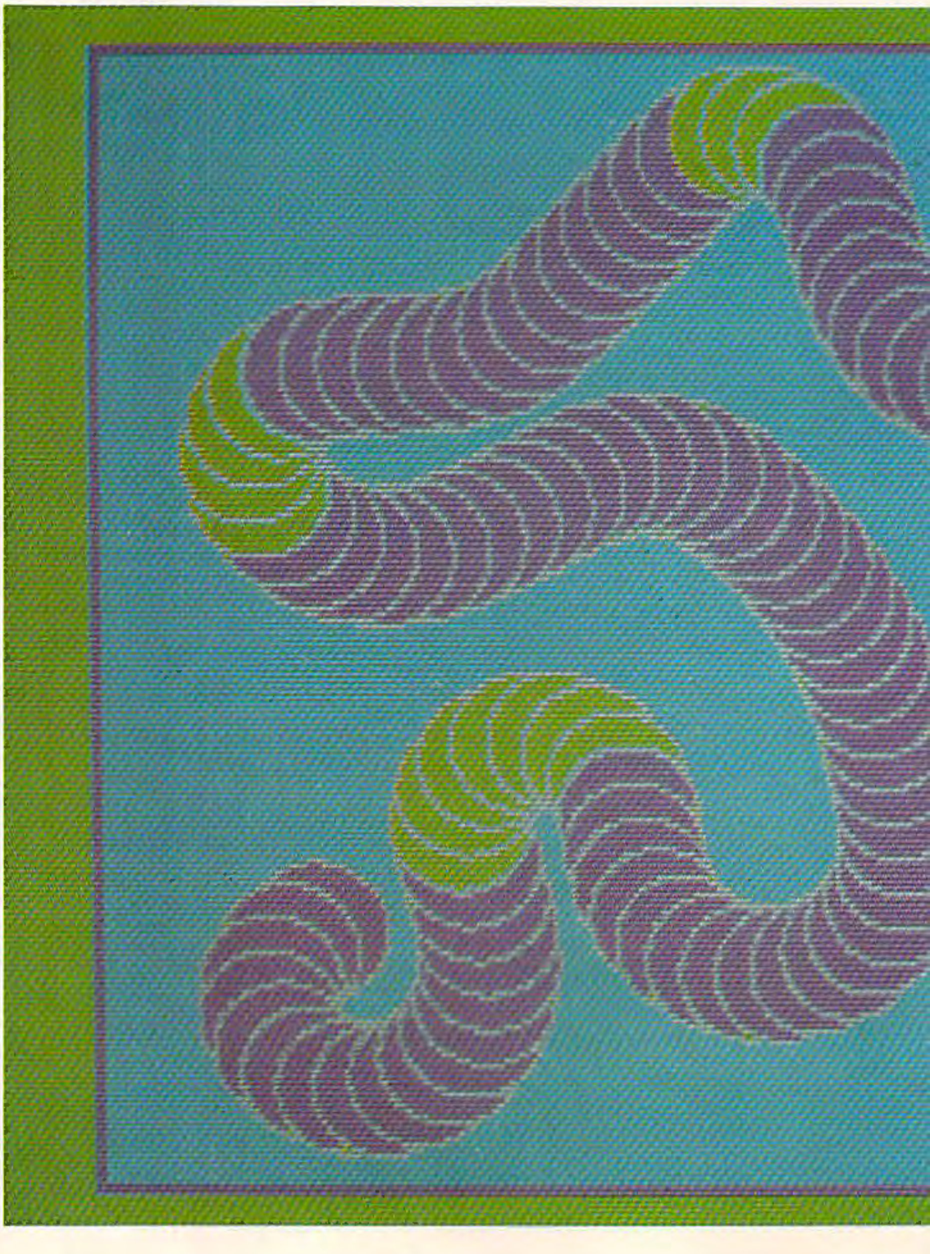
COMPUTING

for families

New Family Learning Games

Fred D'Ignazio, Associate Editor

This month COMPUTE!'s GAZETTE is consolidating two previous columns—the monthly "Computing For Kids" and the bimonthly "Computing For Grown-Ups"—into one new monthly column, "Computing For Families." Each month, Computing For Families will cover topics of interest to all members of home-computing families, both young and old. And as before, the column will be written by Fred D'Ignazio, himself the head of a home-computing household.



These colorful designs and the ones on following pages were created with Spinnaker Software's Delta Drawing.

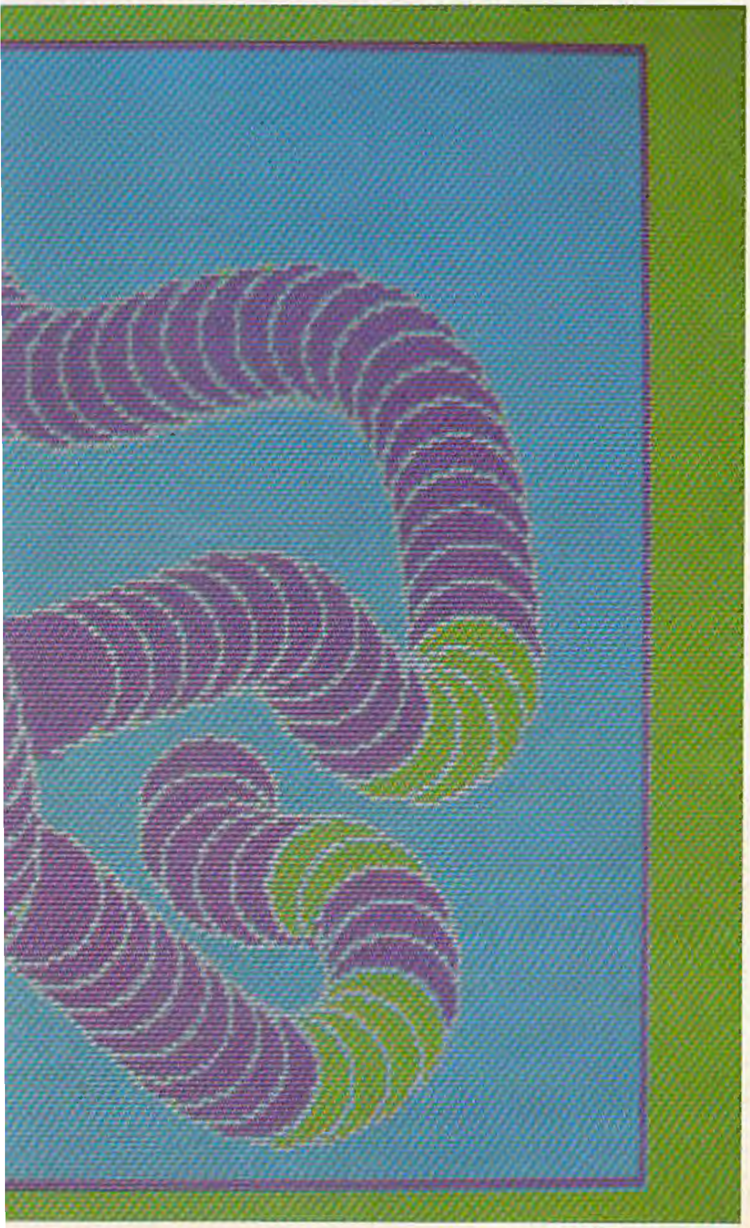
Ambushing The Mailman

When I was a kid I used to belong to all sorts of mail-order book clubs. When I knew a book was coming I would rush home from school or spend an entire Saturday prowling around my front yard waiting for the mailman to pull up our driveway. It was a great feeling when he brought a big box addressed to me. I knew that a new book was inside the box. It didn't matter that I could never remember what book I had ordered. That was part of the fun.

Now I have two children (Catie, 8, and Eric, 4) who have followed in my footsteps. Catie and Eric get as excited as I did about receiving packages in the mail, and they are as good as I was at ambushing the mailman. On Saturday mornings, they lie in wait behind two big pine trees just outside the porch door. When the mailman arrives they spring out and grab all his packages and run into the house. They reach the living room, and they start ripping the packages apart.

But do you think they are looking for books?

Nope. They are looking for new computer learning games arriving in the mail. And they act like wild things until they unwrap the games, load them into the computer, and begin playing them.



Champions And Cheerleaders

Here, below, is a group of seven games that captivated me and my family. They are remarkably diverse and quite varied in the thinking and skill they demand from the human player. But they are similar in four key traits. First, they are just as much fun for adults as they are for kids. Second, they can be played at many different levels, so, with help from an adult, even a toddler can benefit from them. Third, the games are constructive and nonviolent. They let families build things rather than train them in creative destruction. Fourth, the games are much more fun when people play them together.

All these games encourage interaction among family members, schoolmates, and friends. At our house we almost always play the games together. The approach we use is to have one person step forward as the stalwart champion and have the other family members be coaches, cheerleaders, and the peanut gallery. At the end of each game we rotate all the roles.

Playground Or Swamp?

It was interesting to see Eric and Catie approach these new games. They never want to read any directions before starting. They equate direction-reading with "adult," "slow," "dense," and "boring." On their own, they never read directions—*unless they appear on the screen*. They just boot up a disk or plug in a cartridge. Then they start madly pressing buttons or swiveling a joystick or game paddle. Pretty quickly something begins happening. Then it's "play it by ear" all the way.

This sort of approach makes me very nervous. Nevertheless, I usually climb aboard for the ride, just to see where we'll all end up.

Most of the time, the kids wander through a program—*at a gallop*—and usually figure out what's going on. Then they begin playing with a passion.

But sometimes their approach is akin to turning down a blind alley and running, full-speed, into a brick wall. Then, with their noses out of joint, the kids turn around to me and announce, "Daddy, this is a dumb game!"

And, that's that. When the computer doesn't respond, when my kids feel powerless and out of control, they abandon the game. It's like watching their playground turn into a yucky swamp. While it's still a playground they love to race around

and use the equipment. But when suddenly the ground turns into sticky glue, the kids feel paralyzed.

I used to think they had reached a real dead end. Now I look at it as an opportunity to start doling out some game rules and special "power" buttons that get the game moving again and put the kids back in control.

Pipes

Pipes is available on cartridge for the VIC-20 ($29.95) or the Commodore 64 ($34.95). It won the 1983 CES (Consumer Electronics Show) Software Showcase Award for Home Education. It is made by:

Creative Software
201 San Antonio Circle
Mountain View, CA 94040
(408) 745-1655

Pipes is a game that never turns into a swamp. When the program begins there is a plumber, a house, and a water supply tank on the display screen. On the far right is a pipe factory with bins full of pipes of all shapes.

My kids were puzzled by *Pipes* at first. But that didn't stop them from leaning on the joystick and racing the little plumber around the screen. They learned how to use the "radar" display—a little window in the corner of the screen that lets you see the plumber, the house, the water tank, and the pipe factory, all at the same time. And, by randomly pressing the joystick buttons, they discovered they could buy pipes from the factory and hook them up to the house and the water tank.

The first couple of times we played the game the kids created some pretty weird plumbing. Pipes squirreled out of the house, then corkscrewed, pirouetted, and pretzeled themselves into oblivion. We found out how to turn on the water supply (by pressing the V key on the computer) and squirted water all over the ground with ecological abandon.

Eventually we ended up with some pretty decent plumbing. The pipes went in efficient right angles out of the water tank and into the house. When we turned on the water, it flowed in a direct route from the tank to the house.

After hooking up the plumbing to one house had become a snap, we graduated to a whole neighborhood with up to five houses. We even figured out how to do the plumbing with the cheapest pipe and save the most money.

Now the kids mostly play *Pipes* alone. The other day I went into the dining room and found Eric busy building a circular pipe network out of the water tank. I frowned and screwed up my face. "Why would you want to do that?" I asked him.

"Because," he said, not looking up, "this way the water never goes away."

Delta Drawing

My daughter Catie and I reviewed the Apple version of *Delta Drawing* in the June 1983 issue of COMPUTE! Magazine. Now Spinnaker Software has released *Delta Drawing* on cartridge for the Commodore 64 ($39.95). You can reach Spinnaker at:

Spinnaker Software Corporation
215 First Street
Cambridge, MA 02142
(617) 868-4700

The Commodore 64 version of *Delta Drawing* is significantly more powerful than the earlier Apple version. And the Apple version was a knockout.

Catie and I found *Delta Drawing* to be a lot like Logo—only upside down! To make the Logo turtle do something you have to define a procedure (or program) and type in lots of one- or two-letter commands. Then, when you're all done, you have to type the procedure name to make the turtle do its tricks.

This kind of programming is called delayed gratification. It requires a lot of patience—especially when you are only four years old.

Delta Drawing is just the opposite. The payoff comes at the beginning *and* at the end. Here's a typical session with Eric:

Eric plugs the *Delta Drawing* cartridge into the Commodore 64, and, a moment later, a triangle and a blinking dot appear in the center of an empty screen. The triangle is "DeeDee" the turtle. The dot is DeeDee's tail. DeeDee uses her tail to draw.

Eric starts DeeDee on a trip across the screen by pushing the D key (for Draw). DeeDee moves about a quarter of an inch up the screen, then stops. Behind her is a white line.

Eric pushes the D key again, then the R key (for turn right 30 degrees) three times. Then he

COMPUTERS ARE CREATING JOBS FOR NRI-TRAINED PEOPLE.

IF YOU'RE SERIOUS ABOUT MAKING MONEY IN MICROCOMPUTERS, NRI IS SERIOUS ABOUT SHOWING YOU HOW.

The U.S. Department of Labor projects job openings for qualified computer technicians will soon double. International Resource Development, Inc., estimates a 600% increase in these jobs in a decade. And most of these will be new jobs, created by the expanding role of computers.

NEVER HAS THERE BEEN A FASTER-GROWING FIELD OF TECHNOLOGY.

Many people are afraid of losing their jobs to computers, but thousands of jobs will be created for those who are prepared to meet the challenge.

With NRI training, you'll be prepared. You can have a profitable, exciting future as an expert who can handle the operational, programming and technical aspects of all kinds of microcomputers and microprocessors.

LEARN IN YOUR SPARE TIME.

NRI trains you in your own home, at your convenience...no classroom schedules to meet, no need to quit your job. As a class of one with complete course materials and the backing of a staff of professional electronics instructors, you'll get extraordinary hands-on training on the latest, most popular microcomputer: the new TRS-80™ Model 4 with Disk Drive. Designed to perform diverse personal and business functions and accept more software, the TRS-80 is yours to keep.

Your NRI course will include the new TRS-80 Model 4 with Disk Drive or the TRS-80 Color Computer with NRI Computer Access Card...plus a professional LCD multimeter, NRI Discovery Lab and hundreds of demonstrations and experiments. It's all yours to keep.

LEARN HOW TO USE, PROGRAM AND SERVICE STATE-OF-THE-ART MICRO-COMPUTERS.

Through your carefully designed NRI course, you'll get a wealth of practical experience. You'll build circuits... from the simplest to the most advanced...with your NRI Discovery Lab.® You'll use a professional 4-function LCD digital multimeter for analysis and troubleshooting. With NRI training you'll explore your computer's registers, memory and input-output ports. You'll even write programs to control the circuits you've designed and built. You'll perform hundreds of challenging experiments, always backed by a full-time faculty ready to help you personally.

When your NRI training is complete, you'll be a computer technician, ready for your first job — servicing, testing or programming all types of microcomputers — in a rewarding and challenging new career.

THE CATALOG IS FREE. THE TRAINING IS PRICELESS.

Send the coupon today for your FREE 104-page catalog. It's a valuable guide to opportunities and training in the high-tech revolution. You'll see how easily you become part of the growing high-tech world of microcomputers. If the coupon has been removed, write: NRI, 3939 Wisconsin Ave., Washington, DC 20016.

TRS-80 is a trademark of the Radio Shack division of Tandy Corp.

NRI School of Electronics
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016

We'll give you tomorrow.
The catalog is free.
The training is priceless.

All career courses approved under GI bill.
☐ Check for details.

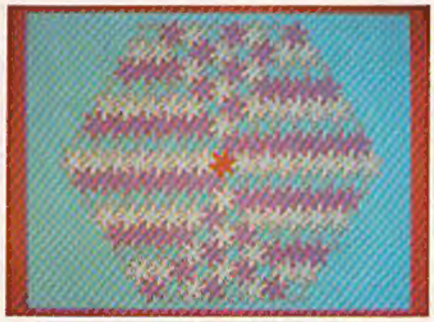
Please check for one free catalog only.
- ☐ Industrial Electronics
- ☐ Computer Electronics Including Microcomputers
- ☐ Color TV, Audio, and Video System Servicing
- ☐ Electronics Design Technology
- ☐ Digital Electronics
- ☐ Communications Electronics
  • FCC Licenses • Mobile • CB • Aircraft • Marine
- ☐ Basic Electronics
- ☐ Small Engine Servicing
- ☐ Appliance Servicing
- ☐ Automotive Servicing
- ☐ Auto Air Conditioning
- ☐ Air Conditioning, Heating, Refrigeration & Solar Technology
- ☐ Building Construction

Name (Please Print) Age

Street

City/State/Zip

Accredited by the Accrediting Commission of the National Home Study Council



pushes the S key.

Eric has made DeeDee do something significant by pushing just five buttons. First, he has made DeeDee move and draw a line—as soon as he presses the button. (This is called immediate gratification.)

Second, he has just created a *program*. The program is extremely simple, but it will act as a building block for the shapes that Eric is planning to make DeeDee draw next.

Eric saves his program by pressing the S key. (At this point Eric's daddy likes to press the T—Text—key to see the actual commands Eric has given DeeDee. This reassures Eric's daddy that Eric is, in fact, creating a real program. Eric, however, is confident that he is programming even without seeing the list of commands. He can see that his programs are working by watching DeeDee whiz around the screen drawing the shapes he has dreamed up.)

When Eric presses the S key the screen goes blank and DeeDee reappears in the home position. With only a moment's hesitation, Eric presses the X and the 1 buttons to run Program 1. DeeDee spurts forward two paces and turns right. Behind her is the straight line.

Eric presses the X and 1 buttons three more times. When he is done DeeDee is back in her home position. She has just drawn a square. Eric types the R button to turn DeeDee 30 degrees to the right. Then he types an S to save his second program.

Next Eric presses the X and the 2 keys seven times to run Program 2 seven times. When he is finished he smiles. DeeDee has just created a flower made up of little boxes rotated around a central axis.

Is Eric done? Not yet. He likes flowers so much he wants them all over the screen, and he wants them in different colors. He presses a couple more keys and colors the flower petals orange and blue and green. Then he presses the M button and holds it down. DeeDee scoots up the screen. Eric presses the S button to save his third program.

Now he's finally ready to do his picture. To make the picture he uses the building block Programs 1–3 that he has just created. To fill the screen with colorful flowers, he has to press only two keys: the X and the 3. Each time he runs his third program, DeeDee draws a flower, colors it in, then zips to a new part of the screen.

Pretty soon Eric and DeeDee have filled the entire screen with flowers. Eric is done. He gets up from the computer and goes looking for his family to show off his latest creation.

The Tip Of The Iceberg

Delta Drawing is a spectacular learning game. I have described only a tiny bit of what kids can do with it. But the neatest thing about *Delta Drawing* is that children can explore all its powerful features, or they can spend hours on a single part of *Delta Drawing* and still not exhaust it. The program is made for children to explore. And if my children are any guide, they love doing it.

Kids On Keys

Kids on Keys is available from Spinnaker Software. The Commodore 64 disk costs $29.95; the Commodore 64 cartridge costs $34.95.

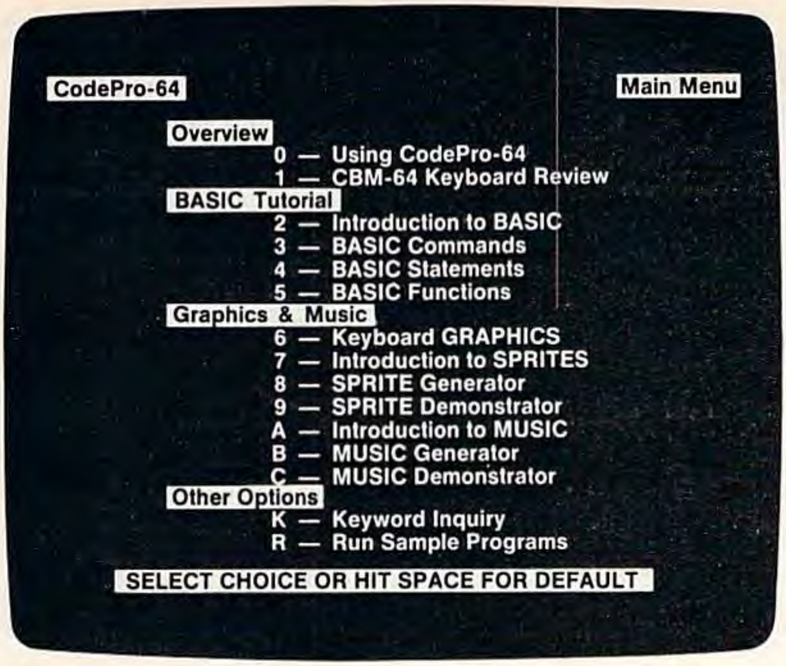
Kids on Keys is one of those programs that my family ought to like. It teaches all sorts of good things like the alphabet, shapes and colors, problem-solving, and, last but not least, the computer (or typewriter) keyboard.

It turns out that my family really does love *Kids on Keys*, but not because it teaches all that sound educational stuff. We love it for lots of little intangible reasons, like the neat music it plays. Or like the little person who whimsically floats up and down in a balloon. Or the way the letters we correctly identify make a loud *BURP!* and crumble like cookies. Or the funny way the cats, rabbits, boots, and faces fly off the screen after we correctly identify them.

Somehow, subtly, and disarmingly, *Kids on Keys* is charming. So we all love to play it. (Even though some of us are 34 years old, and we're supposed to already know our alphabet.)

And for those adults out there who are snickering in their sleeves, I dare you to try *Kids on Keys*, Game 3, Level 4. Just try to guess all those fragmented shapes, especially after they have changed color and scrambled their positions. Let me tell you, it is no laughing matter. Especially since the key word is quickly fading away.

How well-developed is *your* skill of pattern recognition? Play *Kids on Keys*, and you'll find out.



NEW! For the Commodore 64™ ANNOUNCING

CodePro-64™

A new concept in interactive visual learning . . .

Now you can learn to code in BASIC and develop advanced programming skills with graphics, sprites and music—**visually.** You learn by interacting with CodePro-64, a new concept in interactive visual learning.

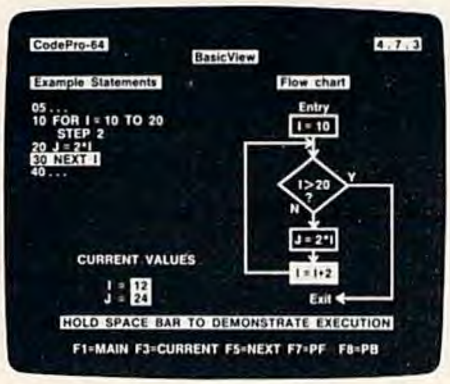
SEE PROGRAM EXECUTION

Imagine actually seeing BASIC statements execute. CodePro-64 guides you through structured examples of BASIC program segments. You enter the requested data or let CodePro-64 do the typing for you. (It will not let you make a mistake.)

After entering an example you invoke our exclusive **BasicView™** which **shows you how the BASIC program example executes.**

You step through and actually see the execution of sample program statements by simply pressing the space bar. CodePro-64 does the rest.

You see statements with corresponding **flow chart graphics** and variable value displays. You learn by visual examples.



EXTENSIVE TUTORIAL

CodePro-64's extensive tutorial guides you through each BASIC command, program statement, and function. You get clear explanations. Then you enter program statements as interactive examples. Where appropriate, you invoke BasicView to see examples execute and watch their flow charts and variables change.

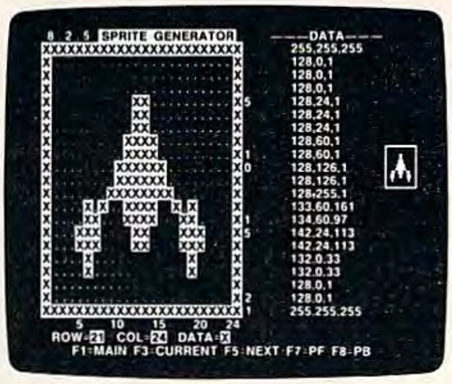
By seeing graphic displays of program segment execution you learn by visual example. You **learn faster and grasp programming concepts easier** with CodePro-64 because you immediately see the results of your input.

You control your learning. You can go through the tutorial sequentially, or return to the main menu and select different topics, or **use keywords** to select language elements to study. You can page back and forth between screens within a topic at the touch of a function key.

CodePro-64 **lets you follow your interests** and practice with interactive examples. But you can never get "lost". F1 will always return you to the main menu. Once you have practiced and mastered the BASIC language elements you move on to more advanced concepts. You learn about sprite and music programming.

SPRITE GENERATOR & DEMONSTRATOR

CodePro-64's sprite generator lets you **define your own sprites** on the screen. You learn how to define sprites and what data values correspond to your sprite definitions. (You can then use these values to write your own programs.) You can **easily experiment** with different definitions and make changes to immediately see the effects.



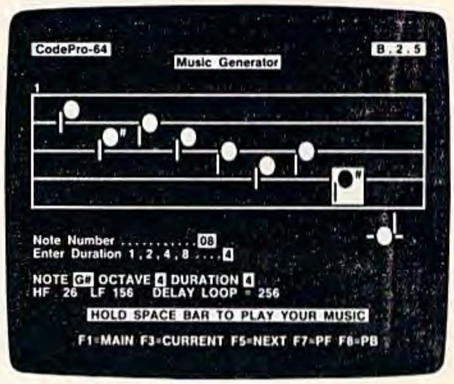
We also help you learn to program with sprites by giving you a **sprite demonstrator** so you can see the effect of changing register values. You can experiment by moving your sprite around in a screen segment, change its color or priority, and see the effects of your changes. You learn by visual examples.

MUSIC GENERATOR & DEMONSTRATOR

To teach you music programming CodePro-64 gives you an interactive music generator and demonstrator. First we help you set all your SID parameters (attack/decay, sustain/release, waveform, etc.). Then you enter notes to play and **we show your tune graphically as it plays,** note by note, on the scale. You learn by seeing and hearing the results of your input.

OUR GUARANTEE

We guarantee your satisfaction. You must be satisfied with CodePro-64 for the Commodore-64. Try it for 10 days and if for any reason you are not satisfied return it to us (undamaged) for a full refund. **No risk.**



Our music demonstrator **lets you experiment** with various combinations of music programming parameters and hear the results. You can **quickly modify any of the SID register values** to hear the effects of the change. For example, you could easily change waveform and attack/decay values while holding all other SID values constant. By seeing your input and hearing the result you quickly learn how to create new musical sounds and special sound effects.

AND MORE . . .

We don't have enough space to tell you everything CodePro-64 offers. You need to see for yourself. BASIC tutorials, graphics, sprites, music, keyboard review, sample programs—the main menu shown above gives you just a summary of the contents of this powerful educational product.

Whether you're a beginning programmer or an experienced professional, CodePro-64 will help you improve your Commodore 64 programming skills. We're sure because CodePro-64 was developed by a team of two professionals with **over 25 years** of software development experience.

CodePro-64 is a professional quality educational program for the serious student of personal computing. And it's **fully guaranteed.** Order yours today.

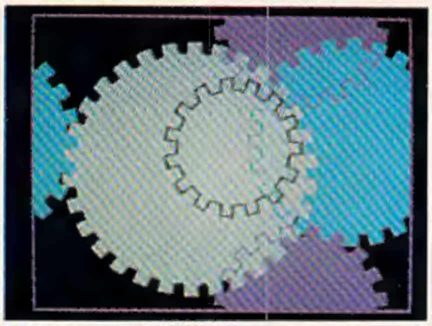
HOW TO ORDER

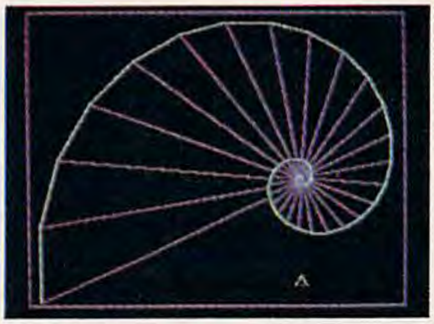
Order your copy of CodePro-64 today by mail or phone. Send only $59.95 plus $3.00 shipping and handling to:

SYSTEMS MANAGEMENT ASSOCIATES
3700 computer Drive, Dept. CA
Raleigh, N. C. 27609

Available on **diskette only.** MasterCard/VISA accepted. For faster service on credit card orders call (919) 787-7703.

Commodore 64 is a trademark of Commodore Business Machines, Inc.
Ad no. 733, Copyright 1983, SMA
Dealer Inquiries invited.





Alphabet Zoo

Alphabet Zoo is also available from Spinnaker Software. The Commodore 64 disk costs $29.95; the Commodore 64 cartridge costs $34.95.

Alphabet Zoo is a trip—a trip into a dark maze filled with colorful glowing letters. At the heart of the maze is a fox or a vase or a bottle of ink or a pair of socks (or dozens of other objects). Your goal is to guess the first letter in the object's name (like "f" for fox or "s" for socks). Then you run through the maze and chase down that letter. The letter skulks around the maze trying to elude you, but you can enter special doorways and take shortcuts through the maze. When you capture the letter, the computer plays a musical tune, you win points, and you get plopped down in a new maze with new letters and a new object.

Alphabet Zoo is very flexible. You and your child can play alone or together. You can choose to chase down capital letters, lowercase letters, or a mixture of both. Also, you can graduate to game 2 where you have to chase down entire words that match the picture in the maze's center.

There are six levels in each game. This lets your child work on different types of letters and words: easy and difficult consonants, vowels, etc., and words of anywhere from two to nine letters.

Alphabet Zoo is a valuable game for you and a child to play. It teaches all sorts of reading readiness skills, including letter recognition, letter sequence, and letter sounds. And having to chase the letters around the maze helps children develop fine motor skills that they will need when they begin writing.

All these things are terrific, but I've saved the best part for last. When you start each new trip into the alphabet maze, you get to choose your own player-creature. And the creatures are hilarious. One is a plump, pumpkin-like happy face. Another is a little, excited monster who keeps jumping up and down.

All the player-creatures are very lovable. Making them hop and bump their way around the maze hunting letters is a big part of the game's charm. And that's the secret of a good game. The game has worthwhile goals and desirable rewards. But it's also fun just playing. You and the child will still enjoy yourselves even if you never do track down one of those tricky letters.

Cosmic Life

Cosmic Life is available from Spinnaker Software. The cartridge for the Commodore 64 costs $34.95.

Cosmic Life originated long, long ago, in the mists of time, before the Apple, before the PET, and before the TRS-80.

In that long-ago time there was a math wizard named John Conway. Conway created a game called *Life*. In Conway's little world, creatures lived according to three very simple rules:

• Survival

Every creature with two or three neighbors was happy and survived until the next generation.

• Death

When a creature was surrounded by four or more neighbors the creature felt overcrowded, became sad, and died. If the creature had only one neighbor or no neighbors at all, the creature became lonely and died.

• Birth

Whenever three creatures got together and shared an empty space, they produced a new creature for the next generation.

Conway published his game of *Life* in *Scientific American* over ten years ago. But it wasn't until recently that Ken Madell, the author of *Cosmic Life*, showed Spinnaker that he could convert Conway's intellectual parlor game into a fun computer learning game for kids and adults.

The creatures in *Cosmic Life* are known as Digi-Bugs, cute little *Pac-Man*-like creatures. They are born, they live, and they die according to Conway's original rules.

When you play *Cosmic Life* you begin with a barren, uninhabited planet. You pilot a joystick-controlled spaceship down to the planet and begin seeding it with Digi-Bugs.

Then prepare to be entranced. Digi-Bug colonies start popping up all over the screen. The little creatures grow, multiply, dwindle, and disappear, right before your eyes.

You can set everything in motion, then retreat to a cloud to watch the action, or you can dive your spaceship back down and continue to seed the planet's surface with new Digi-Bugs.

Pretty soon you will develop a real affection

The Most Practical Software — Now Has Graphics

The **Graphics Assistant,** the latest addition to the ASSISTANT SERIES, lets you and your 64 produce charts and graphs in three formats. You can display them on screen or print them out. On screen display is 30 columns by 14 rows — about 60% of the screen. Print-out can be two sizes: a compact 4" x 4" or a full page, 7" x 9", display.

Bar chart format accepts up to 30 bars per chart; line chart allows 200 points per chart; pie chart can be sliced as thin as you desire. Vertical and horizontal labels are clearly displayed. On the pie chart a label with pointer is displayed outside the graph and indicates percentage or raw numeric data, i.e. Rainbow (73) or Graphics (141). You can assign range, limits, and values to create charts. Most importantly, however, you can retrieve data from files created by the **Spreadsheet Assistant.**

The ASSISTANT SERIES is now better than ever! You can now attach graphs to documents created by the **Writer's Assistant.** And produce comparison charts from data that has been calculated and replicated on the **Spreadsheet Assistant.**

The Graphics Assistant **$79.95**

RAINBOW Computer Corporation

490 Lancaster Pike

for the little creatures. You will learn what patterns help them grow and which patterns make them sad and vanish.

Something happens each Digi-Bug day. Each day lasts about four seconds. You can create a game of anywhere from 10 to 250 Digi-Bug days. At the end of each day the computer scores points based on how many Digi-Bugs are currently living on the planet. Your goal is to create settlement patterns for the Digi-Bugs that make them happy, fruitful, and fertile. But you have to keep a balance. If your Digi-Bug planet gets too crowded, the Digi-Bugs will start disappearing again.

Up For Grabs

Up for Grabs is also available from Spinnaker Software. The program costs $39.95 and comes on a cartridge for the Commodore 64.

Up for Grabs is supposedly for kids eight and up. But it is an instant swamp for kids, and maybe for adults, too. This is not to say that the program is not fun, because it is fun. But *Up for Grabs* is not an intuitively charming game like the other games above. It takes lots of practice and you'd better read the instruction book if you want to know what's going on.

Up for Grabs is an electronic *Scrabble* game. A cube spins around in the center of the screen. On each of the cube's faces is a letter. The letter rotates around, in view, then disappears. When the cube face comes around the next time, a new letter has replaced the old letter.

You pick a letter by pushing the button on your joystick. An arrow appears and points, in turn, at each of the letters on the cube that are visible. When the arrow points at the letter you want, you press the joystick button again.

There are four letter boards for up to four *Up for Grabs* players. Once you have chosen a letter, you can place it on one of the squares on your board by manipulating a row pointer and a column pointer.

When Catie and I first tried playing *Up for Grabs* without reading the directions, we got nowhere.

Later, my wife Janet and I played. Janet spent most of the first couple of games fuming and fussing at the computer. She claimed it was stealing her letters, putting them on the wrong squares on the board, and substituting other letters for the ones she'd chosen.

I had the same problem.

But then things started improving. We got better at manipulating the letters and the game boards. All of a sudden, we were hooked. We played game after game.

We kept playing. I looked at my watch. It was ten o'clock, it was a school night, and the kids were upstairs noisily dismantling their bedrooms. But Janet and I played on.

If you like *Scrabble* and you are a patient learner, you'll like *Up for Grabs*.

Tonight I'm going to talk to Catie. I'm going to try to persuade her to give the game a second chance. I think it's worth it.

Fraction Fever

Fraction Fever is available from Spinnaker Software. It costs $34.95 and comes on a cartridge for the Commodore 64.

This is one of the most frustrating yet most addictive games I have ever played. (Spinnaker recommends *Fraction Fever* for *people* eight and up. *Fraction Fever, Up for Grabs,* and *Cosmic Life* are the first three games in Spinnaker's Family Learning Game series.)

The game is not a swamp, it's just so darned tough!

When you enter the world of this game you become a little person on a pogo stick. You start bouncing the pogo stick around on the bottom floor of a crazy, 20-floor building.

As you bounce the pogo stick, using your joystick, you discover boxes beneath the floor at intervals. The boxes, some filled and some empty, represent fractional quantities. Your goal is to find a group of boxes that matches the fraction hovering in the top-center part of your screen. For example, let's say the fraction is ½. You would bounce your pogo stick until you found, say, four boxes together where two boxes were full and two were empty. When you bounce onto the square with these boxes you press the joystick button.

A neat thing happens. A *fraction elevator* springs out of the floor, picks you and your pogo stick up and carries you to the next floor. You bounce off the elevator and begin hunting boxes to match with a new fraction that is displayed at the top of the screen.

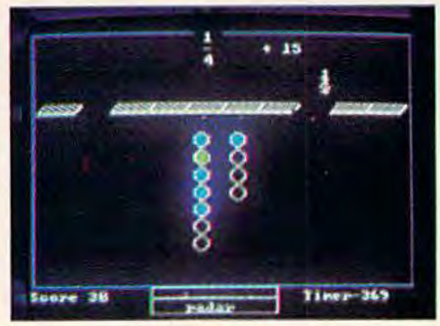
One of the best features of this game is the pogo radar. The little radar screen shows the floor you are on and the floor above and the floor below, each in a different color.

The radar is important because you can use it to estimate where you will find the boxes to match the fraction. The boxes are like distance markers. The fraction they represent is equal to the portion of the whole floor you have traveled, measured from left to right. For example, if you are trying to find boxes representing ¾, you can locate your little pogo-stick person on the radar, then bounce him three-quarters of the way along the floor to the right.

When you find the boxes—four of them, three full; or, perhaps, eight of them, six of them full—you have three visual matches for a particular fraction. First, you have the fraction itself (¾) in

Alphabet Zoo *by Spinnaker Software.*

Kids on Keys *by Spinnaker Software.*



Fraction Fever *by Spinnaker Software.*

the upper part of the screen. Second, you have the four boxes (three full out of a total of four). And, third, you can see the little pogo stick on the radar, and it is exactly three-quarters of the way along the floor (measured from left to right).

The radar is also important because it warns you that holes in the floor are close by. If your pogo-stick person drops through the hole, he falls to the next floor below. This doesn't hurt him, and he can summon the fraction elevator to go back up by matching a new fraction to new boxes. But he can only fall ten times. After that he runs out of pogo sticks.

Where did the holes come from? The only way to get points in this game is by punching holes in the floor with your pogo stick. You get points each time you punch a hole in the floor over a set of boxes that do *not* match the fraction that is appearing on the screen.

But watch out. You have to punch and run, or else you will drop through the hole you just created and fall down to the floor below.

And there's the rub. Those holes are a darned nuisance. The first few times I played *Fraction Fever* I deliberately punched lots of holes to score lots of points. But then my floors had holes everywhere, and I ended up falling down a hole before I could find the correct boxes and catch a ride upward on the fraction elevator.

So I changed my tactics. I tried to get to the topmost (20th) floor first. Then I planned to work my way backwards, punching holes and falling through the floor.

This tactic worked fine until the 16th floor. Then the boxes changed to partly filled beakers. I had to see if the current fraction matched the amount of liquid in the beakers, and then check to see if the partly filled beakers matched the portion of the floor I had traversed. By the time I went through all this estimating and guessing, my time would run out and I would have to hop off the current floor (or fall through a hole) and drop to the floor below. Then the timer would start again and I would try to match the fraction, the beaker, and the floor, and catch another ride upward on the fraction elevator.

Unfortunately, I kept timing out and falling through holes faster than I could estimate fractions. Pretty soon I was back near the bottom of the building with no more pogo sticks to bounce on.

Now I'm a veteran of *Fraction Fever*. Even so, I've never made it past the sixteenth floor, and I've never scored over 16 points.

But I'm going to keep trying. And because I'm persisting, I'm becoming a better fraction-guesser and a better pogo stick bouncer.

I just wish that Tom Snyder, the designer of this game (along with other Spinnaker best sellers, such as *In Search of the Most Amazing Thing* and *Snooper Troops*), would have been more generous with his point allotment. After scoring thousands of points with videogames, I found it quite hard to be content with scores like 6, 11, or 3.

Also, I would have loved it if Snyder had awarded me points for guessing the correct fraction rather than for punching holes in the floor whenever I spotted an incorrect fraction (or group of boxes representing a fraction).

Last, I wish that Snyder had designed the game with several levels, including three or four below the level the game operates at now. I can live with the knowledge that I've only made it to the sixteenth floor (that's $^{16}/_{20}$ of all the floors, or $^{8}/_{10}$, or $^{4}/_{5}$, or four full boxes out of a total of five). But it would have made it easier for me to get Catie and Eric past the first floor.

I've caught a terminal case of fraction fever. Now I'm anxious to pass it on to my kids.

64 Electronic Notepad

Dan Carmichael, Assistant Editor

If you've ever collected so many games and programs on disk or tape that you lost track of how to use each one, then the "64 Electronic Notepad" is just what the doctor ordered.

Have you ever looked at a directory on a disk (or a list of programs on a tape) that was jam-packed with programs and realized that you had forgotten the instructions on how to run them? Or have you ever wanted a convenient way to store anything like notes on that program you're writing or a list of names and phone numbers? If so, then the "64 Electronic Notepad" is the program for you.

The Electronic Notepad contains the most basic functions of a word processor and the ability to store a screen full of information to either tape or disk. Don't leave yet; there's an added bonus: a built-in cipher that will, at the touch of a finger, scramble or unscramble your notes. It's a simple scrambler, but good enough to fool the average nosey person.

How To Use The Program

First, type in the program and SAVE it to tape or disk before running. Be extra careful when typing in the DATA statements; they're for a machine language program, and as with all machine language subroutines, a mistake in just one DATA statement can freeze up your computer.

After the program has been typed in accurately, type RUN, press RETURN, and wait a few seconds while the BASIC program loads the machine language routines into memory.

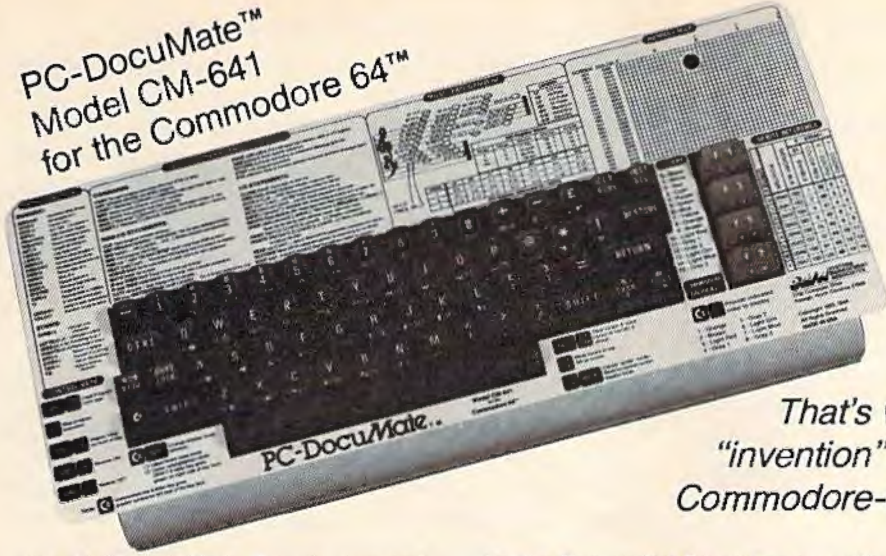
The first user prompt you will see is ENTER 2 SECRET CODES (0–255):. This is for the cipher part of the program. The scrambling of your notepad pages (which, by the way, is optional) is done twice, using two different numbers. Two scrambles will make it that much harder for the curious or the nosey to decode your secret notes. When you enter these two numbers (between 0 and 255), separate them by a comma. Don't be alarmed when you type in the numbers and they aren't seen on the screen. This is intentional. If there are other people around, they won't be able to see your secret codes as you type them in. If you want to see the numbers as you type them, you can delete *POKE 646, PEEK (53281)* from line 17 (be sure to remove the colon, too). Remember to separate the two numbers by a comma. As an example, you might enter *100,200*. If you won't be needing the cipher, enter two zeros.

The second user prompt that will be displayed is DISK OR TAPE?. Here you'll want to press either D or T. The program is written for one or the other, but not both at the same time. For example, if you're in the tape mode, you won't be able to get a disk directory with the program, even if you have a disk drive connected to the computer.

If you've typed in the program correctly, it should now be running, and you should see the options page. The options are:

View notepad page. Press the f1 key for this option. The program calls in a notepad page from either disk or tape and sends it directly to the screen for viewing.

Create notepad page. If you press f3, the program enters the basic word processor mode. Now



EUREKA!

That's what we said when our new "invention" solved all our VIC-20™ and Commodore-64™ programming problems

We had a problem. So we invented PC-DocuMate™ to solve it. The problem was how to quickly master the VIC-20 and CBM-64 keyboards and easily start programming in BASIC on our new personal computers. First we went through the manuals.

INCONVENIENT MANUALS

The user's guide was a nuisance and the programmer's reference manual was just plain inconvenient to use. We found the control key combinations confusing and the introduction to BASIC to be too "basic" for our needs. We needed a simple solution to our documentation problems.

So we decided to surround the keyboard of each PC with the information we wanted. We decided to print whatever we needed on sturdy **plastic templates** which would fit the keyboard of either the VIC-20 or Commodore 64.

SIMPLE SOLUTION

This was the simple solution to our problem. Now we could have the essential information right at our fingertips.

On the left side and top of the templates we put **BASIC** functions, commands, and statements. On the lower left we used **key symbols** to remind us of how to use SHIFT, RUN/STOP, CTRL and the "Commodore" key. Over on the bottom right side we put some additional keys to help remember about CLR/HOME and RESTORE. But we were still a little confused.

STILL CONFUSED

We found we were confused about music programming, color graphics, and sprites. On both the VIC-20 and the CBM-64 templates we carefully organized and summarized the essential reference data for **music** programming and put it across the top—showing notes and the scale. All those values you must POKE and where to POKE them are listed.

Then to clarify **color graphics** we laid out screen memory maps showing character and color addresses in a screen matrix. (We got this idea from the manuals.)

For the VIC-20 we added a complete memory address map for documenting where everything is in an expanded or unexpanded VIC.

For the Commodore 64 we came up with a really clever summary table for showing almost everything you ever need to know for **sprite** graphics.

GETTING EASIER

Now we had organized the most essential information for our VIC and 64 in the most logical way. BASIC, music, color graphics, and sprites all seemed a lot easier. Our initial problem was solved by PC-DocuMate™

But we have a confession to make.

WE CHEATED

We had solved this kind of problem before. In fact, many times before. You see, we at SMA developed the original PC-DocuMate for the IBM PC. We've made templates for IBM BASIC and DOS, for WORDSTAR™, VISICALC™ and other best-selling software packages for the IBM PC.

So we knew we could invent another PC-DocuMate™ to solve our problems with the VIC-20 and Commodore 64. Now our solution can be yours and you can join the thousands of satisfied users of our template products.

Take advantage of our experience and success with PC-DocuMate templates. Get one for your personal computer.

SOME SPECIFICS

Our templates for the VIC and 64 are made from the same high quality **non-glare** plastic as the more expensive IBM PC versions.

The templates are an attractive **gray** color and are imprinted with a special black ink which bonds permanently to the plastic. They are precision **die-cut** to fit your keyboard.

Unlike some other products we've seen in this category, PC-DocuMate templates are professionally and expertly designed. And they are fully guaranteed.

OUR GUARANTEE

We guarantee your satisfaction. **You must be satisfied** with your PC-DocuMate for your VIC-20 or CBM-64. Try it for 10 days and if for any reason you are not satisfied return it to us (undamaged) for a full refund. **No risk.**

SOLVE YOUR PROGRAMMING PROBLEMS WITH PC-DocuMate™

Order your PC-DocuMate today (by phone or mail) and solve your VIC-20 or CBM-64 programming problems. Send only **$12.95** and specify which computer you have. We pay for shipping and handling. Use the coupon below or call **919-787-7703** for faster service.

YES! Please RUSH me ______ VIC-20 templates and/or ______ CBM-64 templates at $12.95 each. I have enclosed $__________ by:

Check ___ Money order ___MC/VISA ___

Name ______________________

Address ______________________

City ______ State ______ Zip ______

Card # ______ Exp ______

Signature ______________________

Foreign orders (except Canada) add $5.00 US

Mail to: **Systems Management Associates**
3700 Computer Drive, Dept. J-1
P.O. Box 20025
Raleigh, North Carolina 27619

Canadians: Please send $18.95 CDN for each template to:

Systems Management Associates
55A Westmore Dr., Dept. J-1
Rexdale, ONTARIO M9V3Y6

VIC-20 and Commodore 64 are trademarks of Commodore Business Machines, Inc.

Dealer inquiries invited.

you can create a notepad page and save it to tape or disk with any filename you choose. While you're in this mode, pressing f1 performs the SAVE, and pressing f8 aborts the page and returns to the options page.

View disk directory. This option (called by pressing f5) displays the disk directory. The program will send it directly to the screen, and it will not affect the BASIC program in memory. While you're in the tape mode, you cannot get a disk directory.

Change program options. Pressing f6 restarts the program. This enables you to change your secret codes if you wish. You can also use this option to change to either disk or tape. Pressing f8 ends the program.

Using The Cipher Option

The scramble option runs all the time and is controlled by the back-arrow key at the upper-left corner of the keyboard. Pressing it once (if you entered your secret codes) will scramble the screen; pressing it again will unscramble.

To use the cipher option to scramble a page before saving, create the page as you normally would using the f3 *create notepad page* option. When you're ready to save the page, press the back-arrow (to scramble the page) and then press f1 to SAVE. The notepad page will then be saved to either tape or disk in the scrambled form.

To use the cipher to unscramble a notepad page, load the notepad page file as you normally would using the f1 *view notepad page* option. After the page has been loaded and is displayed on the screen, press the back-arrow to unscramble it. If you're using the same secret codes you did when you saved the page, the page will now be readable.

How The Program Works

The key to the Electronic Notepad is screen memory page-flipping. This is a technique where you can create one or more extra screens in RAM memory. When you're in the *create notepad page* mode, you're actually writing to the screen (1024–2047); but as soon as you press f1 for the SAVE, you "flip" to another screen memory page that has been set up past the end of the program in BASIC memory. This way you can display the user prompts, such as ENTER NOTEPAD PAGE NAME, without disturbing the page you've written. We'll come back to this later.

For you machine language programmers, the program uses some Kernal routines. The load-a-page option is done with the Kernal LOAD routine ($FFD5), and the save uses the Kernal SAVE routine ($FFD8). The display directory option can be found in the cassette buffer and is a machine language subroutine.

The cipher option can be found in an unused area of the 64's memory between 679 and 752 ($02A7 to $02F0). The subroutine works by looking at screen memory itself and then flipping the bits (with the EOR command) of the characters that are displayed on the screen. It does this alternately (every other byte of screen memory) with the two secret codes you entered. The first secret code is used to cipher all the odd bytes (1,3,5...), and the second code is used for the even numbers (2,4,6,...).

Hints And Tips

Remember your secret codes. If you use a different secret code when you save a notepad page than when you load it back in, the cipher will not correctly unscramble the page.

The word processor portion in the create mode is designed to be a very elementary word processor. You do not have full editing capabilities, and a few keys, like the CRSR left/right and the HOME/CLR, will not work. The inconveniences are minimal if you proofread the text as it is being created. If you make a mistake, use the DELETE key to backspace/erase and then make your corrections. To end a line, press RETURN. Don't use the last three positions (lower-right corner) on the screen. This can cause the screen to scroll, and you may lose the top one or two lines of your text.

Organization of the notepad page filenames can make things easier. For example, when saving to disk, you might want to end each filename with an EN, which stands for Electronic Notepad. That way, when you're looking at the disk directory, you'll automatically know that a filename like SPACE GAME.EN is the electronic notepad page of instructions on how to play "Space Game" on the same disk.

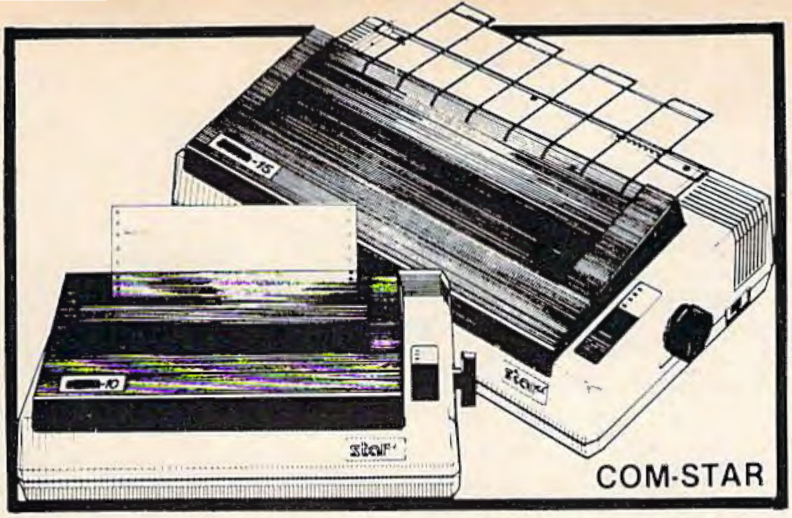
Don't scramble notepad pages unless absolutely necessary. The program was written using page-flipping for a specific reason. If you want to quickly load in a notepad page in the immediate mode, you can do it without running the Electronic Notepad program. First you have to fill color memory, then you can load in the notepad page. To do this, enter:

```
FORA=55296TO56319:POKEA,PEEK(646):NEXT:
  LOAD"filename",dn,1
```

then press RETURN. Filename is the name of the Electronic Notepad page. The device number, dn, is 1 for tape or 8 for disk. If the page was scrambled before it was saved, you'll have to run the Notepad program to unscramble it.

You'll probably find that keeping notes or instructions on the same disks or tapes that contain your programs is a lot easier than shuffling papers and trying to keep track of handwritten notes.

See program listings on page 186.



FANTASTIC PRINTER SALE

as low as **$149.00**

• *15 Day Free Trial - 180 Day Immediate Replacement Warranty*

	LIST	SALE
80 COLUMN THERMAL PRINTER — 60 CPS Bi-directional, dot matrix, prints 8½" letter size paper, full 80 columns, high resolution graphics, dot bit addressable, special symbols and true decenders! (Centronics parallel interface)	$199	**$149**
80 COLUMN TRACTOR-FRICTION PRINTER — 80 CPS Bi-directional, dot matrix, impact, prints single sheets, continuous feed paper, adjustable columns, 40 to 132 characters! Roll paper adapter $32.95. (Serial or Centronics parallel interface)	$399	**$209**
PREMIUM QUALITY 10" CARRIAGE T/F PRINTER — 120 CPS Bi-directional, impact, 9 x 9 dot matrix with double strike for 18 x 18 dot matrix. High resolution bit image (120 x 144 dot matrix) underlining back spacing, left and right margin settings, true lower decenders, with super and sub scripts. Prints standard, italic, block graphics, special characters, plus 24 of user definable characters and much more!! Prints single sheets, continuous feed and roll paper! (Centronics parallel interface)	$499	**$289**
PREMIUM QUALITY 15½" CARRIAGE PRINTER — 120 CPS Has all the features of the Premium Quality 10" Carriage T/F Printer above plus a 15½" carriage and more powerful electronic components to handle large business forms! (Centronics parallel interface)	$599	**$379**
HIGH SPEED PREMIUM QUALITY T/F 10" PRINTER — 160 CPS Save printing time with these plus features: 160 CPS speed, 100% duty cycle, 8K buffer diverse character fonts special symbols and true decenders, vertical and horizontal tabs. This is Red Hot Efficiency!!! (Serial or Centronics parallel interface)	$699	**$499**
HIGH SPEED PREMIUM QUALITY T/F 15½" PRINTER — 160 CPS Has all the features of the 10" Carriage high speed printer plus a 15½" carriage and more powerful electronics to handle larger business forms! (Serial or Centronics parallel interface)	$799	**$599**

PARALLEL PRINTER INTERFACES: (IN STOCK)

- For VIC-20 and COMMODORE 64 — $49.00
- For all APPLE COMPUTERS — $69.00
- For ATARI 400 and 800 COMPUTERS — $79.00

NOTE: *Other printer interfaces are available at computer stores!*

SALE SALE

WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard. We ship C.O.D.

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

Alpha-Shoot

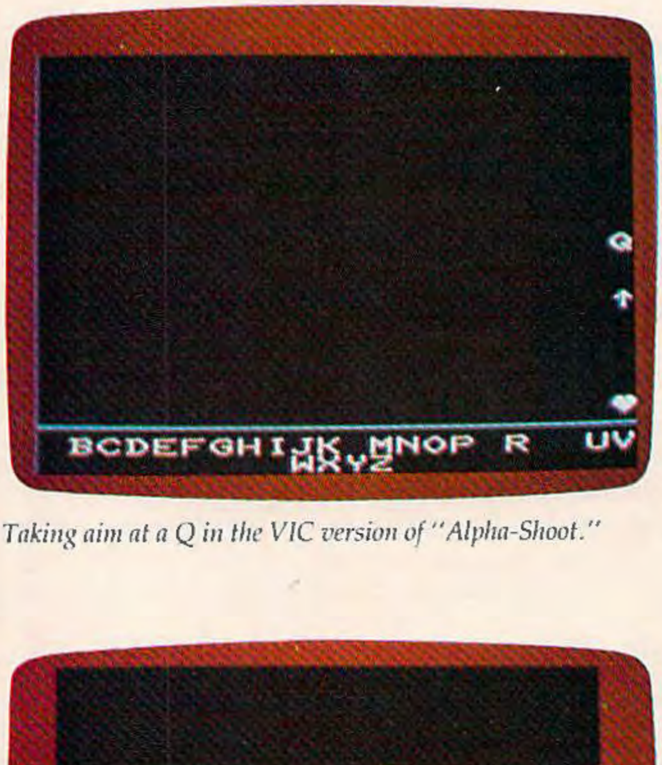
Neil T. Capaldi

The author wrote this educational game for the unexpanded VIC-20; we've added a version for the Commodore 64. It works with either keyboard controls or a joystick.

"Alpha-Shoot" is a game I wrote for my son to help him learn and recognize the letters of the alphabet.

The object of the game is to line up the heart-shaped character at the bottom of the screen with the letter displayed above. The heart can be moved left or right with the C and B keys or with the joystick. (With the Commodore 64 version, plug the joystick into port 2.)

Pressing the space bar or joystick fire button shoots an arrow up the screen. As each letter is hit, it explodes and is placed in alphabetic order at the bottom of the screen. When all the letters in the alphabet have been "captured" this way, the game redisplays the alphabet to the familiar children's tune of "Twinkle Twinkle Little Star."



Taking aim at a Q in the VIC version of "Alpha-Shoot."

Four Games In One

Alpha-Shoot has four possible variations. When you first run the program, it asks you to choose from these four options:

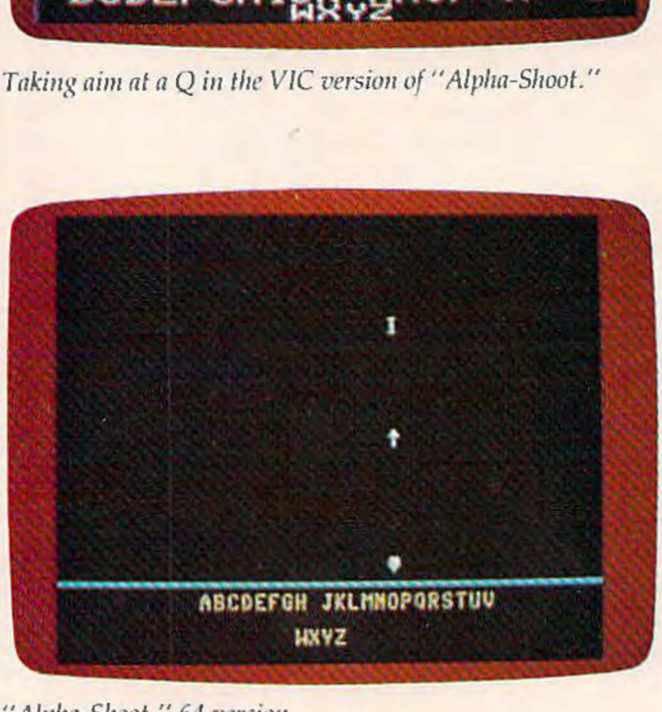
1. The letters of the alphabet are displayed randomly.

2. Letters are displayed in alphabetic order, A-Z.

3. The letter to be displayed can be selected from the keyboard.

4. Letters are displayed randomly and move across the screen.

Parents should select the variation they want and have the child name each letter as it appears on the screen. Also, children can learn alphabetic order by singing along as it is played.



"Alpha-Shoot," 64 version.

See program listings on page 199.

REVIEWS

Children's Educational Games

Tony Roberts, Assistant Managing Editor

ALPHA-BECi uses simple graphics to teach letters of the alphabet.

The goal of Boston Educational Computing is to provide owners of the most elementary computer systems with educational software that can be used easily by those with little knowledge of computing.

In its Child Development Series, BECi (pronounced Becky) meets this goal. Among the software in the series are a counting program and an alphabet program aimed at preschool children and an addition-subtraction program targeted for the slightly older child.

All of the programs are on tape and are designed to work on an unexpanded VIC-20, but they perform just as well with a memory expander. *NUMER-BECi* and *ADD/SUB* are also available for the Commodore 64 and Atari computers.

Introducing The Alphabet

ALPHA-BECi is intended to help preschool children learn the letters of the alphabet, both upper- and lowercase, associate each letter with a word beginning with that letter, and see how each letter relates to the others in the alphabet.

For each letter, the program provides a screen which includes the capital letter, the small letter, and a picture of an object. The screen is slowly drawn, giving a child the opportunity to call out the name of the letter or object as soon as he recognizes it.

First, the capital letter is drawn, then the associated small letter, then the object with its name below. The entire alphabet is then printed at the bottom of the screen with the target letter highlighted.

ALPHA-BECi can be run in one of three ways. From a menu you decide whether to run the program sequentially, randomly, or under keyboard control. That is, the program will either step through the alphabet from A to Z, will display random letters, or will show screens for the letters selected by the user. The mode can be changed at any time by pressing RUN/STOP, then rerunning the program.

For a child, watching the colorful objects appear on screen is like opening a present. The program's only sound effect is a boop-boop-boop that comes as the letters of the alphabet are being printed along the bottom of the screen. For an adult, it may become a little annoying, but it seems to be music to a child's ears.

Shapes And Colors

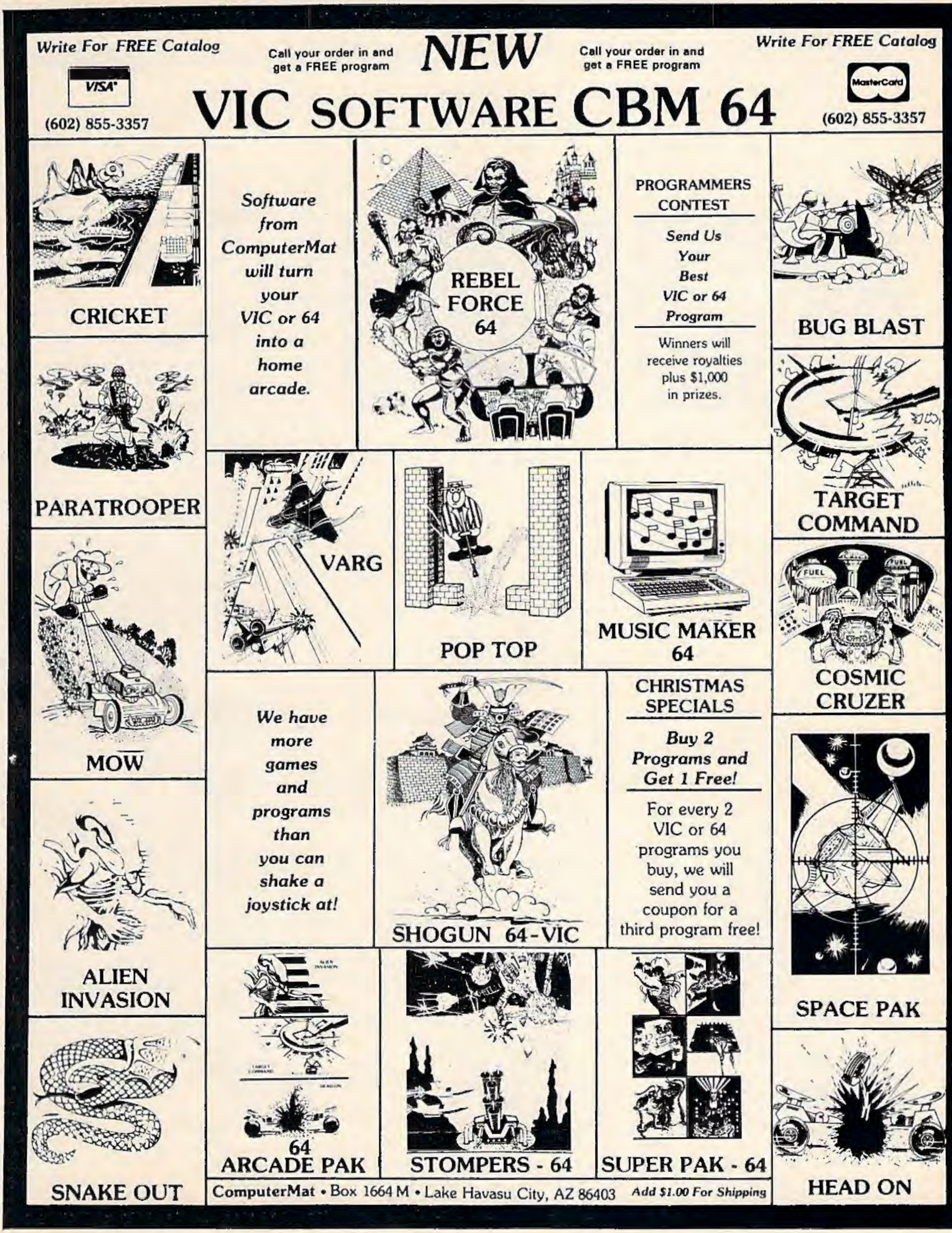
NUMER-BECi teaches the preschooler to identify the numbers from 1 to 12, to identify shapes, and to identify colors. And it is set up so the youngster can learn these new ideas at his own pace.

The menu offers the following options:

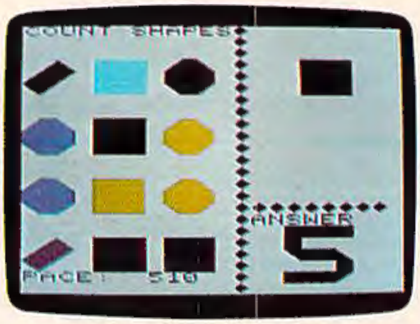
1. COUNT LIKE THINGS
2. COUNT UNLIKE THINGS
3. COUNT COLORS
4. COUNT SHAPES
5. COUNT COLORS AND SHAPES

A second menu allows you to select a time limit for the answers. Your options here are to let the program run itself, filling in the answers after a specific time delay; have it wait until the child fills in an answer; or have it set a time limit.

When running the program with a time limit set, a correct answer will shorten the time allowed for the next problem. As long as the child continues providing correct answers, the time limit is shortened until he misses. Then the time limit is



Write For FREE Catalog
VISA
(602) 855-3357
Call your order in and get a FREE program
NEW
Call your order in and get a FREE program
Write For FREE Catalog
MasterCard
(602) 855-3357
VIC SOFTWARE CBM 64
CRICKET
Software from ComputerMat will turn your VIC or 64 into a home arcade.
REBEL FORCE 64
PROGRAMMERS CONTEST
Send Us Your Best VIC or 64 Program
Winners will receive royalties plus $1,000 in prizes.
BUG BLAST
PARATROOPER
VARG
POP TOP
MUSIC MAKER 64
TARGET COMMAND
MOW
We have more games and programs than you can shake a joystick at!
SHOGUN 64-VIC
CHRISTMAS SPECIALS
Buy 2 Programs and Get 1 Free!
For every 2 VIC or 64 programs you buy, we will send you a coupon for a third program free!
COSMIC CRUZER
ALIEN INVASION
SPACE PAK
64 ARCADE PAK
STOMPERS - 64
SUPER PAK - 64
SNAKE OUT
ComputerMat • Box 1664 M • Lake Havasu City, AZ 86403
Add $1.00 For Shipping
HEAD ON



NUMER-BECi *teaches youngsters to recognize and count various colored shapes.*

The top arrow in ADD/SUB *reminds youngsters when to carry or borrow.*

incremented slightly. This feature challenges a child to increase the speed at which he inputs his answers.

When the program is run in the wait-for-answer mode, the child is simply asked to count the number of shapes printed on the screen. If the answer is correct, he moves on to the next problem. If the answer is wrong, the word WHOOPS! is printed in red and the correct answer is displayed.

Under the Count Colors option, the program prints shapes of different colors on the screen and the child is shown a color block and asked to count the number of shapes that match the color block. Count Shapes is a similar exercise with shapes. Shapes and colors are mixed in the final exercise. The child must count the number of black squares, or red triangles, for example.

The child's answer to each problem is printed in large block letters. It takes a second or two to appear for the answer to be printed on the screen after it has been typed in, and this can be a bit disconcerting if you're used to seeing what you type appear on the screen instantly.

Putting 2 And 2 Together

ADD/SUB is BECi's program for older children. The program, which is available for the Commodore 64 as well as the VIC-20, offers choices of addition or subtraction, one to four digits, and decimal or whole numbers. In addition, the user can decide whether to include problems that require carrying or borrowing. For those just learning about carrying and borrowing, there is an option that provides a hint in the form of a large arrow pointing to the column from which a borrow has been made or to which a carry must go.

With *ADD/SUB* you also can select a pace for solving the problems. The standard pace gives the child about seven seconds to answer. When the time is up, the computer will fill in the answer. If the wait-for-answer option is selected, the computer will wait until the numbers are filled in. The set pace option can be used to speed up or slow down the pace to meet a child's needs.

In *ADD/SUB* the problems are displayed in large black numbers with a green plus or minus sign. In multidigit problems, an arrow points to the first digits to be added or subtracted. The answers are filled in from right-to-left, just as they would be done on paper.

Using The Programs

The programs are, as advertised, easy to use. Simply load them from tape and type RUN. Options for play are presented in simple menus. To change the mode of play, press the STOP key, type RUN, and select a new option.

Loading the programs takes a few minutes; a preschooler with a short attention span might spend less time using the program than it takes to load.

The programs are low key. No scores are kept, and the rewards for correct answers are understated. The adjustable nature of the programs and the pacing options, however, provide these programs with an extended life. They are entertaining to a child just beginning to grasp the concepts covered, and they can challenge the more advanced child to solve the problems more quickly.

ALPHA-BECi
NUMER-BECi
ADD/SUB
Boston Educational Computing Inc.
78 Dartmouth Street
Boston, MA 02116
(617) 536-5116
$16.95 each

COMPUTE!'s Gazette Back Issues

JULY 1983: Commodore 64 Video Update, Snake Escape, Alfabug, VIC Marquee, Word Hunt, VIC Timepiece, product reviews, Learning To Program In BASIC, Quickfind, 64 Paddle Reader, Machine Language For Beginners, Enlivening Programs With Sound, Using Joysticks On The 64, Simple Answers To Common Questions, VICreations — Speedy Variables, 64 Explorer.

AUGUST 1983: Your First Hour With A Computer, Should You Join A Users Group, Guide To Commodore Users Groups, The Viper, Cylon Zap, product reviews, VIC/64 Mailing List, Word Spell, Global Scan For VIC/64, Machine Language For Beginners, VIC Title Screens, 64 Hi-Res Graphics Made Easy, VIC/64 Four-Speed Brake, Disk Menu, Using A 1540 Disk Drive With The 64, Playing Computer Music, Simple Answers To Common Questions, HOTWARE, VICreations — Caring For Disk Drives/Cassettes, 64 Explorer, News & Products.

SEPTEMBER 1983: Telecomputing Today, Telecomputing Glossary, Commodore's Nationwide Party Line, Commodore Bulletin Boards, Demon Star For VIC/64, Potholes, product reviews, Checkbook Reporter, States & Capitals Tutor For VIC/ 64, MiniTerm-20, TeleTerm 64, POKEing Graphics, Machine Language For Beginners, 64 Searcher, Better Commodore Input, Using The Function Keys, Simple Answers To Common Questions, HOTWARE, VICreations — Understanding Random Numbers.

OCTOBER 1983: The Anatomy of Computers, Telegaming Today And Tomorrow, Commodore's Public Domain Programs, Oil Tycoon, Re-Beep, product reviews, Aardvark Attack, Word Match, A SHIFTy Solution: The WAIT Command, Program Transfers, Machine Language For Beginners, Improved Paddle Reader Routine, How To Use Tape And Disk Files, Understanding 64 Sound — Part 1, Speeding Up The VIC, Simple Answers To Common Questions, HOTWARE, Horizons 64 — Improving 64 Video Quality, VICreations — Using The VIC's Clock, News & Products.

NOVEMBER 1983: Binary Numbers — Part 1, Getting Started With A Disk Drive — Part 1, Chicken Little, Martian Prisoner, product reviews, Munchmath, VIC Super Expander Graphics, 64 Aardvark Attack, 64 Timepiece, Connect The Dots, Custom Characters For VIC/64, Making Custom Characters On The 64, Making Custom Characters On The VIC, VIC/64 Program Lifesaver, Understanding 64 Sound — Part 2, Merging Programs On The 64, Tutorial On DATA, READ, RESTORE Statements, One-Touch Commands For The 64, VIC/64 Disk Defaulter, Machine Language For Beginners, Simple Answers To Common Questions, HOTWARE, VICreations — Animation With Custom Characters, Horizons 64 — Software And Hardware Reviews, News & Products, Automatic Proofreader.

Back issues of July, August, and September 1983 are $2.50 each. Issues from October forward are $3. Bulk rates are 6 issues for $15 or 12 issues for $30. All prices include freight in the U.S. Outside the U.S. add $1 per magazine order for surface postage. $4 per magazine for air mail postage. ALL BACK ISSUES ARE SUBJECT TO AVAILABILITY.

In the Continental U.S. call
TOLL FREE 800-334-0868
(in North Carolina call 919-275-9809)

Or write to: **COMPUTE!'s Gazette for Commodore Back Issues**
P.O. Box 5406
Greensboro, North Carolina, 27403, USA

Prepayment required in U.S. funds. MasterCard, VISA, and American Express accepted. North Carolina residents please add 4% sales tax.

Computer Baseball And *Ringside Seat* For Commodore 64

Gregg Keizer, Assistant Book Editor

Ever wanted to manage your own major league team? Most people who follow baseball closely just *know* that they could take their team to the World Series, if only *they* were manager. *Computer Baseball* can give you the feeling of the close game, a pennant race, even the final game of the Series. And it costs far less than buying the Minnesota Twins, the latest team up for sale. You won't even have to pay players' salaries.

This isn't an arcade baseball game like others you may have played. Although players move and hits cross the field, the animation is minimal. *Computer Baseball* is a simulation of the strategy and tactics of a major league team. Using actual statistics, opposing managers can select lineups, choose starting and relief pitchers, set up defense for the bunt or double play, and signal runners to steal. When you put on the manager's cap, *you* make the decisions for your team, hoping to outwit the opposing manager's strategy.

The game includes 28 team rosters, World Series teams all, which you can use to play another opponent, or even the computer manager, Casey. You can even enter data for new teams yourself, or send for a disk containing the most recent American and National League teams, so you can replay an entire season if you want.

How To Play

After you've booted the disk for *Computer Baseball*, you'll be offered several choices. Do you want to play a two-player game, play against the computer, enter new player data, or watch a demonstration game? Playing against Casey, the computer manager, is a good way to learn the game. Choosing this option presents more decisions. Do you want Casey to manage the home team, the visiting team, or both? I let Casey have the visiting team. But which team?

You'll see 28 teams listed on the screen, ranging from the '06 White Sox to the '81 Dodgers. All 28 played in a World Series, so you can replay a complete seven-game Series, or you can play a What If. What if the '27 Yankees, with Babe Ruth, could have met the '81 Yankees? Who was the better team? Has baseball gotten better, or worse? Once you select Casey's team, as well as your own, the computer will display the team lineups.

Now the decisions become more important. You can choose the starting pitcher(s) and set the batting order for your team. If you want, you can set the order for Casey's team too, but I let him do that. Although setting your batting order takes time, it adds to the game's realism. You have to fill each of the fielding positions, and you only have so many players. You could force a catcher to play outfield, but it's probably not a good idea. As you enter the order, take your time, for any errors you make means you'll have to go through it all again. Unfortunately, there's no option to change your mind in this section of the game.

The screen display appears once you've chosen your team and selected the batting order. The display shows a playing field, players, a scoreboard, and pitcher and batter status information. Once the game starts, you'll be able to tell if the batter is right- or left-handed, where the defense is playing, and the base-running abilities of men on base.

Whether your team is on the field or up to bat, you have several decisions to make. I took the field first, being the home team, so I had to choose my defensive alignment. You can play your outfielders shallow, or in their normal positions. Your infielders you can place even more carefully, moving them in, guarding the lines or setting up for the double play. You can hold base runners, if there are any, tight or loosely, depending on the situation. You can even visit the mound and talk to your pitcher, check to see if he's tired, and perhaps bring in a reliever. Make sure that your reliever is warmed up, though, or he could easily be hit off of.

When your team is up to bat, you have fewer choices to make. You can hit away, hit and

run, bunt, steal, or signal your runners to edge off the bag a bit more. If you want, you can send in a pinch runner or hitter.

Each pitcher-batter confrontation is displayed on the screen by a single pitch. There are no balls and strikes called, although batters can be struck out. A message appears at the bottom of the screen after each pitch, telling you what's happened. Pop flies, grounders, and base hits are shown to you this way. Runners advance, are thrown out, or score. Although only nine innings are displayed on the screen, extra innings are possible.

Managing It All

Computer Baseball is not for the impatient. It's not a fast-moving, arcade-style game, but it's not meant to be. Instead, the game is for those who love baseball, who have always wanted to step into a manager's cleats. As a simulation, it gives you the feeling of managing a team. Just as in reality, once the basic decisions are made, the players run, hit, and throw in their own ways. A major league manager cannot hit for a player, and neither can you in this game. The statistics of each player determine that.

Some games are pitchers' battles, with low scores. Other games display hitters' powers, and the scores run up quickly. No two games are the same. Again, that reflects the simulation's excellence. As the innings pass, your decisions on pitching and running change, just as in a real game. Strategy is vital. Outguessing the opposing manager is just as important.

If you enjoy baseball, you'll enjoy this game. The computer takes all the routine drudgery out of keeping track of statistics and lets you concentrate on decision-making. You'll think you've paced and worried in the dugout just like a major league manager.

Ringside Seat

In many ways like *Computer Baseball, Ringside Seat* is a game of strategy and statistics. But instead of managing a major league baseball team, you play the part of a boxer's manager. You're not the fighter, but his manager, telling him how to fight when he's in the corner between rounds. Once the bell rings, he boxes in his own way, simply following your strategic suggestions.

When the game is loaded from disk, you have the option of managing either, both, or neither of the fighters. If you want, you can also act as the third judge in the fight, or let the computer handle it. Then you select the two boxers.

The game includes a variety of boxers available to you. Divided into weight classifications, from bantamweight to heavyweight, you simply enter the fighters' names and weight divisions. To see a complete list of the fighters on the game disk, you should press *E* the first time the computer asks you to enter a choice. After a short delay, you should press *L* to see the fighter lists, and then the weight classification. The screen will then show the fighter's identification number, his name, rating, style (slugger or boxer), and his weight division. The lists contain

The Yankees and Dodgers face off in a demo game of Computer Baseball.

The famous Dempsey-Tunney heavyweight title fight is reenacted with Ringside Seat.

current fighters, as well as famous boxers from the past. If you wanted to see how Tunney would have matched up against Foster, for example, you can create this What If scenario.

Once you've chosen the fighters and started the game, you'll see a display on the screen. The boxing ring is shown, as well as representations of the fighters, each still in his corner. Other information, such as the fighters' names, the strategy picked by each manager, condition of the fighter, from cuts to stamina remaining, and even the cumulative judges' score will be displayed. At the top of the screen you'll see descriptions of the fighters' movements during a round. Near the bottom of the screen you'll see messages displayed for the color commentary and blow-by-blow descriptions of the fight.

As in a real fight, you decide how you want your fighter to box that round. The options range from fighting flatfooted, which lets him rest in a round, saving his strength, to going for the knockout. Your fighter's abilities in each of the strategies are listed at the bottom of the screen before the round starts. Some fighters are better at charging in, while others are more effective in the stick and move. Choosing your fighter's style for that round is the most important part of managing. You have to use your fighter wisely, not overworking him, for each strategy reduces the boxer's stamina level. A more aggressive style, such as charging in, or sticking and moving, uses up more stamina than a defensive posture, like staying away or protecting a cut.

Once the round begins, the fighter is on his own. You'll see a blow-by-blow description of the fight at the bottom of the screen, as well as see the animated figures move in the ring. The only thing you can do once the bell rings is to tell your fighter to cover up. This is helpful if he is getting badly beaten, or if he has a cut opened. Fighters jab, hook, and punch as their statistics allow. Sometimes they'll tie each other up, or back an opponent against the ropes. All you can do is chew at your fingernails, in the true manager tradition.

After each round, the judges' scores are displayed at the top of the screen, showing how each judge awarded points. Each fighter's stamina is changed to reflect how tired he is, and you can choose a different strategy for the next round. The fight continues as many rounds as you selected earlier, from a three-round preliminary bout, to a fifteen-round title fight. When the fight ends, the judges will tabulate their scores and announce a winner, unless a knockout or technical knockout has been called earlier in the fight. Whatever the decision, you can see the judges' scorecards after the fight, seeing how each awarded points and how many knockdowns each fighter had.

As with *Computer Baseball*, you have the option in *Ringside Seat* to enter new data for other fighters, or even to create a fictional boxer, giving him abilities of your own choice.

Fighting It Out

This game is much faster moving than *Computer Baseball*. In a way, that's a disadvantage, for it moves quicker because there are fewer decisions for you to make. I didn't feel as involved in this game as in the baseball simulation for that reason.

The game still gives you the flavor of managing a fighter, and of a bout itself. Pacing is important in the game, for if you expend too much energy early in the fight by constantly choosing to go for the knockout, or to charge in, your fighter will be weak before the fight ends. As the fighter's stamina falls, so does his effectiveness in many of the strategies. When his effectiveness falls below 2, his punches come with less frequency, and land less often.

As in *Computer Baseball*, the animation in *Ringside Seat* is not up to arcade standards. The sound is weak as well. But these detract little from the game's attractions. No arcade-style boxing game gives you the strategic choices and actual fighters that *Ringside Seat* does. Learning how to direct a fighter takes time and practice. This game lets you experience the thrills and agonies of professional boxing, without ever stepping into a gymnasium. Maybe *your* fighter can be a contender.

Computer Baseball
Ringside Seat
Strategic Simulations, Inc.
883 Stierlin Road, Building A-200
Mountain View, CA 94043
$39.95 each

C. REGENA

Built-In Functions

In my previous columns I've tried to show some *fun* things you could do with your computer. This month I'm going to discuss some built-in functions so you can see that your computer can perform technical tasks, and not just games, music, and graphics. Your microcomputer can do many things that the "mainframe" computers can.

The computer can be a very powerful tool in mathematical calculations. A computer can go beyond a calculator by putting procedures into a program for repetitious work or for logic-dependent answers. The slide rule generation was limited to three significant figures, and problems may have taken hours of work and pads of paper to solve. The calculators streamlined problem-solving and took the drudgery out of mathematics. Now the microcomputers can solve problems with even less time and effort—and to many decimal places of accuracy. The VIC-20 and Commodore 64 display numbers with nine significant figures (ten are stored internally).

The π key is handy in any calculations involving pi (the ratio of the circumference to the diameter of a circle). π has a built-in value on the VIC and 64 and is available directly from the keyboard. Use SHIFT and the up-arrow key to get the π symbol. Try the command PRINT π and you will get the decimal equivalent of pi. To use π in any calculations, just use the symbol. For example, try PRINT 3* π and press RETURN. This short program, "Circles," illustrates the use of this key:

Circles

```
100 PRINT "{CLR} ** CIRCLES **"    :rem 162
110 PRINT "{DOWN}ENTER THE RADIUS.{DOWN}"
                                   :rem 218
120 INPUT "R = ";R                 :rem 131
130 PRINT "{DOWN}AREA = ";↑*R*R    :rem 254
140 PRINT "{DOWN}CIRCUMFERENCE = ";↑*2*R
                                   :rem 129
150 PRINT "{3 DOWN}ANOTHER CIRCLE? (Y/N)
    {SPACE}";                      :rem 254
160 GET A$                         :rem 220
170 IF A$="Y" THEN 100             :rem 38
180 IF A$<>"N" THEN 160            :rem 95
190 PRINT A$                       :rem 140
200 END                            :rem 105
```

The square root function is available on our computers. SQR(X) will return the square root of a number with the variable name of X. Try PRINT SQR(16) to get the square root of 16, or 4. Some valid statements are:

```
10 X=SQR(T)                        :rem 145
30 A=SQR(B)+3+Y                    :rem 76
70 Z=SQR(C*3+F)                    :rem 86
40 ON SQR(R) GOTO 80,100,150       :rem 186
```

"Radius" is a sample program that illustrates the use of the square root function to calculate the radius of a circle if the area is given ($A = \pi r^2$).

Radius

```
100 PRINT "{CLR}RADIUS"            :rem 189
110 PRINT "{DOWN}ENTER AREA OF CIRCLE."
                                   :rem 128
120 INPUT "A = ";A                 :rem 97
130 IF A>0 THEN 160                :rem 152
140 PRINT "{DOWN}SORRY, MUST BE > 0."
                                   :rem 174
150 GOTO 110                       :rem 97
160 R=SQR(A/↑)                     :rem 220
170 PRINT "{DOWN}RADIUS = ";R      :rem 12
180 PRINT "{3 DOWN}ANOTHER CIRCLE? (Y/N)
    {SPACE}";                      :rem 1
190 GET A$                         :rem 223
200 IF A$="Y" THEN 100             :rem 32
210 IF A$<>"N" THEN 190            :rem 92
220 PRINT A$                       :rem 134
230 END                            :rem 108
```

ABS(X) is a function that returns the absolute value of a number X. The absolute value of a number is the numeric value without regard for the sign. The absolute value of a negative number is the number without the minus sign. Some valid statements are:

```
10 A=ABS(Y)                        :rem 95
20 IF ABS(SC)=100 THEN 350         :rem 67
30 T=T+ABS(T1-T2)                  :rem 210
```

INT(X) returns the integer value of a number X, or the whole number part of a number which contains a decimal. The integer function truncates the decimal portion of a number, but it does not round the number. The result is always the largest whole number smaller than the given number, or the whole number to the left of a given number on the number line. Thus, INT(4.56) will be 4, but for negative numbers, INT(−4.56) will be −5. Some valid statements are:

```
10 I=INT(X/Y)                          :rem 3
30 J=J+INT(A)                          :rem 220
50 ON INT(S) GOSUB 200,250,270         :rem 43
```

SGN(X) returns the sign of a number X. The value will be 1 for positive numbers, 0 for zero, and −1 for negative numbers. This function is useful in games where the position of an object may be positive, negative, or zero in relationship to another object. The score could also be tested with the SGN function. Valid statements are:

```
10 S=SGN(X-Y)                          :rem 8
20 ON SGN(SC-T) GOTO 150,370,370 :rem 170
40 IF SGN(R)=-1 THEN 430               :rem 223
```

The following program illustrates the absolute value function, integer function, and sign function for several numbers.

```
100 PRINT "{CLR}NUMBER{3 SPACES}ABS
    {2 SPACES}INT{2 SPACES}SGN"    :rem 103
110 FOR I=1 TO 7                    :rem 10
120 READ N                          :rem 253
130 PRINT N;TAB(8);ABS(N);TAB(14);INT(N);
    TAB(19);SGN(N)                  :rem 214
140 NEXT I                          :rem 29
150 DATA 3.4,0,0.6,-2.1,-5,7.2,-5.3
                                    :rem 139
160 END                             :rem 110
```

The VIC-20 and Commodore 64 have several built-in trigonometric functions. Specify a number, numeric variable, or numeric expression within the parentheses (called the "argument" of the function).

SIN(X) returns the sine of an angle specified as X radians.

COS(X) returns the cosine of an angle specified as X radians.

TAN(X) returns the tangent of an angle specified as X radians.

ATN(X) returns the arctangent of a number X. Arctangent is the angle with the tangent of X. The angle will be expressed in radians.

When you are working with angles, remember that the computer uses angles expressed in radians. Since 180 degrees equals pi radians, you can convert D number of degrees to R radians with $R=D^* \pi/180$. The conversion from radians to degrees is $D=R^*180/\pi$.

If you need some of the other trigonometric functions, remember these conversions.

Cotangent (X) = cosine(X)/sine(X) or 1/tangent(X)

Secant(X) = 1/cosine(X)

Cosecant(X) = 1/sine(X)

Some functions are not defined for certain angles (such as the tangent of 90 degrees), and you need to be careful of overflow conditions or division by zero for the reciprocal functions.

The programs following this column illustrate the use of these trigonometric functions. Enter an angle expressed in degrees, D. Line 170 converts the degrees to radians. The sine, cosine, tangent, cotangent, secant, and cosecant of the angle are printed.

Two more technical functions are the exponential and logarithmic functions. EXP(X) returns e to the power of X, where X is a numeric expression that must be less than or equal to 88.02969191. LOG(X) returns the natural logarithm of X, and X must be a number greater than zero. No longer do you need a book of math tables, nor do you need to calculate interpolations—your computer can calculate logarithms and exponentials almost instantly. Sample valid statements are:

```
10 PRINT LOG(X/Y)                   :rem 1
20 A=EXP(B)                         :rem 96
50 G=LOG(H)-LOG(I)                  :rem 13
70 IF EXP(F)>=50 THEN 200           :rem 27
```

If the computer does not have a built-in function that you need, you can define your own function. The definition procedure is useful if you have a long mathematical formula that is used several places in the program. You can save computer memory and typing time by defining the function at the beginning of the program, then every time you need the function, it is called by the function name.

To define a function, use DEF FN with a variable name (one or two letters long) including a variable name within parentheses. For example,

```
10 DEF FNG(X)=3*X*X+4*X+2
```

Here a function G(X) is defined with a formula. Later in the program you can use a statement such as

```
50 PRINT FNG(7)
```

and G(X) will be evaluated with X=7.

The definition statement needs to be executed before the function is used in the program, so it is a good idea to put all definitions at the beginning of the program.

The above example used a function dependent upon a variable X. The defined formula does not have to contain a variable. For example, we could define a function R(Y) as follows.

```
10 DEF FNR(Y)=INT(8*RND(1))+1       :rem 93
```

R(Y) is defined as a random number from 1 to 8.

Commodore 64 and VIC-20

SuperTerm

$149.95

Telecommunications with a difference!

Unexcelled communications power and compatibility, especially for professionals and serious computer users. Look us over; **SuperTerm** isn't just "another" terminal program. Like our famous Terminal-40, **it's the one others will be judged by.**

- **EMULATION**—Most popular terminal protocols: cursor addressing, clear, home, etc.
- **EDITING**—Full-screen editing of Receive Buffer
- **UP/DOWNLOAD FORMATS**—CBM, Xon-Xoff, ACK-NAK, CompuServe, etc.
- **FLEXIBILITY**—Select baud, duplex, parity, stopbits, etc. Even work off-line, then upload to system!
- **DISPLAY MODES**—40 column; 80/132 with side-scrolling
- **FUNCTION KEYS**—8 standard, 52 user-defined
- **BUFFERS**—Receive, Transmit, Program, and Screen
- **PRINTING**—Continuous printing with Smart ASCII interface and parallel printer; buffered printing otherwise
- **DISK SUPPORT**—Directory, Copy, Rename, Scratch

Program options are selected by menus and function keys. For maximum convenience, an EXEC file sets all options on start-up. SuperTerm may be backed-up for safety. Software on disk with special cartridge module.

Write for the full story on SuperTerm; or, if you already want that difference, order today!

Requires: Commodore 64 or VIC-20, disk drive or Datasette, and compatible modem. VIC version requires 16K memory expansion. Please specify VIC or 64 when ordering.

Smart ASCII Plus . . . $59.95

The only interface which supports streaming —sending characters simultaneously to the screen and printer — with SuperTerm.

Also great for use with your own programs or most application programs, i.e., word processors. **Print modes:** CBM Graphics, TRANSLATE, DaisyTRANSLATE, CBM/True ASCII, and PIPELINE.

Complete with printer cable and manual. On disk or cassette.

VIC 20 and Commodore 64 are trademarks of Commodore Electronics, Ltd.

(816) 333-7200

Send for a free brochure.

MAIL ORDER: Add $1.50 shipping and handling ($3.50 for C.O.D.); VISA/Mastercard accepted (card# and exp. date). MO residents add 5.625% sales tax. Foreign orders payable U.S.$, U.S. Bank ONLY; add $5 shp/hndlg.

MIDWEST MICRO inc.

311 WEST 72nd ST. • KANSAS CITY • MO • 64114

You deserve a TOTL business solution.

WORD PROCESSING
TOTL.TEXT

MAILING LIST AND LABELS
TOTL.LABEL

TIME MANAGEMENT
TOTL TIME MANAGER

KEYWORD CROSS REFERENCE
RESEARCH ASSISTANT

For Commodore 64™ and VIC 20™

Announcing the newest members of the family...

BUSINESS ACCOUNTING
TOTL.BUSINESS FOR **$95** (SUG. RETAIL)

SPELLING CHECKER
TOTL.SPELLER (64 only) FOR **$35** (SUG. RETAIL)

DATABASE MANAGEMENT
TOTL.INFOBASE FOR **$50** (SUG. RETAIL)

TOTL SOFTWARE, INC.
quality you can afford

Ask your dealer about TOTL Software or send in the coupon for further details and ordering information.

1555 Third Avenue, Walnut Creek, CA 94596
PLEASE SEND ME MORE INFORMATION ON TOTL SOFTWARE

Name: ______________________

Address: ______________________

______________________ Zip ______

Now, within the program every time we need a random number from 1 to 8, we can use R(Y):

```
50 C=R(Y)                              :rem 225
80 A=R(Y)+R(Y)                           :rem 9
90 IF R(Y)>4 THEN X=2                   :rem 92
```

A defined function may combine other functions, such as

```
10 DEF FNF(X)=1-SIN(X)                 :rem 225
```

You can see that by using user-defined functions you can really customize your programs and make technical calculations less complicated.

I hope this discussion of the built-in numeric functions and the user-defined functions has shown you the powerful potential of your "home" computer. Technical applications which used to be possible only with large computers or with hours of calculation time are now possible with a combination of built-in functions on our home computers. In a later column I'll discuss the versatility of the string functions available on the VIC and 64.

Angles (VIC Version)

```
100 PRINT "{CLR}** ANGLES **"           :rem 87
110 PRINT "{DOWN}ENTER ANGLE IN DEGREES"
                                       :rem 239
120 INPUT "D = ";D                     :rem 103
130 IF D>=Ø THEN 16Ø                   :rem 216
140 PRINT "SORRY, Ø<D<36Ø"             :rem 182
15Ø GOTO 11Ø                            :rem 97
16Ø IF D>36Ø THEN 14Ø                    :rem 5
17Ø R=D*↑/18Ø                           :rem 92
18Ø PRINT "EQUIVALENT RADIANS = ",R
                                        :rem 37
19Ø PRINT "SINE OF ANGLE =",SIN(R)
                                       :rem 14Ø
2ØØ PRINT "COSINE OF ANGLE =",COS(R)
                                        :rem 17
21Ø IF D=9Ø THEN T$="INFINITY":GOTO 24Ø
                                       :rem 167
22Ø IF D=27Ø THEN T$="INFINITY":GOTO 24Ø
                                       :rem 216
23Ø T$=STR$(TAN(R))                     :rem 62
24Ø PRINT "TANGENT OF ANGLE = ",T$:rem 85
25Ø IF T$="INFINITY" THEN C$="Ø":GOTO 28Ø
                                       :rem 221
26Ø IF TAN(R)=Ø THEN C$="INFINITY":GOTO 2
    8Ø                                 :rem 168
27Ø C$=STR$(1/TAN(R))                  :rem 145
28Ø PRINT "COTANGENT OF ANGLE = ",C$
                                       :rem 218
29Ø IF COS(R)=Ø THEN S$="INFINITY":GOTO 3
    1Ø                                 :rem 183
3ØØ S$=STR$(1/COS(R))                  :rem 157
31Ø PRINT "SECANT OF ANGLE = ",S$:rem 255
32Ø IF D=Ø OR D=18Ø OR D=36Ø THEN CS$="IN
    FINITY":GOTO 34Ø                    :rem 41
33Ø CS$=STR$(1/SIN(R))                 :rem 232
34Ø PRINT "COSECANT OF ANGLE = ",CS$
                                       :rem 215
35Ø PRINT "{DOWN}ANOTHER ANGLE? (Y/N)"
                                        :rem 88
36Ø GET A$                             :rem 222
37Ø IF A$="Y" THEN 1ØØ                  :rem 4Ø
38Ø IF A$<>"N" THEN 36Ø                 :rem 99
39Ø PRINT A$                           :rem 142
4ØØ END                                :rem 1Ø7
```

Angles (64 Version)

```
1ØØ PRINT "{CLR}** ANGLES **"           :rem 87
11Ø PRINT "{DOWN}ENTER ANGLE IN DEGREES"
                                       :rem 239
12Ø INPUT "D = ";D                     :rem 1Ø3
13Ø IF D>=Ø THEN 16Ø                   :rem 216
14Ø PRINT "SORRY, Ø<D<36Ø"             :rem 182
15Ø GOTO 11Ø                            :rem 97
16Ø IF D>36Ø THEN 14Ø                    :rem 5
17Ø R=D*↑/18Ø                           :rem 92
18Ø PRINT "{DOWN}EQUIVALENT RADIANS = ";R
                                        :rem 69
19Ø PRINT "{DOWN}SINE OF ANGLE =";SIN(R)
                                       :rem 172
2ØØ PRINT "{DOWN}COSINE OF ANGLE =";COS(R
    )                                   :rem 49
21Ø IF D=9Ø THEN T$="INFINITY":GOTO 24Ø
                                       :rem 167
22Ø IF D=27Ø THEN T$="INFINITY":GOTO 24Ø
                                       :rem 216
23Ø T$=STR$(TAN(R))                     :rem 62
24Ø PRINT "{DOWN}TANGENT OF ANGLE = ";T$
                                       :rem 117
25Ø IF T$="INFINITY" THEN C$="Ø":GOTO 28Ø
                                       :rem 221
26Ø IF TAN(R)=Ø THEN C$="INFINITY":GOTO 2
    8Ø                                 :rem 168
27Ø C$=STR$(1/TAN(R))                  :rem 145
28Ø PRINT "{DOWN}COTANGENT OF ANGLE = ";C
    $                                  :rem 25Ø
29Ø IF COS(R)=Ø THEN S$="INFINITY":GOTO 3
    1Ø                                 :rem 183
3ØØ S$=STR$(1/COS(R))                  :rem 157
31Ø PRINT "{DOWN}SECANT OF ANGLE = ";S$
                                        :rem 31
32Ø IF D=Ø OR D=18Ø OR D=36Ø THEN CS$="IN
    FINITY":GOTO 34Ø                    :rem 41
33Ø CS$=STR$(1/SIN(R))                 :rem 232
34Ø PRINT "{DOWN}COSECANT OF ANGLE = ";CS
    $                                  :rem 247
35Ø PRINT "{3 DOWN}ANOTHER ANGLE? (Y/N)"
                                       :rem 122
36Ø GET A$                             :rem 222
37Ø IF A$="Y" THEN 1ØØ                  :rem 4Ø
38Ø IF A$<>"N" THEN 36Ø                 :rem 99
39Ø PRINT A$                           :rem 142
4ØØ END                                :rem 1Ø7
```

HORIZONS: 64

CHARLES BRANNON
PROGRAM EDITOR

Mailbag

Time to bounce back some of the ideas and suggestions you've sent me. In talking about Commodore 64 video, I mentioned that the new 8-pin 64s have improved video clarity, but said that there are no functions assigned to the additional pins (previous 64s had five-pin plugs for audio/video).

J. Robinson of Santa Monica, California, differs. He's rigged up a cable using an 8-pin DIN plug (sorry, Radio Shack doesn't carry them) with the chroma (color) signal coming from pin 8. Normally, chroma is slightly distorted by the luminance mixed with it, but the signal from pin 8 is pure, and the picture is better than ever, as evidenced by some color slides Mr. Robinson sent me. If you can find an 8-pin DIN plug, try it out. By the way, DIN stands for Deutsche Industre Norm (German Industrial Standard).

We've received some letters about the Automatic Proofreader, asking how the checksum is computed on each line. We send program listings to a disk, then run our Lister program, which reads the programs straight from the disk and formats the listings with cursor controls and graphics spelled out.

In addition, the Lister automatically generates the ":rem" checksums by adding together the ASCII values of all the characters in the line. The reason that the numbers are never larger than 255 is that the addition is done internally in only 8 bits, so it will wrap around from 255 to zero (like an odometer past 99999) if the sum is too large. That's why some numbers for long lines are smaller than other numbers for short lines. It all depends where the number wraps around.

Printer Interfaces

A few issues back, columnist Larry Isaacs talked about a parallel printer interface by CardCo called Card/Print (also written as "Card/?"). We've used this interface here and have been generally pleased with its performance, although it will not translate certain 64 control codes in the listing mode. Another interface I've been using is the Tymac Connection. Unlike the Card/Print, the Connection is set up specifically for your printer (ROM chips are available for several printers). The Card/Print will work with almost any Centronics parallel printer, but the Connection uses the graphics capabilities of your Epson, Prowriter, or other dot-matrix graphics printer.

The Connection offers "almost total emulation" of the VIC printers, such as the 1525. In the emulation mode, it will respond identically to the control codes the VIC printer uses, such as dot graphics, elongated/normal text, cursor-up mode, cursor-down mode, and more. You can therefore use all the special features of your printer (high-quality print, italics, double-strike, etc.) and still be able to run programs specific to the VIC printers, such as high-resolution screen dumps.

In addition, the Connection uses your printer's dot graphics capabilities to actually print the built-in graphics characters on the keyboard. Program listings with graphics symbols will appear as they do on the screen. Unfortunately, the graphics characters are formed rather strangely. Characters which should connect, such as Commodore Q and SHIFT-asterisk, do not. The cursor symbols are hard to read. The Connection also cannot reproduce reverse-video text, since your printer's normal character set is used for alphanumerics. But support for graphics is better than no graphics symbols at all.

The Connection has a listing mode, though it cannot interpret some characters. There is also a "transparent" mode, where it just sends the characters along without any interpretation. This is what you would do with some word processing programs. Unlike the Card/Print, there is no way to "lock" in any one mode.

Hardware-wise, the Connection has a 6502 microprocessor with RAM, ROM, and a printer port. How strange to buy another computer for your 64. It has enough RAM to serve as a 2K printer buffer. With a buffer, characters coming from the computer are stored until the printer can "catch up." If you sent something less than 2K long, it would be instantly printed from the computer's

COMPUTE!'s First Book Of Commodore 64
$12.95

Contents

The Commodore 64 is a powerful home computer. Its state-of-the-art features include sprite graphics and an advanced musical synthesizer chip. *COMPUTE!'s First Book of Commodore 64* is the definitive resource guide for owners and users of the 64. It includes some of the best articles and programs from *COMPUTE!* Magazine and *COMPUTE!'s GAZETTE*, plus many more that have never before appeared in print.

First Book Of Commmodore 64 contains something for everyone: BASIC programming techniques; data storage programs; examples of sprite graphics and musical synthesis; information about writing games, using peripherals, and working with many utilities (including a machine language assembler and a monitor); a complete memory map; and advanced programming techniques. Also includes many ready-to-type-in programs and games.

If you are a Commodore 64 owner who wants to learn more about using your computer to its full potential, this is the book for you.

Now Available From COMPUTE! Books

- *COMPUTE!'s First Book Of 64 Sound & Graphics* $12.95
Clear explanations to help you use all the 64's powerful sound and video features. Plus great programs for music synthesis, hi-res art, and sprite and character design.

- *COMPUTE!'s First Book Of Commodore 64 Games* $12.95
Packed full of games: Snake Escape, Oil Tycoon, Laser Gunner, Zuider Zee. Arcade-action machine language games for fast hands; strategy games for sharp minds.

- *COMPUTE!'s Reference Guide To 64 Graphics* $12.95
A complete, step-by-step tutorial for programming graphics. You'll like the clear writing, the example programs, and the full-featured sprite, character, and screen editors.

- *Creating Arcade Games On The 64* $12.95
The principles and techniques of fast-action game design, including custom characters, movement, animation, joysticks, sprites, and sound. With complete example game programs.

- *Commodore 64 Games For Kids* $12.95
Dozens of games for kids of all ages, making this an instant library of educational software. Learning, creativity, and excitement.

How To Order

COMPUTE! Books are available at bookstores, newsstands, and computer stores nationwide. If your local store has run out of **COMPUTE! Books** or does not carry them, you can order them directly:

By Mail. Send a check or money order in U.S. funds (no cash please) to: Order Department, COMPUTE! Books, P.O. Box 5406, Greensboro, NC 27403. Please include $2 shipping/handling for each book you order. All orders must be prepaid. (C.O.D. orders are not accepted.) Allow 4–6 weeks for delivery.

By Phone. If you have VISA, MasterCard, or American Express, call our **toll-free number 800-334-0868** (919-275-9809 in NC) between 8:30 AM and 4:30 PM, EST. Your order will be processed promptly; we will bill your credit card (plus $2 shipping/handling). Allow 4–6 weeks for delivery.

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

point of view. The buffer would meanwhile be feeding characters to the printer at the printer's speed. My printer has a 2K buffer, so the combination of the printer and the Connection gives me an effective 4K of buffer space.

The Connection (Tymac)
Distributed by Microware
1342 B Rt. 23
Butler, NJ 07405
$119

Strange Lock-Up Bug

It may have happened to you. It's extremely frustrating, and totally unexpected. Fortunately, it's predictable, hence preventable. Go down to the bottom of the screen (the very bottom, the last line, scroll if you have to). Now start typing. Anything. You could just type a bunch of X's. Keep typing as you wrap around the right margin on the first line, then stop right after you type past the right margin of the second 40-column line (the screen will scroll). Now press the INST/DEL key (unSHIFTed) to erase the last character on the previous line.

Whoa! Suddenly, the command LOAD is printed, you get a ?SYNTAX ERROR, the word RUN appears, and if you had a program in memory, it starts. Who typed SHIFT-RUN/STOP?

This is not a trivial error. If you had a program running, you can't stop it. If not, the cursor appears to be flashing merrily, and all seems well. Try typing. Worry begins to creep into your mind as no characters appear. You reach for the panacea of RUN/STOP-RESTORE. Panic sets in when this does nothing, and you press it over and over again, pounding and smashing at the keyboard in a frantic attempt to regain control. Too bad. You have to reach for the power switch and turn your computer off. RAM is wiped clean. Your program, if any, is gone.

I don't know what causes this error. One theory is that when the 64 tries to scroll color memory, it reaches one line too many past $DBE7 (end of color memory) and mangles the registers of the CIA chip, which controls all interrupts. There is no way around it, other than the emergency reset I covered last month. Just keep it in mind.

Hope you enjoy the word processor in this issue. Since it is a complex software product, we'll use this column in the future as a forum for answering your questions about it, as well as tips for using it.

I'll leave you with something to play with: extended background color mode. Enter it with POKE 53265,PEEK(53265)OR64 and try typing the letter "A", SHIFT-A, then CTRL-9 (reverse on), and inverse video "A" and SHIFTED-A again. See what conclusions you come up with, and try changing memory locations 53282 and 53283.

DAN CARMICHAEL, ASSISTANT EDITOR

Using The Dynamic Keyboard

The *dynamic keyboard* is a useful programming technique that can be used with both the VIC-20 and the 64. This technique enables you to POKE values into the *keyboard buffer* and "fool" the computer into thinking you typed the command from the keyboard. It can be used to do everything from simply running a program to chain-loading programs from tape or disk.

The Keyboard Buffer

The keyboard buffer is a block of memory ten bytes (characters) long that runs from memory addresses 631 to 640. The buffer is a temporary holding area that is used to store data input from the keyboard. If you could type faster than the VIC could read the keyboard (it does this 60 times a second), you could fill up the keyboard buffer. Obviously you can't type this fast, but there are other times the computer's operating system does use the keyboard buffer to temporarily store data. If a BASIC program is running and is at a stage where it is not ready to accept data input from the keyboard (a FOR/NEXT time-delay loop, for example), the keys that are pressed will be temporarily stored in the buffer until the program is ready to process the data. To see how this happens, enter the following line in the direct mode, press RETURN, and while the FOR/NEXT loop is running type the following ten keys: 1 2 3 4 5 6 7 8 9 0.

```
FOR A=1 TO 10000: NEXT
```

As you can see, while the time-delay loop is running, the keys you pressed are not displayed on the screen. As soon as the loop is finished, the computer reads what is stored in the keyboard buffer and processes the data accordingly and prints it on the screen. Now enter the above FOR/NEXT statement again, and while it is running type the following 12 keys: 1 2 3 4 5 6 7 8 9 0 1 2. When the loop is finished, you will see the same ten keys displayed on the screen as before. But what happened to the two extra keys that were pressed, the 1 and 2? Those keys were lost because the keyboard buffer, which runs from 631 to 640, can hold a maximum of ten characters. Any keys that are pressed after the buffer is filled are lost.

The Other Keyboard Buffer Byte

You may wonder how the operating system knows that there is data in the keyboard buffer waiting to be processed. Memory address 198 tells the computer how many characters are in the keyboard buffer ready to be processed. Each time you pressed a key while the computer was executing the above time-delay loop, memory location 198 was incremented by 1. After the loop finished running, the operating system took a look at location 198 (which had a value of ten) and knew that there were ten characters in the buffer waiting to be processed. Enter the following commands in the immediate mode, then while the FOR/NEXT loop is running, type 1 2 3 4 5 6 7 8 9 0.

```
FOR A=1TO10000:NEXTA:POKE 198,0
```

After the loop is finished, you'll notice that our ten characters were not printed on the screen. The reason is that after we came out of the loop we POKEd 198 with a 0. Even though our ten characters *were* in the cassette buffer, the operating system didn't print them as it did before. When we POKEd a value of zero into 198, we told the computer no characters were waiting in the buffer.

Using The Dynamic Keyboard

The keyboard buffer can be a very useful tool when properly used. For example, did you know that when you "chain-load" programs (the first program automatically loading and running the second, the second loading the third, and so on) the

first program has to be longer than the second? That's because of the variables. The start of BASIC variables always stays just past the end of your BASIC program. When you chain-load programs, any variables that are used in the first program are stored a few bytes past the end of BASIC. If the second program in the chain is longer, it will write over these variables, and all their values will be lost. To illustrate this, enter the following two short programs and save them to tape with the filenames specified below. Be sure to save the second program with the filename "TEST.LOAD.2" right after the first program (use the filename TEST.LOAD.1).

TEST.LOAD.1

```
10 A=10:B=20:C=30                        :rem 120
20 PRINT"{CLR}{DOWN} THIS IS PROGRAM 1"
                                         :rem 244
30 FORT=1TO500:NEXT                      :rem 189
40 LOAD"TEST.LOAD.2"                     :rem 182
```

TEST.LOAD.2

```
10 PRINT" THIS IS A DUMMY LINE"   :rem 251
15 PRINT" THIS IS A DUMMY LINE"     :rem 0
20 PRINT" THIS IS A DUMMY LINE"   :rem 252
25 PRINT" THIS IS A DUMMY LINE"     :rem 1
30 PRINT" THIS IS A DUMMY LINE"   :rem 253
35 PRINT" THIS IS A DUMMY LINE"     :rem 2
40 PRINT" THIS IS A DUMMY LINE"   :rem 254
50 PRINT"{CLR}{DOWN} THIS IS PROGRAM 2"
                                  :rem 248
60 PRINT"A=";A;"B=";B;"C=";C       :rem 41
```

As you can see, the first program sets three variables (A, B, and C), PRINTs a quick message to the screen, then LOADs in the second program, which prints a few messages to the screen (we made it longer here for our demonstration). It then prints the variables A, B, and C that were set during the first program. Note that the printed variable values are zeros, even though we initially set them at 10, 20, and 30, respectively.

What's the answer? The dynamic keyboard, of course. Change the last line (line 40) in the first program to read:

```
40 POKE 631,131 : POKE 198,1
```

Then reSAVE it using the same filename, and RUN it again. For our demonstration, this modification will work only with cassette. Save this modified version to tape, and be sure to save another copy of the second program right after it.

If you changed line 40 and saved both programs correctly, you'll notice that the first program successfully called in and ran the second, even though we removed the LOAD command in line 40.

This is accomplished by the two POKEs we used in line 40. POKE 631,131 places the token for SHIFT-RUN into the keyboard buffer. When you press SHIFT and RUN on the keyboard to LOAD and RUN a tape program, this character (131) is placed in the buffer. Jot this down; you probably won't find it in your *VIC-20 Programmer's Reference Guide*. We POKEd the SHIFT-RUN into location 631, the first byte of the keyboard buffer, because the buffer is of the FIFO (first in—first out) type. That is, when you press a key on the keyboard, the operating system places it in the first byte of the buffer; when ready to be processed, this will be the first character pulled out.

The other POKE we made in line 40 was POKE 198,1. This tells the operating system that there is one character in the buffer waiting to be processed. As soon as the 1 was POKEd into 198, the operating system was fooled into thinking you had typed SHIFT-RUN from the keyboard, and the computer LOADed and ran the next program.

The obvious drawback here is that it will only LOAD and RUN the next program *on tape*. This is because the POKEs we used did not specify a filename. So when using this method of chain-loading, be sure the programs are saved one after the other.

You'll notice that we still have a problem passing variables because of the longer length of the second program. There is another technique which can easily solve that.

Using The Buffer And The Screen

Delete line 40 from the first program and add the following lines:

```
32 PRINT"{CLR}{2 DOWN}LOAD"            :rem 11
40 PRINT"{6 DOWN}5 A=";A;":B=";B;":C=";C;
                                      :rem 113
45 PRINT"RUN"                          :rem 47
50 POKE631,19:FORA=632TO636:POKEA,13:NEXT
   :POKE198,6                          :rem 96
```

As before, SAVE it to tape and reSAVE the second program right after it. Now RUN the first program and see what happens. Be sure to leave the cassette PLAY button pressed down after the first program is loaded.

This time we've successfully passed our variable values from the first program to the second. Here's how we did it.

Line 32 clears the screen (which also moves the cursor to the HOME position). It then moves the cursor down two lines, and prints the word LOAD. Line 40 shows how we pass our variables from one program to another. It prints a line on the screen that looks like (and is) a BASIC statement: the number 5 (the BASIC line number) and the variables A=, B=, and C=. It then prints the current value of these variables, in this case 10, 20, and 30, respectively. Line 45 prints the word RUN and starts our second program.

Line 50 is the key to it all. The first POKE (POKE 631,19) places the value 19 into the first byte of the keyboard buffer. In this case, the ASCII

value of 19 is the control character which stands for HOME CURSOR. The cursor, then, is simply moved to the home position (the upper-left corner) without clearing the screen. The next five POKEs we make to the keyboard buffer are the same (a value of 13), and are done with a FOR/NEXT loop. The ASCII value of 13 is the control character which represents the RETURN key. POKE 198,6 fools the operating system into thinking that six keys have been pressed on the keyboard. The computer starts processing these commands, and away we go. Here's what happens.

The value of 19 moves the cursor to the HOME position, and then the operating system automatically moves it down one more line. The cursor is now sitting on top of the word LOAD we printed on the screen. The next character in the keyboard buffer is a 13, which stands for RETURN. We've fooled the computer into thinking that we typed the word LOAD and then pressed RETURN. The system now loads in the next program on the tape. While the computer is loading the program, the processing of data in the keyboard buffer is temporarily suspended.

After the LOAD has been completed (we now have the second program in BASIC memory), the operating system continues reading data from the keyboard buffer. The next few bytes in the buffer stand for RETURN. The cursor is now sitting on the BASIC line we printed on the screen in the first program. When the computer reads the next value of 13 from the keyboard buffer, it is again fooled into thinking we have just typed in a new BASIC line for the second program. It then enters that line. The cursor has now moved to the next line on the screen, which reads RUN. Reading another 13 from the buffer, it thinks we typed RUN from the keyboard and pressed RETURN, and away it goes with the second program.

Many Different Techniques

The above techniques could have been accomplished a number of different ways using the keyboard buffer and the screen. Just remember one important fact. In most cases, when the computer reads data from the buffer, it thinks it was entered directly from the keyboard. Almost anything you can do from the keyboard can be done from the keyboard buffer. As a matter of fact, you could probably create a computer program itself by POKEing various values into the buffer.

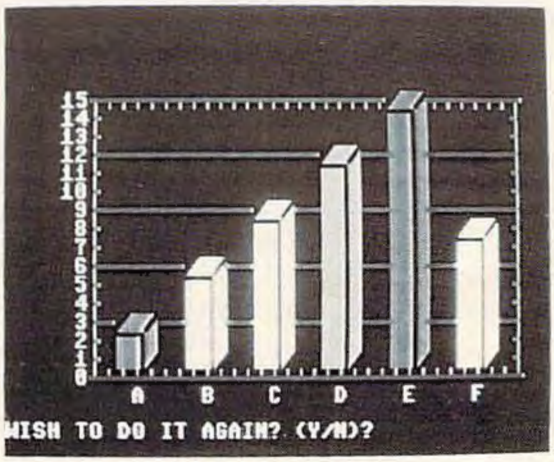
Experiment with this technique and I'm sure you'll find a useful application for it.

If you've got questions or ideas about subjects you'd like to see covered in this column, write to: VICreations, c/o COMPUTE!'s GAZETTE, P.O. Box 5406, Greensboro, NC 27403.

Graph Plotter

Ruth A. Hicks

Not only is "Graph Plotter" an interesting tool for drawing 3-D columnar charts, but the accompanying article takes you step by step through the program itself so you can learn how it was written. The program originally was designed for the Commodore 64, and we've added a version for the unexpanded VIC-20.



3-D bar graphs are a snap with "Graph Plotter" (64 version).

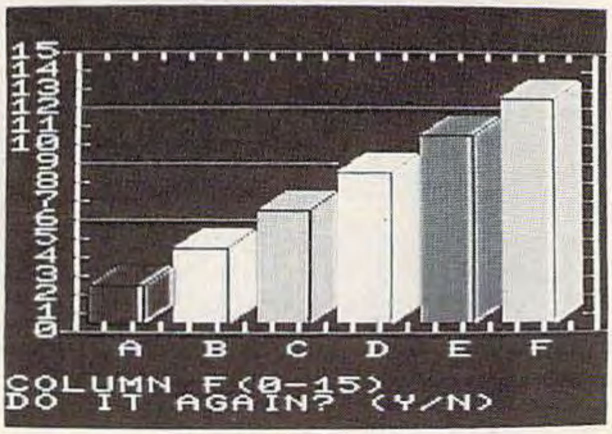
"Graph Plotter" is a good demonstration of what beginning programmers can accomplish in the way of graphics on the Commodore 64 and VIC-20. Different graphics techniques were used to create this program. By reading this article and following along with the program listing, you can increase your knowledge of graphics formatting. Of course, if you're not into learning programming, there's no reason why you can't just type in the program listing anyway.

Graph Plotter creates attractive bar graphs with three-dimensional columns. The graphs are particularly exciting in color. There are six columns, each a different color, to which you assign a value from 0 to 15 for the column height. You tell the computer what values each column has, and then you can interpret their meaning.

Modular Programming

Graph Plotter was written with a technique known as *modular* or *block programming*. This means a section at a time was written on the computer and then checked for eye appeal, function, and (of course) that familiar message, "?SYNTAX ERROR." There are five main blocks to this program.

When typing the program, I suggest that you omit unnecessary spaces except in any INPUT or PRINT statements between quotation marks. All



The VIC-20 version of "Graph Plotter."

other spaces are not needed by the computer and only consume more memory. Since this is an article to learn from, let's start some good habits right away by not typing those useless spaces.

Block One

Block one, lines 100-180, creates the graph, including the segments and the outlining border. Instead of using line after line of PRINT statements, we'll be POKEing the information directly into memory inside FOR/NEXT loops.

Line 100 clears the screen and sets the background color to black and the border blue. Line 110 starts the top border on the Commodore 64 at screen memory location 1230 and runs it across the screen to location 1261, drawing a continuous line (these addresses are 7726 to 7745 on the VIC). Refer to the manual which came with your computer for the "Screen and Color Memory Maps." Each time the FOR/NEXT loop is executed, it places the new value of I into the POKE statement with the symbol number 114 (refer to your manual, "Screen Display Codes"). The I value tells the computer *where* to put the symbol and the 114 tells *what* symbol to put in that spot.

The second POKE in line 110 colors the symbol green. Since the "Color Memory Map" (see manual) corresponds to the screen memory map, only with a different set of numbers, all we have to do is calculate the offset. The difference between 55296 and 1024 (the starting address of color and screen memory in the Commodore 64) is 54272, a simple subtraction problem. So, we POKE I+54272 with the color code for green (5) and presto, we have a green symbol at the correct location! The same thing works on the VIC-20, except the offset between screen and color memory is 30720 instead of 54272.

Line 120 draws the left border, beginning at screen memory location 1270 and ending at location 1790 on the Commodore 64 (7748 to 8034 on the VIC). The STEP 40 is used because a Commodore 64 has 40 characters per line across its screen (STEP 22 for the VIC). If you look at your manual and find screen location 1270, then add 40, you'll find that location 1310 is exactly one line below 1270. On the VIC, 7748+22 adds one screen line. The rest of line 120 and the next two lines are similar to lines 110 and 120, except for different screen symbol codes.

The last four lines (150-180) in this section were constructed in the same manner, using FOR-NEXT loops to POKE information directly into screen memory. These lines draw continuous lines on the graph, making it more readable.

Designations

Block two of the program prints a series of numbers on the left side of the graph and letter designations for each of the six columns. Line 190 positions the following PRINT statement at the right spot horizontally so the numbers can be displayed along the left side of the graph. We want the numbers to start at the fourth space right of the border, so we place a SPC(4) after the PRINT, and then place the number to be printed inside quotation marks.

So, lines 190 through 220 label the Y-axis with a sequence of numbers from 15 to 0. Notice that between each colon is a complete PRINT statement, and even though they are all crunched together in only four program lines with *no spaces*, they result in 16 lines of vertical display. Also, notice that with one-digit numbers the SPC() statement is increased from four to five in the Commodore 64 version for proper placement.

The last line of this section (230) puts letter designations along the bottom of the graph beneath the columns. Notice there is only one PRINT since this line is displayed horizontally. In the Commodore 64 version, the first letter is positioned with TAB(9) and the following letters are all equally spread with SPC(4) statements. Again, because of the VIC's smaller screen size, a TAB(4) and SPC(2) statement are used to position the letters properly.

READ-DATA Block

In the third block of the program (lines 240 to 300), DATA is READ that will be used in a later routine to position each vertical bar on the graph and decide its color. Line 240 prevents this DATA from being reREAD unnecessarily with any subsequent passes through the program.

The first statement that READs DATA in this section is in line 260. Here, a READ command is contained in a FOR-NEXT loop so it is executed six times. This causes six strings, representing the six column labels (A,B,C,D,E,F) to be READ and set equal to the string array variable, A$(I).

In line 280, a second set of DATA is READ and assigned to D(I). This string array variable denotes the color code for each vertical bar on the graph.

The last group of DATA in this block is READ from line 300. The values taken from line 290 are the screen memory addresses necessary to properly locate each bar on the graph.

The use of arrays in this section significantly shortens the length of the program. Instead of requiring six separate blocks of code to locate and draw each vertical bar, we will now be able to perform this in one routine.

Input Block

The fourth block of the program (lines 310-420) is the INPUT routine. Notice that that much of this routine is contained within a FOR/NEXT loop

(lines 310–370).

In this loop, you are asked what value you want for each column. The value that you INPUT determines the height of each vertical bar. Your response is checked in line 360 to make sure it is within the limits of 0 and 15.

After INPUTting the height of each column, the screen memory address (A) for the top of the column is determined by the first statement in line 370.

Here's how it works: A(J) was set as a starting screen location in the first line, then AA (the response) is multiplied by 40, because our screen is 40 characters across (22 for the VIC version). Then AA times 40 (or 22) is subtracted from A(J), because the columns are drawn upward. So, if the response is 10, the column rises 10 segments high. Then 80 (or 44 on the VIC) is added to A to bring it down two rows so we have room for our three-dimensional side. Program execution is then transferred to the subroutine at line 430, which actually draws each column on the graph.

In the process, the variables necessary to this subroutine are passed. The variable C defined in 310 is the offset between the screen memory map and the color memory map as explained above. The actual color of each column (variable D) and the starting screen location of each column, or variable X, are also transferred.

Once a column has been drawn, the user's previous INPUT is erased in line 320 by POKEing blank spaces into this area of screen memory. If you didn't do this, the prior answer, of course, would remain on the screen.

Line 320 enables you to position a PRINT statement exactly where you want vertically without disrupting any printing already on the screen. The cursor is first HOMEd, and then a blank PRINT statement is placed inside a FOR-NEXT loop. As the loop is executed, starting at the HOME position, it counts down vertically to the maximum number set by the FOR-NEXT loop.

The next line is the INPUT statement, now in the right position to be printed. At the end of the INPUT statement is the variable AA, which receives whatever value you enter between the limits of 0 and 15. If the response is less than 0 or greater than 15, the computer erases the answer and asks the same question again.

Once all six vertical bars have been drawn, you will be asked in line 390 if you wish to do another bar graph. If you do, the program will start again at line 100. Otherwise, it will END in line 420.

The Subroutine

The heart of this program is the subroutine beginning at line 430. This is the block which draws the columns by POKEing symbols onto the screen.

Let's start explaining this section with lines 430 and 440. These two lines check to see if the value AA from the INPUT block is a 1 or 0. If AA = 1, the program branches to line 530, which draws the top of a column one segment high on the graph. When AA = 0, it is a null entry, and the program gets another INPUT.

Lines 470 and 480 begin to actually draw the columns, which are three characters wide. Reflecting back to the INPUT block, you'll recall that variables A and X were set for the starting point and top part of the column. So, by POKEing the screen memory locations with the desired character symbols in a FOR-NEXT loop, we can draw the columns to any height we've chosen. Notice there are three POKEs, I, I + 1, I + 2. Each addition to I moves its location over one spot to the right, yielding a three-character-wide column. The different screen display codes create a three-dimensional appearance with reversed characters. The program reverses the character codes by adding 128 to the symbol code (32 + 128 = 160, 101 + 128 = 229). Line 480 follows up line 470 with the color information by adding the color variables C and D to the same locations from line 470.

Lines 510-540 follow the same format as lines 470 and 480. They draw the three-dimensional top segments of the columns. Six character symbols and six color locations are POKEd into the appropriate locations with the variables A, C, and D. By adding or subtracting numbers from A, we can position the symbols on the row above or to the right.

Once this subroutine is completed, line 550 RETURNs to the INPUT block.

Formatting

By now, you should have "Graph Plotter" typed in and SAVEd on tape or disk. The difference between this program and others you have typed is that you now know exactly how it was programmed. Remember the techniques of using PRINT statements for displaying characters vertically and horizontally; of blank PRINT statements and SPC() commands for positioning INPUT or PRINT statements exactly where you want them; of directly placing symbols and colors onto the screen with POKEs and variables. In planning your own programs, use these techniques for your screen displays and see how handy and timesaving they can be for you.

See program listings on page 202.

Happy Holiday Savings

Commodore 64
$219

Avalon Hill Game Company
B-1 Nuclear Bomber (C) .... $ 12
Midway Campaign (C) .... $ 12
North Atlantic Convoy Raider (C) .... $ 12
Nukewar (C) .... $ 12
Planet Miners (C) .... $ 12
Computer Stocks & Bonds (C) .... $ 15
Andromeda Conquest (C) .... $ 14
Computer Football Strategy (C) .... $ 12
Telengard (C) $16

Automated Simulations
Crunch, Crumble, Chomp .... $ 20
Temple of Apshai .... $ 27
Jumpman .... $ 27

Broderbund
David's Midnight Magic .... $ 23
Choplifter (CT) .... $ 34
Serpentine (CT) .... $ 27
Sea Fox (CT) .... $ 27
Bank Street Writer .... $ 45

Creative Software
Home Inventory .... $ 12
Crisis Mountain .... $ 20

Datamost
Swashbuckler (D) .... $ 23
Aztec (D) .... $ 27
Paint Wizard .... $ 34

Human Engineered Software (HES)
6502 Professional Dev. System (C) .... $ 23
Retro Ball (CRT) .... $ 27
Hesmon (CRT) .... $ 27
Turtle Graphics II (CRT) .... $ 45
Heswriter 64 (CRT) .... $ 35
Turtle Tutor .... $ 27
Paint Brush .... $ 20

Infocom
Zork I, II, III .... Ea. $ 27
Deadline (D) .... $ 34
Starcross (D) .... $ 27
Witness .... $ 34
Planetfall .... $ 34

Sierra On-Line
Frogger (D) .... $ 25
Crossfire .... $ 15
Jaw Breaker .... $ 20
Threshold (CRT) .... $ 27
Sammy Light Foot (CRT) .... $ 25

Sirius Software
Blade of Blackpoole (D) .... $ 27
Type Attack (CRT) .... $ 27
Repton (D) .... $ 27
Critical Mass (D) .... $ 27
Snake Byte (D) .... $ 23
Bandits .... $ 27
Squish 'em (CRT) .... $ 23
Final Orbit (CRT) .... $ 23

Spinnaker
Snooper Troops #1 (D) .... $ 30
Facemaker (D) .... $ 23
Kindercomp (D) .... $ 20
Hey Diddle Diddle .... $ 20
In Search of the Most Amazing Thing .... $ 27
Fraction Fever (CRT) .... $ 20
Alphabet Zoo (CRT) .... $ 20
Delta Drawing (CRT) .... $ 20

Synapse Software
Ft. Apocalypse .... (D) $ 23, (C) $ 23
Dreibs .... (D) $ 23, (C) $ 23
Survivor .... (D) $ 23, (C) $ 23
Pharoah's Curse .... (D) $ 23, (C) $ 23

1525 Printer *$229*
1530 Datasette .... *$ 64*
1541 Disk Drive *$249*
1600 Modem .... *$ 89*
1701 Commodore Monitor *$289*
VIC 1311 Joystick .... *$ 8*
VIC 1312 Game Paddles .... *$ 16*

VIC 1210 3K Memory Expander $34
VIC 1110 8K Memory Expander $52
VIC 1111 16K Memory Expander $89
VIC 1011 RS 232 Terminal Interface $43
VIC 1211 Super Expander $59
VT 106A Recreation Pack .... $ 45
VT 107A Home Calculation Pack .... $ 45
VIC 1600 Vicmodem .... $ 89
VM Programmer's Reference Guide .... $ 14
Commodore Programmer's Ref. Guide $ 18

★★ Stocking Stuffers ★★
Koala Touch Tablet (Apple, IBM, Commodore, Atari) $85
Wico "BOSS" Joystick $15
Elephant Disk S/S $18
Flip & File Diskette Box $21

Educational

Books
Kids and the VIC .... $ 18
Programmer's Reference Guide-VIC .... $ 14
Programmer's Reference Guide-64 .... $ 18

Reading & Language Arts
Super Hangman (C) .... $ 14
Simon/Hess (C) .... $ 13
Concentration (C) .... $ 13
Home Babysitter .... $ 23
Word Search .... $ 15
Facemaker .... 64-$ 23 VIC-$ 27
Kindercomp/64 .... $ 20
Snooper Troops I/64 .... $ 27

Mathematics
Sky Math (C) .... $ 12
Space Division .... $ 12
Bingo Speed Math (CT) .... $ 23
Number Crunch (CT) .... $ 23

Number Chaser .... $ 17
Number Gulper .... $ 17

Music
Fun with Music .... $ 27
VIC Music Composer (CT) .... $ 42
HES Synthesound (CT) .... $ 49

Programming Techniques
Intro to Basic Prog. I .... $ 22
Intro to Basic Prog. II .... $ 22
Programmer's Aid Cart .... $ 22
Turtle Graphics/Hess (CT) .... $ 29
Gorteck & The Microchip .... $ 23
Kids on Keys/64 .... $ 23

Social Studies/Science
Visible Solar System .... $ 23
Reagonomics (CT) .... $ 27

Commodore Software
Avenger .... $ 23
Super Alien .... $ 23
Jupiter Lander .... $ 23
Draw Poker .... $ 23
Midnight Drive .... $ 23
Radar Rat Race .... $ 23
Raid on Fort Knox .... $ 23
Sargon II Chess .... $ 29
Cosmic Cruncher .... $ 23
Gorf .... $ 29
Omega Race .... $ 29
Sea Wolf .... $ 23
Adventureland .... $ 29
Pirate Cove .... $ 29
Mission Impossible .... $ 29
The Count .... $ 29
Voodoo Castle .... $ 29
The Sky is Falling .... $ 23
Mole Attack .... $ 23
Bingo Speed Math .... $ 23
Home Babysitter .... $ 23
Visible Solar System .... $ 23
Personal Finance .... $ 29
Quick Brown Fox .... $ 56

1-800-634-6766
Call Toll Free — Order Line Only — Ask for Operator CO64

Information & Order Inquiries
(702) 369-5523
We Accept VISA & MasterCard

COMPUTER OUTLET

1095 E. Twain, Las Vegas, Nevada
(702) 796-0296

VIC 20
$99

Avalon Hill
Tank Arcade .... $ 12
Nuke War .... $ 12

Automated Simulations
Rescue at Rigel (C) .... $ 20
Ricochet (C) .... $ 15
Monster Maze (CT) .... $ 27
Sword of Fargoal .... $ 27

Broderbund
Martian Raider .... $ 15
Multisound Synthesizer .... $ 15
Sky Blazer (CT) .... $ 27
Sea Fox (CT) .... $ 27
A.E. (CT) .... $ 27

Creative Software
Black Hole (CRT) .... $ 36
Astroblitz (CRT) .... $ 36
Apple Panic (CRT) .... $ 36
Serpentine (CRT) .... $ 36
Choplifter (CRT) .... $ 36
Terraguard (CRT) .... $ 36
Household Finance .... $ 20

HES Software
HES Mon (CT) .... $ 29
HES Writer (CT) .... $ 29
Synthesound Music Synthesizer (CT) .... $ 49
Turtle Graphics (CT) .... $ 29
VIC Forth (CT) .... $ 45

Sierra On-Line
Apple Cider Spider (CT) .... $ 27
Sammy Lightfoot (CT) .... $ 27
Frogger .... $ 27

Sirius
Type Attack .... $ 27
Snake Byte .... $ 27

Thorn EMI
River Rescue (CT) .... $ 27
Submarine Commander .... $ 27

Tronix
Galactic Blitz (C) .... $ 17
Swarm (C) .... $ 20
Sidewinder (C) .... $ 20
Gold Fever (CT) .... $ 27
Deadly Skies (CT) .... $ 27

United Microware
Spiders of Mars (CT) .... $ 34
Meteor Run (CT) .... $ 34
Amok .... (C) $ 15 (CT) $ 23
Alien Blitz .... (C) $ 17 (CT) $ 27
Super Hangman (C) .... $ 14
The Alien (C) .... $ 17
Kosmic Kamikaze (C) .... $ 17
Sub Chase (C) .... $ 15
Renaissance .... $ 20
Cloud Burst (CT) .... $ 27
Satellites & Meteorites (CT) .... $ 34
Outworld (CT) .... $ 34
Wordcraft .... $ 65

ORDERING INFORMATION AND TERMS: *For Fast Delivery send cashier checks, money orders or direct bank wire transfers. Personal and company checks allow 3 weeks to clear. Charges for C.O.D. orders are $3.00 minimum or 1% for orders over $300. School purchase orders welcomed. Prices reflect a cash discount only and are subject to change without notice. Please enclose your phone number with any orders.* ***SHIPPING — Software:*** *$2.50 minimum.* ***SHIPPING — Hardware:*** *(Please call)* ***SHIPPING — Foreign Orders:*** *APO & FPO orders: $10 minimum and 15% of all orders over $100. Nevada residents add 5 3/4% sales tax. All goods are new and include factory warranty. Due to our low prices, all sales are final. All returns must be accompanied by a return authorization number. Call 702-369-5523 to obtain one before returning goods for replacement.*

RICHARD MANSFIELD, SENIOR EDITOR

Addressing

You'll soon notice that most of your ML programming involves sending bytes around in the computer's memory. It's quite similar to PEEKing and POKEing, but you've got more options on how you *address* these bytes before you send them somewhere. Addressing bytes is like addressing a letter—you want it to get to its destination so you must write the destination on the letter. There are even ways to send the byte c/o another address, but we'll get to that in a minute. First, we need to review our all-ML game in terms of some addressing options we can use when writing an ML program.

A Cumbersome List

So far, our ML game can be divided into three sections (like subroutines). Let's use the 64 version, Program 2, as our example this month; it's essentially the same as the VIC version. The first section (from address 49152 through 49169) puts the number 8 into all the addresses of Color RAM memory. We usually have a choice of which ML *addressing mode* we want to use. We could have used the simplest mode, absolute addressing, and just listed every address we wanted to POKE the 8 into. It would have looked like this:

```
STA 55296
STA 55297
STA 55298
STA 55299 and so on
```

but that's pretty inefficient. We would have had to list a thousand addresses. Instead, we chose to do our POKEing within a loop. The Y register is quite useful for addressing things because it can be used as an *offset*. That is, you can address something so that the actual address you give is *added to whatever Y equals at the time*. This is a special form of absolute addressing called *absolute indexed* (you can use the X register this way too).

How does it work? First we set Y to equal zero (at 49152). Then we load the A register with our color value, 8. Then we have four STAs lined up, using the absolute indexed addressing mode. The first time the computer comes across this list, it will put 8 into 55296, 55552, 55808, and 56064. It will add Y to these addresses, but Y equals 0 this first pass through the loop. Then, we INY (raise it by one). The three registers (A,X, and Y) can only count up to 255; after that, they reset themselves to 0. So, when Y=1 after the INY, the BNE instruction will "fail" and we'll branch back to 49156 for the second pass through the loop. We can't get past BNE until Y resets itself to zero—BNE means Branch Not Equal (to 0). And Y isn't yet reset to 0.

But, notice what happens the second time through the loop. Since Y now equals 1, we'll be putting our 8 into 55297, 55553, and so on. This storage of 8's continues until all the locations between 55296 and 56319 have been filled.

The Most Common ML Bug

The second section of our game (lines 49171 through 49184) is quite similar and makes use of the same addressing mode. But here we're drawing a horizontal line across the top and bottom of the screen. So, since the screen is only 40 bytes wide, we'll have to test Y (line 49182) to see if it's equal to 40. If not, we BNE back and continue the loop. There are several "B" instructions; all of them begin with the letter B and branch somewhere (if

conditions pass their test). BEQ means Branch if EQual to 0. We'll get to the others in the future. BEQ and BNE, though, are by far the most commonly used ones.

The other instructions here are also the most frequently used ML commands. STA (STore the A register), LDA (LoaD the A register), STY, LDY, CPY (ComPare Y), INY (raise Y register by 1; literally INcrement Y), DEY (reduce Y by 1, literally DEcrement Y), and their companion instructions (DEX, INX, LDX, STX, CPX) all operate according to the same rules (and set up flags for the "B" instructions to test and then decide whether or not to branch).

One other thing to notice here: The computer will always assume that you are loading *from an address* unless you specify otherwise. If you write LDA 15, the computer copies whatever is in address 15 into the A register. (Whatever was at address 15 remains there; only a copy of it is placed into the A register.) So, if you want to actually put the *number 15* itself into the A register, you must put a number sign in front of it: LDA #15. As you can see, we do this frequently in our program. But beware—the single most common source of ML bugs is forgetting to put in that # when you mean a number as such, or putting it in when you mean to get a copy of a number from some memory location in the computer.

Sending Something C/O

Now on to the new portion of our ML game (lines 49186 to the end). Here we are going to draw vertical lines down the sides of the screen, completing the frame for our game. To do this, we'll need to POKE the first address of screen RAM memory, then POKE the 39th (the top of the right-hand margin), then the 40th (the second space down on the left margin), then the 79th, and so on down. The easiest way to do this in ML is to send a character to the right side of the frame, add 1 to Y (INY) and send the character again, this time to the left side of the screen. Then we must add 40 to our address (22 on the VIC) to get over to the right side again.

As before, we'll set up a loop, but we first have to prepare two bytes in zero page (the first 250 memory cells of the computer). These two bytes will hold our addresses in a special way. We'll change the contents of these bytes as we go along, adding 40 (or 22). You can visualize these special bytes in zero page as a Ping-Pong paddle and, by shifting its angle, you can send the balls low or high or anywhere in between.

Working In The Real World

Our 6502 chip can send things to 65536 memory cells, but how can we store a number that large when each cell will only hold numbers up to 255? It's simple enough: We gang two bytes together to hold large numbers. Take a number like 1024, the start of the 64's screen memory RAM. Divide it by 256 and you get 4. So put that in one of the two bytes holding our number (call it the *most significant byte*, or MSB). Then put the remainder of the division (0 in this case) into the other byte, the *least significant byte*, or LSB. Line them up in memory as LSB/MSB (it's backwards to us, but the computer likes them put in this way). There you have it. Notice that the Assembler program performs this whole task for you automatically when you type in a large address (as in line 49156).

How does this work in the real world? We do need to set up just such a double-byte address for our routine which draws vertical lines. We're going to use the two bytes at 71–72 (space that's not being used by the computer during an ML program run). Notice that we must use zero page for setting up our special c/o addressing method. We start off (line 49186) by putting 39 into the LSB, address 71. Then we put a 4 into 72, the MSB. Since the real address (the target) is 256 × LSB + MSB, we'll get 1063 as the target when we land on these two bytes. 1063 is the first space on the right-hand side of our vertical screen line. We can start there because the first line is already filled in anyway with our horizontal line, built earlier in the program.

At this point we can formally introduce one of the most significant and useful of the ML addressing modes—*Indirect Y*. (It's usually called *indirect indexed*. Who can remember that? Let's call it Indirect Y.) It takes a minute to get it straight, but it's a minute well spent. You'll find many uses for this handy method of sending bytes anywhere in the computer. When you address something this way, it "bounces off" the number you prepare in zero page, it's indirect, it's like sending a letter c/o someone. In effect, it gets readdressed once the computer lands down in zero page.

Program 2: 64 Version

```
49152  LDY   # 0
49154  LDA   # 8
49156  STA   55296 ,Y
49159  STA   55552 ,Y
49162  STA   55808 ,Y
49165  STA   56064 ,Y
49168  INY
49169  BNE         49156
49171  LDY   # 0
49173  LDA   # 224
49175  STA   1024 ,Y
49178  STA   1984 ,Y
49181  INY
49182  CPY   # 40
49184  BNE         49175
49186  LDA   # 39
49188  STA   71
49190  LDA   # 4
49192  STA   72
49194  LDX   # 24
49196  LDY   # 0
49198  LDA   # 224
49200  STA ( 71 ),Y
49202  INY
49203  STA ( 71 ),Y
49205  DEX
49206  BEQ         49224
49208  CLC
49209  LDA   71
49211  ADC   # 40
49213  STA   71
49215  LDA   72
49217  ADC   # 0
49219  STA   72
49221  JMP   --->   49196
49224  RTS
```

After we load Y (our offset) with 0 again and load A with the framing character (224), we can store the 224 into the address which the computer finds by looking at the double-byte number we put into addresses 71–72. When it sees STA (71),Y—the computer knows what to do. It first calculates the correct target formed by multiplying whatever it finds in cell 72 × 256 and then adding whatever's in cell 71. Then, it also adds the value of the Y register.

So, we can manipulate the number in Y here the same way that we used it with Absolute Indexed above (line 49156), but have the added advantage of being able to manipulate the double-byte address at 71–72 as well. The first time through this loop, the framing character will be sent to 1063. Then we INY and send another framing character to 1064 (the second space down the left side of the screen). Then we DEX. X is counting down from 24 because there are 24 spaces down each side of the screen that we need to fill. If the DEX causes X to equal 0, then the BEQ takes effect and sends us back to BASIC mode via the RTS (ReTurn from Subroutine) at line 49224. If X is not yet zero (and thus we want to continue the looping), we will add 40 to the double-byte number at 71–72.

The adding is done by first clearing the carry, CLC, and then putting the number from 71 into the A register, adding 40 (ADC means ADd with Carry) and then storing the result back into cell 71. Likewise, we get the number from 72, add it to 0, and put it back. Why add to 0? Because there might be a carry from the operation on the number in 71. If so, we need to reflect that in the overall number by adding it to the MSB (in cell 72). After we've added 40 to this special double-byte number, we just jump (JMP) back to the line where we start our loop that prints the framing characters to the screen.

We've covered a good bit of ground this month. You should try out these routines with your Assembler and run them after they've been placed into memory: SYS 12288 (VIC) or SYS 49152 (64). Then, change some of the numbers and see the effects. Try using a different character for the frame. Pay particular attention to the way that Indirect Y addressing accomplishes its effects—we'll be using it frequently from here on.

See program listings on page 201.

For VIC - 20

16K MEMORY EXPANSION

ONLY $49^{95}

- ★ 14 Day Money Back Guarantee
- ★ Boosts VIC to 21K RAM
- ★ Top Quality, Fully Tested
- ★ 90 Day Warranty

EPYX 16K ADVENTURE GAMES	
Temple of Apshai (T)	27.50
Sword of Fargoal (T)	21.80
MICRODIGITAL ARACADE GAMES	
Snakman (T) (Excellent PacMan Style)	10.75
Gridder (T) (fast chase game)	10.75
Skramble (T) (Thrilling Arcade Action)	10.75

for IMMEDIATE SHIPMENT on Credit Card Orders

Call: (303) 245-9012

10 AM - 9 PM MST Every Day

ASSEMBLY TECHNOLOGY
2692 Hwy 50 Suite 210
Grand Junction, CO 81503

Personal checks allow 3 weeks
Shipping & handling $2.50
Colorado Residents add Sales Tax
COD add $2.00

HUNDREDS OF PROGRAMS AVAILABLE FOR THE COMMODORE 64 & VIC 20

Commodore 64

All Prices up to 35% OFF RETAIL

VIC 20

HOME/BUSINESS	
Practicalc (D)	41.50
Bank St. Writer (D)	52.50
Smart 64 Terminal	31.95
Quick Brown Fox (CT)	42.95
Paper Clip (D)	96.25
WordPro SpellRight (D)	72.95
CalcResult Advanced (D)	135.00
Master Type (D)	29.95
Home Acct (Continental) (D)	49.95
EDUCATIONAL	
Facemaker (D/CT)	22.95
Kindercomp (D/CT)	19.95
Delta Drawing (CT)	26.95
The Most Amazing Thing (D)	26.95
Early Games (D/CASS)	22.50
Electronic Party (VIC 20/CASS)	22.50
Square Pairs (VIC 20/CASS)	22.50
Turtle Trax (VIC 20/CASS)	22.50
Hey Diddle Diddle (D)	19.95
Alphabet Zoo (CT)	22.95
Type Attack (D)	29.95

GAMES	
Zork I, II, or III (D)	29.00
Deadline (D)	37.00
Starcross (D)	29.00
Jumpman (D/CASS)	29.95
Spritemaster (D/CASS)	26.95
Neutral Zone (D/CASS)	27.50
Temple of Apshai (D/CASS)	29.95
HARDWARE AND ACCESSORIES	
Cardco Printer Interface	62.95
Printer Utility (Cardco)	15.95
The Connection Parallel Int.	89.95
Cardco 3 Slot Exp. Board	31.95
Cardco 5 Slot Exp. (C-64)	55.95
Data 20 Video Pak 80 (C-64)	143.95
Data 20 8K Display Mgr. (40/80 col. plus Word Proc.)	119.95
0 K Display Mgr. (40/80)	79.95
Zenith 12" Green Monitor	109.00
Brother/Dynax DX-15 (Letter Quality)	485.00
C. Itoh Prowriter 8510AP	375.00
Control Data Diskettes (10)	23.00

Write or call for FREE CATALOG. TO ORDER: CALL 1-714-643-1056

8:00 A.M.-6:00 P.M. PST Mon.-Sat. or send check or credit card number, signature and expiration date. Please include phone number.

CENTURY MICRO PRODUCTS
P.O. Box 2520, Mission Viejo, CA 92690

Visa/Mastercard add 3%. Personal checks allow 2 weeks to clear. CA residents add sales tax. Shipping and handling add $3.00 (hardware extra). Prices subject to change.

80 COLUMNS! 25 LINES

A FULL PROFESSIONAL DISPLAY FOR

Commodore 64 with Screenmaker™

Screenmaker is a video display generator module that plugs into the expansion connector of the Commodore 64.

- ✓ WORD PROCESSING
- ✓ CALCULATIONS
- ✓ BASIC PROGRAMS

Screenmaker provides a B & W video signal that connects to your video monitor to provide a full 80 characters on each line. With Screenmaker, Screen displays will appear the same as the printer output. Trial printouts can be eliminated. Word processing is easier. Forms and reports can be set up faster. Screenmaker features a bank switched memory, 40/80 video switch, and a full character set including graphics.

SCREENMAKER	$159.95
Copy-Writer Word Processor	$ 79.95
SCREENMAKER/ Copy-Writer Package	$199.95

MICROTECH P.O. Box 102 Langhorne Pa 19047
215-757-0284

TAX AID TAX AID

FOR COMMODORE 64™ and VIC 20™

USE TAXAID™ TO PREPARE YOUR INCOME TAX RETURN

Developed by an experienced accounting firm, TaxAid is accurate, easy to use, and comes with a detailed manual. Your tax data is permanently stored on tape or disk. The cost is tax deductible and yearly updates are available.

TAXAID I	TAXAID II	TAXAID III
For Unexpanded Vic 20	For Vic 20 with 16K	For Commodore 64
$19.95	$24.95	$24.95

Prices above are for cassette; add $5.00 for disk version.

NORTHLAND ACCOUNTING, INC.
606-A Second Ave.
Two Harbors, MN 55616
(218) 834-5012

VISA MasterCard

TAX AID TAX AID TAX AID TAX AID TAX AID TAX AID TAX AID TAX AID

TAXAID IS A TRADEMARK OF NORTHLAND ACCOUNTING, INC.
VIC 20 & COMMODORE 64 ARE TRADEMARKS OF COMMODORE ELECTRONICS, LTD.

64 BASIC Aid

Harold D. Vanderpool

This extremely useful utility program adds four commands to BASIC and belongs in every programmer's toolbox. The utility itself is written in machine language. To type it in, use the MLX entry program found elsewhere in this issue.

No version of BASIC has everything. No matter what computer you look at, there are things that could be added to customize it for your particular needs. The VIC and 64 have an excellent version of BASIC, Microsoft, which has been popular for years on microcomputers. But if you do a bit of programming, you might find that "64 BASIC Aid" will be among the most valuable utility programs in your library—it adds four extremely useful commands to the 64's BASIC.

It uses up very little of your RAM memory (about 1000 bytes) and after you've typed in and saved a copy, you can use 64 BASIC Aid anytime you want those extra four commands. You LOAD it and RUN it as you would any other program, but it hides itself high up in memory and becomes invisible. You can then program as always, but you've got those four extra commands available to you.

Since these commands are useful for writing and debugging programs, they are available to you only in *direct* mode. (You can't include them in a program itself, but you'd never have reason to use them that way.)

Four Programming Aids

NUMBER 100,10. With this command, you can renumber any program that's in your computer. Just type the command and press RETURN. The new version of the program will start with line 100 and go up from there in steps of ten. You can use any numbers you want as the starting line number and any number from 1 to 255 as the step size. This can be useful in several ways. For example, you might have used up all the line numbers somewhere in your program: you've got lines 25, 26, 27, 28, 29, and so forth. No room for new numbers to insert a line? Just use NUMBER and they'll be spread apart instantly for you.

Within your program, there probably will be GOSUBs or GOTOs or other references to existing line numbers. 64 BASIC Aid takes care of that, adjusting the references automatically. However, if it finds a GOTO that's targeted to a line that doesn't exist in the program, it will print the number 65535 on the screen. This is helpful when you debug your programs. Also, all adjusted lines will be printed on the screen.

DELETE 100–200. When you type this, all the lines between 100 and 200 (inclusive) will disappear from your program. It works the same way that the LIST command works, using the same format. But be careful with this one. If you just type DELETE without any line numbers after it and then hit RETURN, it will delete the whole program.

FIND/GOTO/,500–900. This would print a list on screen of each line between 500 and 900 which contained a GOTO command. Again, you can indicate how you want the line numbers handled in the same way, using the same options, as with the ordinary LIST command. If you want a

DI-SECTOR

''To The Rescue''

REPAIR - CUSTOMIZE - LEARN

from

STARPOINT SOFTWARE

AT LAST!! DIRECT INTERACTION with SECTORS

an absolute necessity for disk-based Commodore 64™* users

EXAMINE, MODIFY, and RESTORE by DIRECT SECTOR ACCESS

- ★ simultaneous HEX and ASCII display of sectors
- ★ dual cursor editing
- ★ modify in HEX, ASCII or DECIMAL
- ★ restore scratched/deleted files
- ★ repair lost, damaged, or "destroyed" files
- ★ complete operating instructions
- ★ many useful data recovery and modification tips
- ★ source code fully commented by author

Dealer Inquiries Invited

. . . WRITE OR PHONE . . .

STARPOINT SOFTWARE

Star Route 10 Gazelle, CA 96034 (916) 435-2386

$39.95 on 5¼" diskette

VISA/MASTERCARD accepted

C.O.D. add $2

Program by Bruce Q. Hammond

Ad Art by Steven M. Hettema

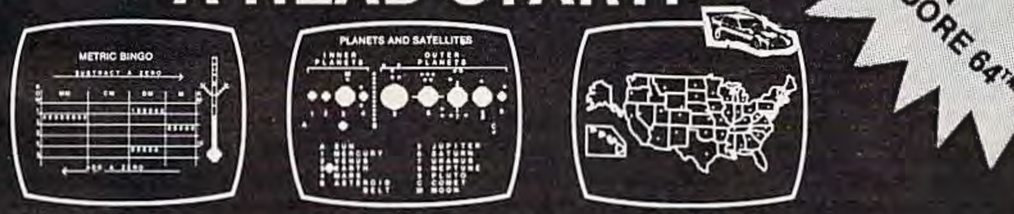
*Commodore 64 is a registered trademark of Commodore Business Machines

---

ONLY $18.95!

GIVE YOUR CHILD A HEAD START!

NEW! FOR COMMODORE 64™



METRIC FEVER
Ages 8-13

SEARCHER OF THE SOLAR SYSTEM
Ages 7-13

RACEWAY U.S.A.
Ages 8-12

MATH MANIA
Ages 6-10

Sungem® Co.

EDUCATIONAL GAMES

- ***Designed by Experienced Educators***
- ***Programmed in Machine Language***
- ***Fast & Smooth Action, Maximum Graphics & Sound***

Each of our educational programs targets a specific learning need, begins with step-by-step interactive learning activities, and ends with a reinforcing game. Our educational software advances the gifted child and motivates the underachiever.

☐ **METRIC FEVER**—Makes it fun to learn the metric conversions of linear, capacity and weight or mass measurements.

☐ **SEARCHER OF THE SOLAR SYSTEM**—Teaches meanings of astronomical terms, order of the inner and outer planets and satellites, many planetary facts.

☐ **RACEWAY U.S.A.**—Identifies each state by shape & name, locates capitals, teaches state nicknames and important points of interest.

☐ **MATH MANIA**—Designed with different levels of mastery for addition, subtraction, multiplication & division; many unique games (drills).

SUNGEM
Educational Games
Dept. CG 84, 900 W. Los Vallecitos
San Marcos, CA 92069 (619) 744-2011

All educational games Copyright SUNGEM CORP. "COMMODORE 64" is a registered trademark of COMMODORE BUSINESS MACHINES, INC.

START MY CHILD LEARNING TODAY!

☐ **Order by mail or phone.** Each educational game comes in either Diskette ($18.95) or Cassette ($15.95). VISA and MASTERCARD welcome. Include $1.50 postage & handling; CA Residents add 6% Sales Tax.

☐ **Send me your free Color Brochure.**

NAME ______________________

ADDRESS ______________________

CITY ______________ STATE _____ ZIP ________ report on the whole program, just leave off the comma and the line numbers. You can search for anything: variables, strings, commands, numbers. One note, however, about looking for things in quotes. The computer won't know if you want the *word* "to" or the *BASIC command* TO unless you use quotation marks instead of the slashes when you're looking for words as literal words. So, to find the word "to" you'd type:

FIND"TO",500-900

CHANGE@PRINT@PRINT#4 @,300–400. Similar to FIND, CHANGE will both locate and *replace* all occurrences of something within your program. All the rules for FIND apply the same way to CHANGE. The example here could be very useful if you have a printer. As written, your program is designed to PRINT everything to the screen. But you can make everything go to the printer instead by adding a line at the start of the program, OPEN 4,4 which alerts the computer that a channel has been opened to the printer. Then this CHANGE will make all printing go to channel 4 (Commodore printers are always Device #4, by convention) instead of the screen.

Another use for CHANGE would become apparent if you'd written a large program and used an illegal variable name like TI (reserved for the clock) or TO (a command name). Instead of hunting through the program, trying to find each illegal variable, just SAVE the program, LOAD and RUN 64 BASIC Aid, LOAD the program back in, and type: CHANGE/TI/TR/ and it's fixed in a flash. Like FIND, the whole program is changed if you leave off the line number information.

KILL turns off 64 BASIC Aid. If you want the computer to be returned to its original state, just type KILL and everything will be as if you'd just turned it on.

Here's the information you need to type in 64 BASIC Aid with the special MLX machine language entry program found elsewhere in this issue:

Start address = 39852
End address = 40961

To activate 64 BASIC Aid once it's loaded, enter SYS 39852.

If you don't want to type this program, send $3, a blank cassette or 1541/4040 disk, and a self-addressed, stamped mailer to:

Harold D. Vanderpool
3705 Enon Xenia Pike
Enon, OH 45323

See program listing on page 203.

Bug-Swatter:

Modifications And Corrections

• Two changes are necessary in the 64 version of "Oil Tycoon" (October). Add the following lines to pick a difficulty level or to press E to end the game:

```
230 PRINT"{DOWN} DIFFICULTY LEVEL? 123456
    78E(END){GRN}":T=1                  :rem 76
235 POKE56194+T,Ø:T1=T:T=T+(PEEK(JS)AND4)
    /4-(PEEK(JS)AND8)/8:IF T>9 THEN T=1
                                        :rem 157
237 IF T<1 THENT=9                      :rem 229
263 IF T=9 THEN SYS2Ø48                 :rem 24Ø
```

When converting the VIC version to the 64, we also neglected to include the feature which allows you to replace the oil rig by pressing the fire button. Add these lines:

```
435 Q=R-1:IF Q=Ø THEN 36Ø              :rem 41
437 J=Z:FORA=ØTO21:A%(A)=Ø:NEXT:B=Q*4Ø+W+
    X:A=PEEK(B):IF A<>7 THEN 48Ø :rem 255
```

• The text-adventure game "Martian Prisoner" (November) does not respond properly when you attempt to hit a Martian guard. This bug escaped our testing because we found it safer to fool the guards rather than provoke them (hint). Nevertheless, if you want to hit the guards, insert a GOSUB command in line 125:

```
125 IFV=14 THEN GOSUB 39ØØ         :rem 154
```

• Many of you who typed in the VIC version of "Aardvark Attack" (October) encountered a syntax error in line 55. The reason is that the programmer used a keyword abbreviation when he wrote the line, and it works fine when typed in with the abbreviation. That's why our testing detected no errors. Unfortunately, listings do not show abbreviations. Here is the line:

```
55 M=(TANDFNM(J))/4 J....           :rem 77
```

The computer interprets T AND FNM(J) as TAN (tangent) without a value. There are four ways to fix it: (1) insert a space between T and AND; (2) abbreviate AND with A-SHIFT-N; (3) put the T in parentheses; (4) reverse the order—FNM(J)ANDT.

• The 64 version of "Munchmath" (November) does not accept answers beginning with 9. To fix this, remove ORAN$>"9" from line 200.

• In November's "Machine Language For Beginners" column, the VIC version of "The Assembler" program requires an 8K memory expander.

LIST Freezer

Doug Ferguson

This very short routine will prove indispensable to BASIC programmers—it allows you to pause or freeze a LISTing of the program on the screen. The routine is a machine language program presented in the form of a BASIC loader, so you need to know nothing about machine language to use it. What's more, it works equally well on the Commodore 64 or VIC-20.

The VIC-20 and Commodore 64 cry out for a pause feature during a LIST. When you're writing or debugging a program, especially if you lack a printer, you can waste a lot of time typing LIST again and again just to get a look at your BASIC code.

"LIST Freezer" is an elegant solution to the problem. It patches directly into the LIST routine in ROM (Read Only Memory) without interfering with anything else. Once it's activated, there is never any need to turn it off. It also eliminates the screen ripple effect of some other LIST pause routines, including one I published in COMPUTE! Magazine in 1982.

The LIST Freezer

The program below activates the pause feature for either the VIC or 64. Type it in exactly, SAVE it, RUN it, LOAD something in BASIC, and give it a try. (Because it destroys the BASIC loader part of itself in line 80, be sure to SAVE it before typing RUN for the first time.)

To use LIST Freezer, LIST any BASIC program and hold down the SHIFT key. The listing will pause. To freeze it entirely while freeing your hands, press SHIFT LOCK. You can restart the listing at any time by releasing SHIFT or SHIFT LOCK.

Technical Details

For the curious, here's how it works. Line 20 sets the low-byte/high-byte address of a machine language "patch" at the top of RAM (Random Access Memory). The patch consumes 23 bytes of memory on either the VIC or 64.

Line 30 redefines the computer's memory size to protect the patch. It also moves the LIST vector at memory addresses 774–775 (hexadecimal \$0306–\$0307) to reroute the indirect jump to ROM (address \$A717 in the 64 or \$C717 in the VIC).

The remaining lines create the patch routine at the top of RAM. Line 50 adjusts the patch to work on either the VIC or 64.

Notice that the program assumes the normal LIST vector at power-up; line 20 thus prevents you from accidentally trying to activate the routine more than once while the power is on.

Also note that the routine clears out the keyboard buffer when activated. Actually this was necessary only for the VIC, but it causes no harm on the 64 and was left in to make the routine universal.

LIST Freezer

```
20 L=232:H=PEEK(56)-1:Q=PEEK(775):IF Q<16
   7 THEN 80                          :rem 236
30 POKE 55,L:POKE 51,L:POKE 56,H:POKE 52,
   H:POKE 774,L:POKE 775,H            :rem 74
40 FOR X=L+H*256 TO X+21:READ D:POKE X,D:
   NEXT                               :rem 51
50 POKE X,Q                           :rem 105
60 DATA 72,152,72,32,159,255,169,1,44,141
   ,2,208,246                         :rem 209
70 DATA 169,0,133,198,104,168,104,76,26
                                      :rem 136
80 NEW                                :rem 82
```

Centering VIC Screens

Mary Conlin

If you've discovered a clever time-saving technique, or a brief but effective programming shortcut, send it in to "Hints & Tips," c/o COMPUTE!'s GAZETTE for Commodore. If we use it, we'll pay you $35.

Some VIC-20s and TV sets don't match up perfectly — the screen image appears to be a little off-center. This is called *overscan*. If you can't compensate by fiddling with the TV controls — or if you prefer to leave the TV controls as they are for regular TV viewing — there's a way to adjust the screen from the computer.

Two memory locations inside the VIC control the horizontal and vertical positions of the screen image. By using simple POKE statements in direct mode or within a program, you can quickly adjust the screen for any TV. This method works on a VIC of any memory size. You can restore the screen to its normal position at any time by holding down the RUN/STOP key and pressing RESTORE.

(The Commodore 64 lacks these adjustments, but has much less need for them because its screen image is smaller than the VIC's and is less subject to overscan.)

Horizontal Adjustments

The horizontal screen position is controlled by memory location 36880. Normally this location contains a 5. POKEing smaller numbers into this location moves the screen left, and POKEing larger numbers moves it right. For example, to move the screen one position left, type:

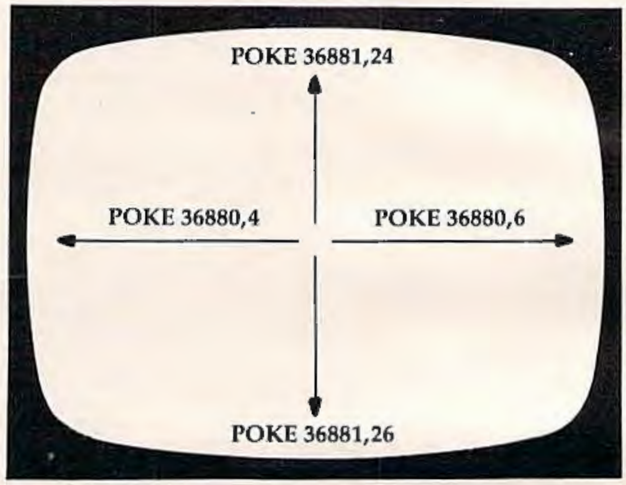
POKE 36880,4 [press RETURN]

Or, to adjust the screen one position right, type:

POKE 36880,6 [press RETURN]

If your screen is off-center by more than one position, try POKEing a 3 or a 7, etc. Once you find the correct value for your particular computer/TV combination, you can include the statement at the beginning of all your BASIC programs so the adjustment is made automatically whenever you type RUN.

Centering VIC Screens On A TV



Vertical Adjustments

The vertical screen position is controlled by memory location 36881. Normally this location contains a 25. POKEing smaller numbers into this location moves the screen up, and POKEing larger numbers moves it down. For example, to move the screen one position up, type:

POKE 36881,24 [press RETURN]

Or, to move the screen one position down, type:

POKE 36881,26 [press RETURN]

Using numbers too high can move the screen completely out of view. If this happens, simply POKE back the 25 or press RUN/STOP-RESTORE. Again, once you find the best number, you can include this statement in all your BASIC programs.

SJB DISTRIBUTORS. ONE STOP SHOPPING FOR COMMODORE SYSTEMS.

commodore

COMMODORE BUSINESS SERIES

SuperPet (5 languages, 2 processors)	$1059
CBM 8032 Computer, 80 Column	625
CBM Memory Expansion, 64K	259
CBM 8050, 1 mg. Dual Drive	995
CBM 8250, 2 mg. Dual Drive	1295
CBM D9060, 5 mg. Hard Disk	1995
CBM D9090, 7.5 mg. Hard Disk	2250
CBM 2031, 170K Single Drive (New)	295
DC Hayes Smart Modem	300

BUSINESS SOFTWARE-8032

Word Pro 4* or 5*	$ 309
InfoPro	219
Administrator	489
VisiCalc (expanded)	199
BPI A/R, G/L, Job Cost, Inventory, Payroll	ea.325
Commodore 64	Call

INTERFACES & ACCESSORIES

80 Column Expander	$ 159
VIC 1600 Modem	95
VIC 1650 (auto answer, auto dial)	150
VIC 1525 Graphic Printer	225
VIC 1530 Datasette Recorder	65
VIC 1541 Disk Drive	249
VIC Switch (connect 8 64's or Vics to printer, dd)	149
PET-IEEE cable	33
IEEE-IEEE cable (2m)	49
5 Slot Expander for 64	65
Parallel Interface (Epson, Okidata, IDS, NEC)	70
Programmers Reference Guide	18
Verbatim Diskettes (10 per box)	26
Hes Modem	75
ADA 1450	149
ADA 1800 (New)	129
Numeric Keypad	35

NEW COMMODORE PRODUCTS

The Executive 64	$ Call
CBM B128-80	825
CBM B256-80	1095
CBM BX700	2990
B Series Software	Call
CBM 1520 Plotter	169
CBM 1526 Printer	349

SOFTWARE FOR CBM 64 BUSINESS

WordPro 3*/64 w/Spell Right Plus	$ 79
Spell Right Plus	55
Calc Result (Advanced)	125
Calc Result (Easy)	75
Busicalc II	95
Mirage Concepts (Powerful Data Base)	95
M File (merge with WordPro)	89
Home Utilities	49
64 Mailing List (Galactic)	28
The Manager	50
Home Accountant (continental)	75
Code Writer (Writes Basic Programs)	95
Stock (investment analysis)	80
Agricultural Management	Call
General Ledger, A/R, A/P, P/R, Inv.	Call

RECREATION

Assembler Package (cassette or disk, compiled, includes editor, loader, disassembler)	$ 39
Sprite Master (access)	30
Neutral Zone (access)	35
Space Belt	19
Coco II (build your own games)	40
Vic Tree (programmers utilities)	75
Micro-Term (save to printer, disk)	39
Hesmon	35
Gothmogs Lair	30
Road Toad	15
Commodore Games	Call

MONITORS - GREAT RESOLUTION (64 OR VIC)

CBM 1701 Color Monitor	$ 249
Amdek Color Plus	299
Panasonic TR-120 (w/speaker)	155
Panasonic CT-160	279
BMC (green screen)	95
Video/Audio Cable	15

PRINTERS - LETTER QUALITY

CBM 6400, 40 cps	$1450
Diablo 620, 25 cps	949
Transtar 140 (serial)	1395
Transtar 130, 16 cps (auto load, wp features!)	769
NEC 3500 Series	1600
NEC 7700 Series	2350
Transtar 120, 14 cps	500

PRINTERS - DOT MATRIX

CBM 8023, 150 cps/graphics	$ 545
CBM 4023 Printer	395
Epson FX Printer, 160 cps	549
Epson MX-80 FT w/graftrax	Call
Epson FX-100	859
Okidata 82A, 120 cps (serial and parallel)	429
NEC 8023A (parallel)	429
Okidata 92	559
Star Gemini, 10X	329
Star Gemini, 15	499
Transtar 315 (hi-res, color)	575

POLICY

VISA/MASTERCARD
C.O.D. FOR CASH OR BANK CHECK

Minimum $50.00 Order
Add 3% Surcharge for Credit Cards.
Exact freight calculated.
Prices subject to change without notice.
Products shipped with manufacturer's warranty.
F.O.B. Dallas, Texas

*Customer's must call for return authorization before returning any product.

PRODUCT SELECTION ADVICE
CUSTOMER SERVICE
(214) 343-1328

SJB DISTRIBUTORS INC.
10520 Plano Road, Suite 206
Dallas, Texas 75238

TO ORDER CALL TOLL FREE
800-527-4893
800-442-1048
(Within Texas)

Business Hours
Mon. - Fri. 8:30 to 5:30
Saturday 10 to 2

Write for free catalog.

NOTE: SJB HAS A FULL LINE OF COMPUTER MEDIA IN STOCK, CALL OR WRITE FOR MORE INFORMATION.

NEWS & PRODUCTS

VIC-20 Spreadsheet

Computer Software Associates has released a VIC-20 version of PractiCalc Plus, a combination spreadsheet-data base manager program.

The program, available on tape or disk, requires a 16K RAM card expander.

PractiCalc Plus allows full use of mathematical and trigonometric function, incorporates search and sort routines, and has a single-key histogram function. The program, which sells for $49.95 on tape and $54.95 on disk, can be used for budget and business projections, expense tracking, investments, and inventory.

Micro Software International, Inc.
The Silk Mill
44 Oak Street
Newton Upper Falls, MA 02164
(617) 527-7510

Space Game For VIC

Ridge Runner is a machine-language space game for the unexpanded VIC-20.

In the game, produced by Bytes and Bits, you maneuver your multicolor ship through a minefield and a volley from enemy ships.

The game, which requires a joystick and sells for $14.95, includes a horizontally scrolling playfield, multicolor graphics, sound, a high-score recorder, and a pause option.

Bytes and Bits
524 East Canterbury Lane
Phoenix, AZ 85022
(602) 942-1475

Commodore 64 Rescue, VIC-20 Educational Games

Zeppelin Rescue, a game of coordination and skill for the Commodore 64, and several educational games for the VIC-20 are available from Micro Software International.

In *Zeppelin Rescue*, you must overcome the forces of gravity and the slow, cumbersome controls of your airship to rescue the inhabitants of a threatened city. The game includes five cityscapes and four levels—daylight, dusk, night, and dawn.

The program is available on disk for $24.95, or on cassette for $19.95.

The VIC-20 programs are all available on cassette and require no memory expansion. They include *Math Duel*, a basic number skills program for students in grades 1 through 6; *Tiny Tutor*, a simple math problem tutorial for children ages 2 through 7; *VIC Sketch*, a drawing program with SAVE and PRINT features; and *Composer*, which teaches the rudiments of music.

Math Duel and *Tiny Tutor* sell for $19.95. *VIC Sketch* and *Composer* are $14.95 each.

Micro Software International
The Silk Mill
44 Oak Street
Newton Upper Falls, MA 02164
(617) 527-7510

One-Handed Bridge

Computer Management Corporation has released *BridgePro*, a program that will allow one person and a Commodore 64 to enjoy a game of bridge.

The program, which is written in machine language, takes care of the shuffling and dealing, and will bid and play three hands.

The program allows replaying hands, prevents illegal bids, and offers a help screen on bidding for bridge newcomers. A game for two players is among *BridgePro*'s other options.

BridgePro is available on disk for $35.

Computer Management Corporation
Customer Service Center
2424 Exbourne Court
Walnut Creek, CA 94596

Help For Programmers

A collection of worksheets, programming aids, grid-sheets, and logs, designed to help simplify VIC-20 and Commodore 64 programming, is available from PM Products.

Programmer's Aids and Logs includes a guide to all keys, POKEs, CHR$ codes, and characters; sound and music worksheets; a condensed BASIC dictionary; grids for screen layout, custom characters, and sprites; and documentation worksheets for variable, subroutine, and file use.

Also included are cutout function key templates and a BASIC-Aid reference card. The package, which sells for $9.95, contains 95 color-coded pages and is punched for use in three-ring binders.

PM Products
4455 Torrance Blvd., #177
Torrance, CA 90503

Programmer's Aids and Logs is a collection of worksheets, reference cards, and programming aids for the VIC or 64.

Tax Preparation Programs

Northland Accounting has produced three tax preparation programs for the Commodore 64 and VIC-20. The programs produce a line-by-line readout of IRS Form 1040 and related schedules. Updates for new tax years will be published annually.

Taxaid I is for the unexpanded VIC-20. The program directs its output to the monitor. *Taxaid II* is for a VIC-20 with 16K. Output can be directed to the monitor or a printer. *Taxaid III* is for the Commodore 64, with output directed to the monitor or printer.

Taxaid I is available on tape for $19.95 or on disk for $24.95. The other two versions are available on tape for $24.95 or on disk for $29.95.

Northland Accounting, Inc.
Software Department
606 Second Ave.
Two Harbors, MN 55616
(218) 834-5012

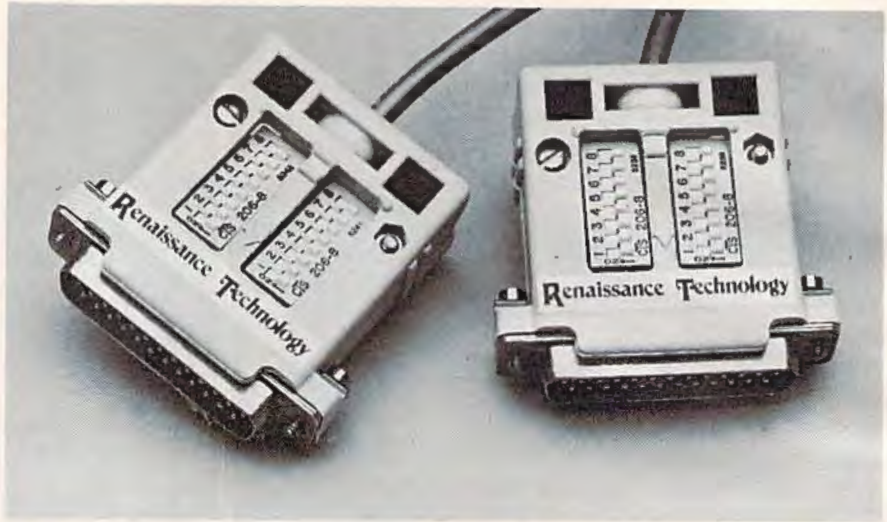
Universal Serial Cable

Renaissance Technology has produced the Universal Serial Cable, which simplifies connections between RS-232 serial computers and peripheral devices.

Built into the cable connectors are sets of DIP switches that can be set according to the requirements of the devices being connected. A cross-reference chart of switch settings is included with the cable.

The Universal Serial Cable sells for $62.

Renaissance Technology Corp.
1070 Shary Circle
Concord, CA 94518
(415) 676-5757



The Universal Serial Cable from Renaissance includes DIP switches in the connectors to eliminate the need for specially wired cables.

Activity-Planning Software

SEI Enterprises has produced a series of programs designed to help groups of users plan their activities.

The programs, which cover vacation, menu, and spending plans, allow up to ten participants to enter their preferences into the computer. Each person is then given a chance to vote on the suggestions made, and the program tallies the results and generates a printout.

The initial series is $24.95 and consists of four programs. They are *Acti-Trip*, for trip planning; *Acti-Spend*, for spending priorities; *Acti-Menu*, for meal planning; and *Acti-Play*, designed for youngsters to determine how to spend their free time.

The programs are available for the Commodore 64 and the VIC-20 with 8K expansion.

SEI Enterprises
17 Serpi Road
Highland Mills, NY 10930

Word Processor For Commodore 64

Easy Script 64, a word processing program from Commodore Software, is available for the Commodore 64.

Features of the program include selectable display colors; hunt and find; search and replace; function key editing; superscripts and subscripts; vertical and horizontal tabs; and the ability to transfer words, phrases and blocks within text. The program includes a form-letter command, and it offers optional sound effect prompts.

Easy Script 64, which sells for $49.95, also is compatible with *Easy Spell 64*, Commodore's spelling checker.

Commodore Software
1200 Wilson Drive
West Chester, PA 19380
(215) 431-9100

VIC Memory Poster

Kevco has produced *Inside the VIC-20*, a 27 by 21-inch color poster that shows the important memory locations in the VIC.

The chart includes information on the BASIC memory map, video screens, sound and color, as well as data on paddles, joysticks and light pens.

The poster, which makes often-used information available at a glance, sells for $6.95. A Commodore 64 version is forthcoming.

Kevco Electronic and Software Engineering
480 Georgia Court
Claremont, CA 91711
(714) 626-4148

Investment Manager

Portfolio Manager is an investment management program for the Commodore 64 or 16K VIC-20 computers.

The program, which is the first in a series of personal finance programs planned by Basic Byte, allows the user to easily record and compute stock transactions. *Portfolio Manager* sells for $29.95.

Basic Byte, Inc.
13108 Ludlow
Huntington Woods, MI 48070
(313) 545-6779

Property Management System

MicroSpec has introduced its *Rental Property Manager* program for the Commodore 64.

With the program, a landlord can keep track of up to 200 rental units per diskette. The program maintains 18 fields per record, including information on the property owner, the tenant, the rent payment record, and the availability of the unit.

The system, which requires one disk drive, can produce a variety of reports including tenant lists, overdue rent lists, expired lease lists, vacancy lists, and income and expense reports.

Rental Property Manager sells for $179.95.

MicroSpec, Inc.
Box 836085
Plano, TX 75086
(214) 867-1333

Educational Games For Commodore 64

Bertamax has reached agreement with Commodore Business Machines to convert 21 educational programs for use on the Commodore 64.

Among the programs being converted are several programs designed for children in kindergarten through third grade. The titles include *Number Match It*, *Addition Match*, *Subtraction Match*, *Multiplication Match*, *Division Match*, and six reading programs in the *Story Mix* series.

Other programs covered in the agreement are: *Spelling in Context*, a 308-lesson program available for grade levels one through eight; *Math Facts Games–Set 2*, a series of four two-player math games; and *Number Cruncher*, 30 lessons in math and problem-solving skills.

Bertamax, Inc.
3647 Stoneway North
Seattle, WA 98103
(206) 547-4056

Spreadsheet For VIC And 64

ESP›Calc is an electronic spreadsheet planning calculator for both the VIC and 64.

The same program runs on both computers, and the size of the spreadsheet is limited only by computer memory. The manual includes step-by-step instructions to help novice spreadsheet users become accustomed to the program.

ESP›Calc is designed to handle things like household budgets, heat and electric use, stock portfolios, and rental property analysis. The program also includes printer options.

The cassette version of *ESP›Calc* sells for $43.50; the disk version is $47.50.

New Leaf Inc.
120 Lynnhaven
Belleville, IL 62223

COMPUTE!'s GAZETTE *welcomes announcements of new products for VIC-20 and Commodore 64 computers, especially products aimed at beginning to intermediate users. Please send press releases and photos well in advance to: Tony Roberts, Assistant Managing Editor,* COMPUTE!'s GAZETTE, *P.O. Box 5406, Greensboro, NC 27403.*

New product releases are selected from submissions for reasons of timeliness, available space, and general interest to our readers. We regret that we are unable to select all new product submissions for publication. Readers should be aware that we present here some edited version of material submitted by vendors and are unable to vouch for its accuracy at time of publication.

How To Type In *COMPUTE!'s Gazette* Programs

Many of the programs which are listed in *COMPUTE!'s Gazette* contain special control characters (cursor control, color keys, inverse video, etc.). To make it easy to know exactly what to type when entering one of these programs into your computer, we have established the following listing conventions.

Generally, any VIC-20 or Commodore 64 program listings will contain bracketed words which spell out any special characters: {DOWN} would mean to press the cursor down key. {5 SPACES} would mean to press the space bar five times.

To indicate that a key should be *shifted* (hold down the SHIFT key while pressing the other key), the key would be underlined in our listings. For example, S̲ would mean to type the S key while holding the shift key. This would appear on your screen as a "heart" symbol. If you find an underlined key enclosed in braces (e.g., {10 N̲}), you should type the key as many times as indicated (in our example, you would enter ten shifted N's).

If a key is enclosed in special brackets, [ ], you should hold down the *Commodore key* while pressing the key inside the special brackets. (The Commodore key is the key in the lower left corner of the keyboard.) Again, if the key is preceded by a number, you should press the key as many times as necessary.

Rarely, you'll see a solitary letter of the alphabet enclosed in braces. These characters can be entered on the Commodore 64 by holding down the CTRL key while typing the letter in the braces. For example, {A} would indicate that you should press CTRL-A. You should never have to enter such a character on the VIC-20, but if you do, you would have to leave the quote mode (press RETURN and cursor back up to the position where the control character should go), press CTRL-9 (RVS ON), the letter in braces, and then CTRL-0 (RVS OFF).

About the *quote mode*: you know that you can move the cursor around the screen with the CRSR keys. Sometimes a programmer will want to move the cursor under program control. That's why you see all the {LEFT}'s, {HOME}'s, and {BLU}'s in our programs. The only way the computer can tell the difference between direct and programmed cursor control is the quote mode.

Once you press the quote (the double quote, SHIFT-2), you are in the quote mode. If you type something and then try to change it by moving the cursor left, you'll only get a bunch of reverse-video lines. These are the symbols for cursor left. The only editing key that isn't programmable is the DEL key; you can still use DEL to back up and edit the line. Once you type another quote, you are out of quote mode.

You also go into quote mode when you INSerT spaces into a line. In any case, the easiest way to get out of quote mode is to just press RETURN. You'll then be out of quote mode and you can cursor up to the mistyped line and fix it.

Use the following table when entering cursor and color control keys:

When You Read:	Press:	See:	When You Read:	Press:	See:	When You Read:	Press:	See:
{CLEAR}	SHIFT CLR/HOME		{CYN}	CTRL 4		[7]	f7	
{HOME}	CLR/HOME		{PUR}	CTRL 5		[8]	f8	
{UP}	SHIFT ↑CRSR↓		{GRN}	CTRL 6		{F1}	F1	
{DOWN}	↑CRSR↓		{BLU}	CTRL 7		{F2}	F2	
{LEFT}	SHIFT ←CRSR→		{YEL}	CTRL 8		{F3}	F3	
{RIGHT}	←CRSR→		[1]	f1		{F4}	F4	
{RVS}	CTRL 9		[2]	f2		{F5}	F5	
{OFF}	CTRL 0		[3]	f3		{F6}	F6	
{BLK}	CTRL 1		[4]	f4		{F7}	F7	
{WHT}	CTRL 2		[5]	f5		{F8}	F8	
{RED}	CTRL 3		[6]	f6				

A Beginner's Guide To Typing In Programs

What Is A Program?

A computer cannot perform any task by itself. Like a car without gas, a computer has *potential*, but without a program, it isn't going anywhere. Most of the programs published in *COMPUTE!'s Gazette* for Commodore are written in a computer language called BASIC. BASIC is easy to learn and is built into all VIC-20s and Commodore 64s.

BASIC Programs

Each month, *COMPUTE!'s Gazette* for Commodore publishes programs for both the VIC and 64. To start out, type in only programs written for your machine, e.g., "VIC Version" if you have a VIC-20. Later, when you gain experience with your computer's BASIC, you can try typing in and converting certain programs from another computer to yours.

Computers can be picky. Unlike the English language, which is full of ambiguities, BASIC usually has only one "right way" of stating something. Every letter, character, or number is significant. A common mistake is substituting a letter such as "O" for the numeral "0", a lowercase "l" for the numeral "1", or an uppercase "B" for the numeral "8". Also, you must enter all punctuation such as colons and commas just as they appear in the magazine. Spacing can be important. To be safe, type in the listings *exactly* as they appear.

Brackets And Special Characters

The exception to this typing rule is when you see the curved bracket, such as "{DOWN}". Anything within a set of brackets is a special character or characters that cannot easily be listed on a printer. When you come across such a special statement, refer to "How To Type In *COMPUTE!'s Gazette* Programs."

About DATA Statements

Some programs contain a section or sections of DATA statements. These lines provide information needed by the program. Some DATA statements contain actual programs (called machine language); others contain graphics codes. These lines are especially sensitive to errors.

If a single number in any one DATA statement is mistyped, your machine could "lock up," or "crash." The keyboard and STOP key may seem "dead," and the screen may go blank. Don't panic – no damage is done. To regain control, you have to turn off your computer, then turn it back on. This will erase whatever program was in memory, *so always SAVE a copy of your program before you RUN it*. If your computer crashes, you can LOAD the program and look for your mistake.

Sometimes a mistyped DATA statement will cause an error message when the program is RUN. The error message may refer to the program line that READs the data. *The error is still in the DATA statements, though.*

Get To Know Your Machine

You should familiarize yourself with your computer before attempting to type in a program. Learn the statements you use to store and retrieve programs from tape or disk. You'll want to save a copy of your program, so that you won't have to type it in every time you want to use it. Learn to use your machine's editing functions. How do you change a line if you made a mistake? You can always retype the line, but you at least need to know how to backspace. Do you know how to enter inverse video, lowercase, and control characters? It's all explained in your computer's manuals.

A Quick Review

1) Type in the program a line at a time, in order. Press RETURN at the end of each line. Use backspace or the back arrow to correct mistakes.

2) Check the line you've typed against the line in the magazine. You can check the entire program again if you get an error when you RUN the program.

3) Make sure you've entered statements in brackets as the appropriate control key (see "How To Type *COMPUTE!'s Gazette* Programs" elsewhere in the magazine.)

We regret that we are not able to respond to individual inquiries about programs, products, or services appearing in COMPUTE!'s Gazette *for Commodore due to increasing publication activity. On those infrequent occasions when a published program contains a typo, the correction will appear in the magazine, usually within eight weeks. If you have specific questions about items or programs which you've seen in* COMPUTE!'s Gazette *for Commodore, please send them to Gazette Feedback, P.O. Box 5406, Greensboro, NC 27403.*

The Automatic Proofreader

"The Automatic Proofreader" will help you type in program listings from COMPUTE!'s Gazette without typing mistakes. It is a short error-checking program that hides itself in memory. When activated, it lets you know immediately after typing a line from a program listing if you have made a mistake. Please read these instructions carefully before typing any programs in COMPUTE!'s Gazette.

Preparing The Proofreader

1. Using the listing below, type in the Proofreader. The same program works on both the VIC-20 and Commodore 64. Be very careful when entering the DATA statements — don't type an l instead of a 1, an O instead of a 0, extra commas, etc.

2. SAVE the Proofreader on tape or disk at least twice *before running it for the first time*. This is very important because the Proofreader erases this part of itself when you first type RUN.

3. After the Proofreader is SAVEd, type RUN. It will check itself for typing errors in the DATA statements and warn you if there's a mistake. Correct any errors and SAVE the corrected version. Keep a copy in a safe place — you'll need it again and again, every time you enter a program from COMPUTE!'s Gazette.

4. When a correct version of the Proofreader is RUN, it activates itself. You are now ready to enter a program listing. If you press RUN/STOP-RESTORE, the Proofreader is disabled. To reactivate it, just type the command SYS 886 and press RETURN.

Using The Proofreader

All VIC and 64 listings in COMPUTE!'s Gazette now have a *checksum number* appended to the end of each line, for example ":rem 123". *Don't enter this statement when typing in a program.* It is just for your information. The rem makes the number harmless if someone does type it in. It will, however, use up memory if you enter it, and it will confuse the Proofreader, even if you entered the rest of the line correctly.

When you type in a line from a program listing and press RETURN, the Proofreader displays a number at the top of your screen. *This checksum number must match the checksum number in the printed listing.* If it doesn't, it means you typed the line differently than the way it is listed. Immediately recheck your typing. Remember, don't type the rem statement with the checksum number; it is published only so you can check it against the number which appears on your screen.

The Proofreader is not picky with spaces. It will not notice extra spaces or missing ones. This is for your convenience, since spacing is generally not important. But occasionally proper spacing *is* important, so be extra careful with spaces, since the Proofreader will catch practically everything else that can go wrong.

There's another thing to watch out for: if you enter the line by using abbreviations for commands, the checksum will not match up. But there is a way to make the Proofreader check it. After entering the line, LIST it. This eliminates the abbreviations. Then move the cursor up to the line and press RETURN. It should now match the checksum. You can check whole groups of lines this way.

Special Tape SAVE Instructions

When you're done typing a listing, you must disable the Proofreader before SAVEing the program on tape. Disable the Proofreader by pressing RUN/STOP-RESTORE (hold down the RUN/STOP key and sharply hit the RESTORE key). This procedure is not necessary for disk SAVEs, *but you must disable the Proofreader this way before a tape SAVE.*

SAVE to tape erases the Proofreader from memory, so you'll have to LOAD and RUN it again if you want to type another listing. SAVE to disk does not erase the Proofreader.

Replace Original Proofreader

If you typed in the original version of the Proofreader (October 1983 issue), you should replace it with the improved version below. We added a POKE to the original version to protect it from being erased when you LOAD another program from tape. The POKE does protect the Proofreader, and the Proofreader itself was not affected. However, a quirk in the VIC-20's operating system means that programs typed in with the Proofreader and SAVEd on tape cannot be LOADed properly later. If you LOAD a program SAVEd while the Proofreader was in memory, you see ?LOAD ERROR. This applies only to VIC tape SAVEs (disk SAVEs work OK, and the quirk was fixed in the Commodore 64).

If you have a program typed in with the original Proofreader and SAVEd on tape, follow this special LOAD procedure:

1. Turn the power off, then on.

2. LOAD the program from tape (disregard the ?LOAD ERROR).

3. Enter: POKE 45,PEEK(174):POKE 46,PEEK(175):CLR

4. ReSAVE the program to tape.

The program will LOAD fine in the future. We strongly recommend that you type in the new version of the Proofreader and discard the old one.

Automatic Proofreader For VIC And 64

```
100 PRINT"{CLR}PLEASE WAIT...":FORI=886TO
    1018:READA:CK=CK+A:POKEI,A:NEXT
110 IF CK<>17539 THEN PRINT"{DOWN}YOU MAD
    E AN ERROR":PRINT"IN DATA STATEMENTS.
    ":END
120 SYS886:PRINT"{CLR}{2 DOWN}PROOFREADER
     ACTIVATED.":NEW
886 DATA 173,036,003,201,150,208
892 DATA 001,096,141,151,003,173
898 DATA 037,003,141,152,003,169
904 DATA 150,141,036,003,169,003
910 DATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,251
958 DATA 003,206,251,003,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 151,003,192,087,208,006
994 DATA 032,205,189,076,235,003
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 251,003,133,214,076,173
1018 DATA 003
```

MLX Machine Language Entry Program

For Commodore 64 And VIC-20

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost failsafe entry of machine language programs published in COMPUTE!'s GAZETTE. You need to know nothing about machine language to use MLX—it was designed for everyone. There are separate versions for the Commodore 64 and expanded VIC-20 (at least 8K). MLX was conceived and written by Program Editor Charles Brannon. Important: MLX is required to type in the machine language programs in this issue.

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

LOAD "filename",1,1 (for tape)
LOAD "filename",8,1 (for disk)

To start the program, you enter a SYS command that transfers control from BASIC to machine language. The starting SYS number always appears in the appropriate article.

Using MLX

Type in and save the correct version of MLX for your computer (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. MLX asks you for two numbers: the starting address and the ending address. These numbers are given in the article accompanying the ML program.

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

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

MLX Commands

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

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later. MLX recognizes these commands:

SHIFT-S: Save	**SHIFT-N: New Address**
SHIFT-L: Load	**SHIFT-D: Display**

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

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

The special MLX commands may seem a bit confusing, but as you work with MLX, they will become valuable. For example, what if you forgot where you stopped typing? Use the Display command to scan memory from the beginning to the end of the program. When you reach the end of your typing, the lines will contain a random pattern of numbers. When you see the end of your typing, press any key to stop the listing. Use the New Address command to continue typing from the proper location.

You can use the Save and Load commands to make copies of the completed program. Use Load to reload the tape or disk, then insert a new tape or disk and use Save to make a new copy.

Be sure to save MLX; it will be used for future ML programs in COMPUTE!'s GAZETTE.

See program listings on page 184.

SpeedScript

(Article on page 38.)

BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs," "A Beginner's Guide To Typing In Programs," and "The Automatic Proofreader" that appear before the Program Listings.

Program 1:
SpeedScript—Commodore 64 Version

```
2049 :011,008,010,000,158,050,238
2055 :048,054,049,000,000,000,158
2061 :032,103,009,076,193,009,179
2067 :165,251,141,051,008,165,032
2073 :252,141,052,008,165,253,128
2079 :141,054,008,165,254,141,026
2085 :055,008,166,181,240,032,207
2091 :169,000,141,186,026,160,213
2097 :000,185,000,000,153,000,131
2103 :000,200,204,186,026,208,111
2109 :244,238,052,008,238,055,128
2115 :008,224,000,240,007,202,236
2121 :208,224,165,180,208,222,000
2127 :096,165,181,170,005,180,108
2133 :208,001,096,024,138,101,141
2139 :252,141,123,008,165,251,007
2145 :141,122,008,024,138,101,119
2151 :254,141,126,008,165,253,026
2157 :141,125,008,232,164,180,191
2163 :208,004,240,013,160,255,227
2169 :185,000,000,153,000,000,203
2175 :136,192,255,208,245,206,089
2181 :123,008,206,126,008,202,038
2187 :208,234,096,169,040,133,251
2193 :195,133,020,169,004,133,031
2199 :196,169,216,133,021,173,035
2205 :182,026,133,155,173,183,241
2211 :026,133,156,162,001,173,046
2217 :185,026,133,012,173,195,125
2223 :026,141,032,208,160,000,230
2229 :173,194,026,145,020,177,148
2235 :155,153,196,026,200,041,190
2241 :127,201,031,240,019,192,235
2247 :040,208,235,136,177,155,126
2253 :041,127,201,032,240,005,083
2259 :136,208,245,160,039,200,175
2265 :132,167,136,185,196,026,035
2271 :145,195,136,016,248,164,103
2277 :167,024,152,101,155,133,193
2283 :155,165,156,105,000,133,181
2289 :156,152,157,060,003,192,193
2295 :040,240,008,169,032,145,113
2301 :195,200,076,246,008,024,234
2307 :165,195,105,040,133,195,068
2313 :133,020,144,004,230,196,224
2319 :230,021,232,224,025,240,219
2325 :003,076,179,008,165,155,095
2331 :141,192,026,165,156,141,080
2337 :193,026,096,169,000,133,138
2343 :155,141,182,026,141,188,104
2349 :026,133,038,169,028,133,060
2355 :156,141,183,026,141,189,119
2361 :026,133,039,169,032,162,106
2367 :179,160,255,198,156,145,132
2373 :155,200,230,156,145,155,086
2379 :200,208,251,230,156,202,042
2385 :208,246,145,155,096,133,040
2391 :167,132,168,160,000,177,123
2397 :167,240,006,032,210,255,235
2403 :200,208,246,096,169,012,006
2409 :141,195,026,169,038,133,039
2415 :001,169,011,141,194,026,141
2421 :032,036,009,169,000,141,248
2427 :185,026,032,115,015,169,153
2433 :255,141,138,002,032,245,174
2439 :012,032,150,009,169,100,095
2445 :160,025,032,086,009,238,179
2451 :184,026,096,032,166,009,148
2457 :169,085,160,025,032,086,198
2463 :009,169,000,141,184,026,176
2469 :096,162,039,169,032,157,052
2475 :000,004,202,016,250,169,044
2481 :019,076,210,255,072,041,082
2487 :128,074,133,167,104,041,062
2493 :063,005,167,096,160,000,168
2499 :177,038,133,002,160,000,193
2505 :177,038,073,128,145,038,032
2511 :032,142,008,032,228,255,136
2517 :208,013,165,162,041,016,050
2523 :240,245,169,000,133,162,144
2529 :076,199,009,170,160,000,071
2535 :165,002,145,038,224,095,132
2541 :208,012,032,160,011,169,061
2547 :032,160,000,145,038,076,182
2553 :193,009,173,184,026,240,050
2559 :007,138,072,032,150,009,151
2565 :104,170,138,201,013,208,071
2571 :002,162,095,138,041,127,064
2577 :201,032,144,070,224,160,080
2583 :208,002,162,032,138,072,125
2589 :173,185,026,240,003,032,176
2595 :140,014,104,032,181,009,003
2601 :160,000,145,038,032,142,046
2607 :008,056,165,038,237,188,227
2613 :026,133,167,165,039,237,052
2619 :189,026,005,167,144,014,092
2625 :165,038,105,000,141,188,190
2631 :026,165,039,105,000,141,035
2637 :189,026,230,038,208,002,002
2643 :230,039,032,231,010,076,189
2649 :193,009,138,174,125,010,226
2655 :221,125,010,240,006,202,131
2661 :208,248,076,193,009,202,013
2667 :138,010,170,169,009,072,163
2673 :169,192,072,189,162,010,139
2679 :072,189,161,010,072,096,207
2685 :035,029,157,137,133,002,106
2691 :012,138,134,020,148,004,075
2697 :019,009,147,135,139,005,079
2703 :136,140,022,145,017,159,250
2709 :018,024,026,016,028,030,035
2715 :006,001,011,008,031,003,215
2721 :150,011,159,011,170,011,161
2727 :227,011,054,012,066,012,037
2733 :080,012,179,012,231,013,188
2739 :139,014,014,014,083,014,201
2745 :201,014,225,014,253,014,138
2751 :024,015,185,015,222,017,157
2757 :205,016,043,018,080,012,059
2763 :179,012,111,018,118,019,148
2769 :023,020,028,012,108,020,164
2775 :186,017,107,023,002,014,052
2781 :039,020,244,012,210,023,001
```

```
2787 :Ø52,Ø25,122,Ø14,Ø32,Ø71,Ø31
2793 :Ø11,Ø56.165,Ø38,237,182,154
2799 :Ø26,133,167,165,Ø39,237,238
28Ø5 :183,Ø26,ØØ5,167,176,Ø3Ø,Ø64
2811 :Ø56,173,182,Ø26,233,ØØØ,153
2817 :133,167,173,183,Ø26,233,148
2823 :Ø28,ØØ5,167,24Ø,Ø13,165,113
2829 :Ø38,141,182,Ø26,165,Ø39,Ø92
2835 :141,183,Ø26,Ø32,142,ØØ8,Ø39
2841 :Ø56,173,192,Ø26,229,Ø38,227
2847 :133,155,173,193,Ø26,229,172
2853 :Ø39,133,156,ØØ5,155,24Ø,253
2859 :ØØ2,176,Ø24,Ø24,173,182,112
2865 :Ø26,1Ø9,Ø61,ØØ3,141,182,Ø59
2871 :Ø26,173,183,Ø26,1Ø5,ØØØ,Ø56
2877 :141,183,Ø26,Ø32,142,ØØ8,Ø81
2883 :Ø76,Ø25,Ø11,Ø96,Ø56,173,248
2889 :188,Ø26,233,ØØØ,133,167,Ø52
2895 :173,189,Ø26,233,2Ø7,ØØ5,144
29Ø1 :167,144,Ø1Ø,169,ØØØ,141,2Ø4
29Ø7 :188,Ø26,169,2Ø7,141,189,243
2913 :Ø26,Ø56,165,Ø38,233,ØØØ,1Ø3
2919 :133,167,165,Ø39,233,Ø28,1ØØ
2925 :ØØ5,167,176,ØØ9,169,ØØØ,123
2931 :133,Ø38,169,Ø28,133,Ø39,143
2937 :Ø96,Ø56,165,Ø38,237,188,133
2943 :Ø26,133,167,165,Ø39,237,126
2949 :189,Ø26,ØØ5,167,176,ØØ1,185
2955 :Ø96,173,188,Ø26,133,Ø38,Ø25
2961 :173,189,Ø26,133,Ø39,Ø96,Ø33
2967 :23Ø,Ø38,2Ø8,ØØ2,23Ø,Ø39,13Ø
2973 :Ø76,231,Ø1Ø,165,Ø38,2Ø8,117
2979 :ØØ2,198,Ø39,198,Ø38,Ø76,2Ø2
2985 :231,Ø1Ø,165,Ø38,133,155,133
2991 :165,Ø39,133,156,198,156,254
2997 :16Ø,255,177,155,2Ø1,Ø32,137
3ØØ3 :24Ø,ØØ4,2Ø1,Ø31,2Ø8,ØØ3,1Ø6
3ØØ9 :136,2Ø8,243,177,155,2Ø1,Ø33
3Ø15 :Ø32,24Ø,ØØ8,2Ø1,Ø31,24Ø,183
3Ø21 :ØØ4,136,2Ø8,243,Ø96,132,ØØØ
3Ø27 :167,Ø56,165,155,1Ø1,167,254
3Ø33 :133,Ø38,165,156,1Ø5,ØØØ,Ø46
3Ø39 :133,Ø39,Ø76,231,Ø1Ø,16Ø,1Ø4
3Ø45 :ØØØ,177,Ø38,2Ø1,Ø32,24Ø,149
3Ø51 :ØØ8,2Ø1,Ø31,24Ø,ØØ4,2ØØ,151
3Ø57 :2Ø8,243,Ø96,2ØØ,24Ø,Ø25,229
3Ø63 :177,Ø38,2Ø1,Ø32,24Ø,247,158
3Ø69 :2Ø1,Ø31,24Ø,243,Ø24,152,12Ø
3Ø75 :1Ø1,Ø38,133,Ø38,165,Ø39,ØØ5
3Ø81 :1Ø5,ØØØ,133,Ø39,Ø76,231,Ø81
3Ø87 :Ø1Ø,173,188,Ø26,133,Ø38,Ø71
3Ø93 :173,189,Ø26,133,Ø39,Ø76,145
3Ø99 :Ø13,Ø12,169,ØØØ,141,182,Ø32
31Ø5 :Ø26,173,189,Ø26,Ø56,233,224
3111 :ØØ4,2Ø1,Ø28,176,ØØ2,169,1Ø7
3117 :Ø28,141,183,Ø26,Ø32,142,Ø85
3123 :ØØ8,Ø76,Ø16,Ø12,238,195,Ø84
3129 :Ø26,173,195,Ø26,Ø41,Ø15,Ø21
3135 :141,195,Ø26,Ø96,238,194,185
3141 :Ø26,173,194,Ø26,Ø41,Ø15,Ø32
3147 :141,194,Ø26,Ø76,142,ØØ8,15Ø
3153 :165,Ø38,133,155,165,Ø39,ØØ8
3159 :133,156,198,156,16Ø,255,121
3165 :177,155,2Ø1,Ø46,24Ø,Ø12,156
3171 :2Ø1,Ø33,24Ø,ØØ8,2Ø1,Ø63,Ø77
3177 :24Ø,ØØ4,2Ø1,Ø31,2Ø8,ØØ4,Ø25
3183 :136,2Ø8,235,Ø96,177,155,Ø94
3189 :2Ø1,Ø46,24Ø,Ø26,2Ø1,Ø33,Ø96
3195 :24Ø,Ø22,2Ø1,Ø63,24Ø,Ø18,139
32Ø1 :2Ø1,Ø31,24Ø,Ø14,136,2Ø8,191
32Ø7 :235,198,156,165,156,2Ø1,222
3213 :ØØØ,176,227,Ø76,169,Ø12,Ø33
3219 :132,167,198,167,2ØØ,24Ø,227
3225 :Ø1Ø,177,155,2Ø1,Ø32,24Ø,2ØØ
3231 :247,136,Ø76,21Ø,Ø11,164,235
3237 :167,Ø76,115,Ø12,169,ØØØ,192
3243 :133,Ø38,169,Ø28,133,Ø39,199
3249 :Ø76,231,Ø1Ø,16Ø,ØØØ,177,Ø63
3255 :Ø38,2Ø1,Ø46,24Ø,Ø29,2Ø1,17Ø
3261 :Ø33,24Ø,Ø25,2Ø1,Ø63,24Ø,223
3267 :Ø21,2Ø1,Ø31,24Ø,Ø17,2ØØ,137
3273 :2Ø8,235,23Ø,Ø39,165,Ø39,Ø93
3279 :2Ø5,189,Ø26,24Ø,226,144,213
3285 :224,Ø76,Ø16,Ø12,2ØØ,24Ø,213
3291 :25Ø,177,Ø38,2Ø1,Ø32,24Ø,133
3297 :247,2Ø1,Ø46,24Ø,243,2Ø1,123
33Ø3 :Ø33,24Ø,239,2Ø1,Ø63,24Ø,223
33Ø9 :235,2Ø1,Ø31,24Ø,231,Ø76,227
3315 :ØØ1,Ø12,169,ØØØ,141,13Ø,184
3321 :Ø27,169,2Ø8,141,131,Ø27,184
3327 :Ø32,166,ØØ9,169,12Ø,16Ø,143
3333 :Ø25,Ø32,Ø86,ØØ9,169,ØØ1,Ø71
3339 :141,184,Ø26,Ø96,Ø56,165,167
3345 :Ø38,233,ØØØ,133,167,165,241
3351 :Ø39,233,Ø28,ØØ5,167,2Ø8,191
3357 :ØØ3,1Ø4,1Ø4,Ø96,165,Ø38,Ø27
3363 :133,251,165,Ø39,133,252,24Ø
3369 :Ø96,Ø56,165,Ø38,133,253,Ø14
3375 :Ø73,255,1Ø1,251,141,134,234
3381 :Ø27,165,Ø39,133,254,Ø73,232
3387 :255,1Ø1,252,141,135,Ø27,2Ø2
3393 :165,251,141,136,Ø27,165,182
3399 :252,141,137,Ø27,165,253,Ø22
34Ø5 :141,138,Ø27,133,251,165,164
3411 :254,141,139,Ø27,133,252,ØØ5
3417 :Ø24,173,135,Ø27,1Ø9,131,176
3423 :Ø27,2Ø1,255,144,Ø2Ø,Ø32,ØØ6
3429 :166,ØØ9,169,135,16Ø,Ø25,253
3435 :Ø32,Ø86,ØØ9,169,ØØ1,141,Ø33
3441 :184,Ø26,169,ØØØ,133,198,Ø55
3447 :Ø96,173,13Ø,Ø27,133,253,163
3453 :173,131,Ø27,133,254,173,248
3459 :134,Ø27,133,18Ø,Ø24,1Ø9,226
3465 :13Ø,Ø27,141,13Ø,Ø27,173,253
3471 :135,Ø27,133,181,1Ø9,131,Ø91
3477 :Ø27,141,131,Ø27,169,ØØØ,132
3483 :141,Ø26,2Ø8,169,Ø32,133,Ø96
3489 :ØØ1,Ø32,Ø19,ØØ8,169,Ø38,172
3495 :133,ØØ1,169,ØØ1,141,Ø26,126
35Ø1 :2Ø8,173,136,Ø27,133,251,Ø77
35Ø7 :173,137,Ø27,133,252,173,Ø5Ø
3513 :138,Ø27,133,253,173,139,Ø24
3519 :Ø27,133,254,Ø56,173,188,254
3525 :Ø26,229,253,133,18Ø,173,167
3531 :189,Ø26,229,254,133,181,191
3537 :Ø32,Ø19,ØØ8,Ø56,173,188,173
3543 :Ø26,237,134,Ø27,141,188,2ØØ
3549 :Ø26,173,189,Ø26,237,135,239
3555 :Ø27,141,189,Ø26,Ø96,Ø32,226
3561 :Ø15,Ø13,Ø32,16Ø,Ø11,Ø32,24Ø
3567 :Ø42,Ø13,Ø56,173,13Ø,Ø27,168
3573 :233,ØØ1,141,13Ø,Ø27,173,182
3579 :131,Ø27,233,ØØØ,141,131,146
3585 :Ø27,Ø96,Ø32,151,Ø11,Ø32,Ø94
```

```
3591 :015,013,032,160,011,076,058
3597 :042,013,032,245,012,169,014
3603 :002,133,012,032,166,009,117
3609 :169,147,160,025,032,086,132
3615 :009,032,228,255,240,251,022
3621 :072,032,150,009,104,041,189
3627 :191,201,023,208,009,032,195
3633 :015,013,032,171,011,076,111
3639 :042,013,201,019,208,009,035
3645 :032,015,013,032,081,012,246
3651 :076,042,013,201,016,208,111
3657 :009,032,015,013,032,025,199
3663 :015,076,042,013,096,056,121
3669 :165,038,237,182,026,133,098
3675 :167,165,039,237,183,026,140
3681 :005,167,240,011,173,182,107
3687 :026,133,038,173,183,026,170
3693 :133,039,096,169,000,133,167
3699 :038,169,028,133,039,076,086
3705 :231,010,160,005,140,156,055
3711 :027,032,140,014,172,156,156
3717 :027,136,208,244,076,228,028
3723 :011,024,165,038,133,251,249
3729 :105,001,133,253,165,039,073
3735 :133,252,105,000,133,254,004
3741 :056,173,188,026,229,253,058
3747 :133,180,173,189,026,229,069
3753 :254,133,181,201,255,208,121
3759 :006,169,001,133,180,230,126
3765 :181,032,080,008,160,000,130
3771 :169,032,145,038,238,188,229
3777 :026,208,003,238,189,026,115
3783 :076,013,012,173,185,026,172
3789 :073,014,141,185,026,096,228
3795 :169,162,160,025,032,086,077
3801 :009,032,228,255,240,251,208
3807 :201,089,096,169,002,133,145
3813 :012,032,166,009,169,185,034
3819 :160,025,032,086,009,032,067
3825 :211,014,240,003,076,150,167
3831 :009,162,255,154,076,013,148
3837 :008,160,000,177,038,201,069
3843 :031,240,015,200,208,247,176
3849 :230,039,165,039,205,189,108
3855 :026,144,238,076,016,012,015
3861 :200,076,001,012,165,038,001
3867 :133,155,165,039,133,156,040
3873 :198,156,160,255,177,155,110
3879 :201,031,240,016,136,192,087
3885 :255,208,245,198,156,165,248
3891 :156,201,028,176,237,076,157
3897 :169,012,056,152,101,155,190
3903 :133,155,169,000,101,156,009
3909 :133,156,056,165,155,229,195
3915 :038,133,167,165,156,229,195
3921 :039,005,167,208,018,132,138
3927 :167,024,165,155,229,167,226
3933 :133,155,165,156,233,000,167
3939 :133,156,076,043,015,165,175
3945 :155,133,038,165,156,133,117
3951 :039,076,231,010,120,169,244
3957 :127,141,013,220,169,027,046
3963 :141,017,208,169,146,141,177
3969 :020,003,169,015,141,021,242
3975 :003,169,001,141,026,208,171
3981 :141,018,208,088,096,169,093
3987 :058,164,012,205,018,208,044
3993 :208,005,169,001,172,195,135
3999 :026,140,033,208,141,018,213
4005 :208,201,001,240,008,169,224
4011 :001,141,025,208,076,188,042
4017 :254,169,001,141,025,208,207
4023 :076,049,234,173,141,002,090
4029 :041,001,208,003,032,245,207
4035 :012,032,166,009,169,200,015
4041 :160,025,032,086,009,160,161
4047 :000,177,038,073,128,145,000
4053 :038,032,142,008,160,000,081
4059 :177,038,073,128,145,038,050
4065 :169,002,133,012,032,228,033
4071 :255,240,251,009,064,201,227
4077 :087,208,009,032,022,016,099
4083 :032,228,011,076,037,016,131
4089 :201,083,208,009,032,022,036
4095 :016,032,180,012,076,037,096
4101 :016,201,080,208,009,032,039
4107 :022,016,032,254,014,076,169
4113 :037,016,076,150,009,165,214
4119 :038,133,253,141,045,027,148
4125 :165,039,133,254,141,046,039
4131 :027,096,056,165,038,133,038
4137 :251,237,045,027,141,134,108
4143 :027,165,039,133,252,237,132
4149 :046,027,141,135,027,032,205
4155 :065,013,173,045,027,133,003
4161 :038,173,046,027,133,039,009
4167 :032,142,008,076,206,015,038
4173 :169,038,229,211,141,190,031
4179 :026,169,000,141,158,027,092
4185 :160,000,169,156,032,210,048
4191 :255,169,018,032,210,255,010
4197 :169,032,032,210,255,169,200
4203 :157,032,210,255,140,191,068
4209 :026,032,228,255,240,251,121
4215 :172,191,026,133,167,169,209
4221 :146,032,210,255,169,032,201
4227 :032,210,255,169,157,032,218
4233 :210,255,169,155,032,210,144
4239 :255,165,167,201,013,240,160
4245 :046,201,020,208,015,136,007
4251 :016,004,200,076,091,016,046
4257 :169,157,032,210,255,076,036
4263 :091,016,041,127,201,032,163
4269 :144,172,204,190,026,240,125
4275 :167,165,167,153,236,026,069
4281 :032,210,255,169,000,133,216
4287 :212,200,076,091,016,032,050
4293 :210,255,169,000,153,236,196
4299 :026,152,096,032,166,009,172
4305 :169,237,160,025,032,086,150
4311 :009,032,051,017,176,031,019
4317 :169,000,133,155,169,028,107
4323 :133,156,174,188,026,172,052
4329 :189,026,169,155,032,216,252
4335 :255,176,010,032,183,255,126
4341 :041,191,208,003,076,028,024
4347 :018,240,039,173,050,017,020
4353 :201,008,144,006,032,169,049
4359 :023,076,028,017,173,050,118
4365 :017,201,001,240,249,032,241
4371 :166,009,169,243,160,025,023
4377 :032,086,009,032,115,015,058
4383 :169,001,141,184,026,096,136
4389 :032,166,009,169,254,160,059
4395 :025,032,086,009,076,028,043
4401 :017,000,032,077,016,240,175
```

```
4407 :024,169,029,160,026,032,239
4413 :086,009,032,228,255,240,143
4419 :251,162,008,201,068,240,229
4425 :012,162,001,201,084,240,005
4431 :006,032,150,009,104,104,228
4437 :096,142,050,017,169,001,048
4443 :160,000,032,186,255,160,116
4449 :000,224,001,240,042,185,021
4455 :236,026,201,064,208,007,077
4461 :185,237,026,201,058,240,032
4467 :028,169,048,141,020,027,036
4473 :169,058,141,021,027,185,210
4479 :236,026,153,022,027,200,023
4485 :204,191,026,144,244,240,158
4491 :242,200,076,156,017,185,247
4497 :236,026,153,020,027,200,039
4503 :204,191,026,208,244,140,140
4509 :044,027,032,166,009,169,092
4515 :236,160,026,032,086,009,200
4521 :173,044,027,162,020,160,243
4527 :027,032,189,255,169,013,092
4533 :032,210,255,076,086,018,090
4539 :032,166,009,169,232,160,187
4545 :025,032,086,009,032,228,093
4551 :255,240,251,032,181,009,143
4557 :009,128,072,173,185,026,030
4563 :240,003,032,140,014,032,160
4569 :150,009,104,076,041,010,095
4575 :056,165,038,233,000,133,080
4581 :167,165,039,233,028,005,098
4587 :167,240,004,169,005,133,185
4593 :012,032,166,009,169,049,166
4599 :160,026,032,086,009,032,080
4605 :051,017,165,012,201,005,192
4611 :240,003,032,036,009,169,236
4617 :000,166,038,164,039,032,192
4623 :213,255,144,003,076,252,190
4629 :016,142,188,026,140,189,210
4635 :026,032,231,255,032,166,001
4641 :009,169,019,160,026,032,192
4647 :086,009,076,028,017,032,031
4653 :166,009,169,055,160,026,118
4659 :032,086,009,032,051,017,022
4665 :169,001,162,000,160,028,065
4671 :032,213,255,032,183,255,009
4677 :041,191,240,211,032,166,182
4683 :009,169,006,160,026,032,221
4689 :086,009,076,028,017,120,161
4695 :169,000,141,026,208,141,004
4701 :033,208,169,049,141,020,201
4707 :003,169,234,141,021,003,158
4713 :169,255,141,013,220,088,223
4719 :096,169,147,032,210,255,252
4725 :169,013,032,210,255,032,060
4731 :086,018,032,162,018,169,096
4737 :013,032,210,255,169,063,103
4743 :160,026,032,086,009,032,224
4749 :228,255,201,013,208,249,015
4755 :032,115,015,076,150,009,032
4761 :032,204,255,169,001,032,078
4767 :195,255,096,032,231,255,199
4773 :169,001,162,008,160,000,153
4779 :032,186,255,169,002,162,209
4785 :078,160,026,032,189,255,149
4791 :032,192,255,176,221,162,197
4797 :001,032,198,255,032,207,146
4803 :255,032,207,255,032,207,159
4809 :255,032,183,255,208,202,056
4815 :032,207,255,240,197,032,146
4821 :204,255,032,228,255,201,108
4827 :032,208,005,032,228,255,211
4833 :240,251,162,001,032,198,085
4839 :255,032,207,255,072,032,060
4845 :207,255,168,104,170,152,013
4851 :160,055,132,001,032,205,060
4857 :189,160,054,132,001,169,186
4863 :032,032,210,255,032,207,255
4869 :255,240,006,032,210,255,235
4875 :076,003,019,169,013,032,067
4881 :210,255,076,199,018,162,169
4887 :000,142,047,027,142,048,173
4893 :027,142,049,027,056,177,251
4899 :155,233,048,144,042,201,090
4905 :010,176,038,014,047,027,097
4911 :046,048,027,014,047,027,000
4917 :046,048,027,014,047,027,006
4923 :046,048,027,014,047,027,012
4929 :046,048,027,013,047,027,017
4935 :141,047,027,200,208,212,138
4941 :230,156,076,033,019,248,071
4947 :173,047,027,013,048,027,162
4953 :240,023,056,173,047,027,143
4959 :233,001,141,047,027,173,205
4965 :048,027,233,000,141,048,086
4971 :027,238,049,027,076,083,095
4977 :019,173,049,027,216,096,181
4983 :056,173,130,027,233,000,226
4989 :141,132,027,173,131,027,244
4995 :233,208,141,133,027,013,118
5001 :132,027,208,016,032,166,206
5007 :009,169,088,160,026,032,115
5013 :086,009,169,001,141,184,227
5019 :026,096,024,165,038,133,125
5025 :251,109,132,027,133,253,042
5031 :165,039,133,252,109,133,230
5037 :027,133,254,056,173,188,236
5043 :026,229,251,133,180,173,147
5049 :189,026,229,252,133,181,171
5055 :024,101,254,201,207,144,098
5061 :016,032,166,009,169,080,157
5067 :160,026,032,086,009,169,173
5073 :001,141,184,026,096,032,177
5079 :080,008,024,173,132,027,147
5085 :133,180,109,188,026,141,230
5091 :188,026,173,133,027,133,139
5097 :181,109,189,026,141,189,044
5103 :026,165,038,133,253,165,251
5109 :039,133,254,169,000,133,205
5115 :251,169,208,133,252,169,153
5121 :000,141,026,208,169,032,065
5127 :133,001,032,019,008,169,113
5133 :038,133,001,169,001,141,240
5139 :026,208,076,231,010,160,218
5145 :000,177,038,170,200,177,019
5151 :038,136,145,038,200,138,214
5157 :145,038,096,160,000,177,141
5163 :038,041,063,240,010,201,124
5169 :027.176,006,177,038,073,034
5175 :064,145,038,076,151,011,028
5181 :133,167,041,064,010,005,225
5187 :167,041,191,133,167,041,039
5193 :032,073,032,010,005,167,136
5199 :096,005,075,066,005,058,128
5205 :002,001,027,014,015,018,162
5211 :000,000,000,000,000,032,123
5217 :166,009,169,132,160,026,247
```

```
5223 :076,086,009,076,129,021,244
5229 :169,004,141,159,027,160,001
5235 :007,173,141,002,041,001,224
5241 :240,054,032,166,009,169,023
5247 :107,160,026,032,086,009,035
5253 :032,228,255,240,251,056,171
5259 :233,048,201,003,144,217,217
5265 :201,008,176,213,141,159,019
5271 :027,032,166,009,169,116,158
5277 :160,026,032,086,009,032,246
5283 :228,255,240,251,056,233,146
5289 :048,048,190,201,010,176,074
5295 :186,168,169,001,174,159,008
5301 :027,032,186,255,169,000,082
5307 :032,189,255,032,096,020,043
5313 :032,192,255,162,001,032,099
5319 :201,255,144,003,076,129,239
5325 :021,169,000,133,155,169,084
5331 :028,133,156,162,000,142,064
5337 :141,027,142,140,027,142,068
5343 :160,027,142,161,027,142,114
5349 :154,027,189,080,020,157,088
5355 :146,027,232,224,007,208,055
5361 :245,169,255,141,155,027,209
5367 :160,000,177,155,016,003,246
5373 :076,073,022,201,031,240,128
5379 :034,153,050,027,200,238,193
5385 :153,027,173,153,027,205,235
5391 :147,027,144,230,136,140,071
5397 :187,026,177,155,201,032,031
5403 :240,009,206,153,027,136,030
5409 :208,244,172,187,026,140,242
5415 :187,026,152,056,101,155,204
5421 :133,155,165,156,105,000,247
5427 :133,156,160,000,173,155,060
5433 :027,201,255,208,003,032,015
5439 :241,021,032,038,022,173,078
5445 :187,026,141,186,026,169,036
5451 :050,133,169,169,027,133,244
5457 :170,032,104,024,032,053,240
5463 :022,173,155,027,205,150,051
5469 :027,144,003,032,153,021,217
5475 :056,165,155,237,188,026,158
5481 :133,167,165,156,237,189,128
5487 :026,005,167,240,034,144,215
5493 :032,169,000,141,140,027,114
5499 :141,149,027,032,153,021,134
5505 :032,225,255,240,251,169,021
5511 :013,032,210,255,032,204,113
5517 :255,169,001,032,195,255,024
5523 :076,150,009,076,247,020,213
5529 :056,173,148,027,237,155,181
5535 :027,168,136,136,240,010,108
5541 :048,008,169,013,032,210,133
5547 :255,136,208,248,173,141,052
5553 :027,240,019,141,186,026,048
5559 :173,144,027,133,169,173,234
5565 :145,027,133,170,032,038,222
5571 :022,032,104,024,169,013,047
5577 :032,210,255,032,210,255,171
5583 :032,210,255,173,152,027,032
5589 :208,026,032,204,255,032,202
5595 :166,009,169,141,160,026,122
5601 :032,086,009,032,228,255,099
5607 :240,251,032,096,020,162,008
5613 :001,032,201,255,238,154,094
5619 :027,173,140,027,240,019,101
5625 :141,186,026,173,142,027,176
5631 :133,169,173,143,027,133,009
5637 :170,032,038,022,032,104,147
5643 :024,169,013,032,210,255,202
5649 :172,149,027,140,155,027,175
5655 :136,136,240,010,048,008,089
5661 :169,013,032,210,255,136,076
5667 :208,248,096,169,032,172,192
5673 :146,027,140,153,027,032,054
5679 :210,255,136,208,250,096,178
5685 :172,151,027,024,152,109,176
5691 :155,027,141,155,027,169,221
5697 :013,032,210,255,136,208,151
5703 :250,096,141,157,027,041,015
5709 :127,032,061,020,201,049,055
5715 :144,007,201,058,176,003,160
5721 :076,075,023,174,162,022,109
5727 :221,162,022,240,012,202,186
5733 :208,248,206,153,027,173,092
5739 :157,027,076,000,021,202,078
5745 :138,010,170,140,156,027,242
5751 :169,022,072,169,134,072,245
5757 :189,174,022,072,189,173,176
5763 :022,072,096,056,173,156,194
5769 :027,101,155,133,155,165,105
5775 :156,105,000,133,156,076,001
5781 :247,020,177,155,201,031,212
5787 :240,001,136,140,156,027,087
5793 :096,010,087,065,076,082,065
5799 :084,066,083,078,072,070,108
5805 :192,022,201,022,210,022,074
5811 :220,022,230,022,240,022,167
5817 :250,022,004,023,007,023,002
5823 :045,023,169,000,141,152,209
5829 :027,200,076,151,022,200,105
5835 :169,001,141,160,027,076,009
5841 :151,022,200,032,022,019,143
5847 :141,146,027,076,151,022,010
5853 :200,032,022,019,141,147,014
5859 :027,076,151,022,200,032,223
5865 :022,019,141,149,027,076,155
5871 :151,022,200,032,022,019,173
5877 :141,150,027,076,151,022,044
5883 :200,032,022,019,141,151,048
5889 :027,076,151,022,076,153,250
5895 :021,056,152,101,155,141,121
5901 :142,027,165,156,105,000,096
5907 :141,143,027,032,037,023,166
5913 :056,152,237,156,027,141,026
5919 :140,027,200,076,151,022,135
5925 :200,177,155,201,031,208,241
5931 :249,136,096,056,152,101,065
5937 :155,141,144,027,165,156,069
5943 :105,000,141,145,027,032,249
5949 :037,023,056,152,237,156,210
5955 :027,141,141,027,200,076,167
5961 :151,022,200,177,155,201,211
5967 :061,240,004,136,076,106,190
5973 :022,200,032,022,019,072,196
5979 :173,157,027,041,015,170,162
5985 :202,104,157,087,020,032,187
5991 :151,022,076,134,022,032,028
5997 :231,255,169,000,032,189,217
6003 :255,169,015,162,008,160,116
6009 :015,032,186,255,032,192,065
6015 :255,144,001,096,032,166,053
6021 :009,169,062,032,210,255,102
```

```
6027 :032,077,016,240,025,162,179
6033 :015,032,201,255,176,012,068
6039 :169,236,160,026,032,086,092
6045 :009,169,013,032,210,255,077
6051 :032,231,255,076,150,009,148
6057 :032,231,255,169,000,032,120
6063 :189,255,169,015,162,008,205
6069 :160,015,032,186,255,032,093
6075 :192,255,176,228,032,166,212
6081 :009,162,015,032,198,255,096
6087 :032,077,016,032,231,255,074
6093 :169,001,141,184,026,096,054
6099 :173,141,002,201,005,240,205
6105 :005,173,158,027,208,037,057
6111 :032,166,009,169,162,160,153
6117 :026,032,086,009,032,077,235
6123 :016,208,003,076,150,009,185
6129 :169,001,141,158,027,141,110
6135 :184,026,169,000,133,155,146
6141 :169,028,133,156,076,017,064
6147 :024,165,038,133,155,165,171
6153 :039,133,156,160,001,076,062
6159 :019,024,160,000,162,000,124
6165 :189,236,026,032,181,009,182
6171 :209,155,240,002,162,255,026
6177 :200,208,011,230,156,165,235
6183 :156,205,189,026,240,002,089
6189 :176,035,232,236,191,026,173
6195 :208,224,024,152,101,155,147
6201 :133,038,165,156,105,000,142
6207 :133,039,056,165,038,237,219
6213 :191,026,133,038,165,039,149
6219 :233,000,133,039,076,231,019
6225 :010,032,166,009,169,172,127
6231 :160,026,032,086,009,169,057
6237 :001,141,184,026,169,000,102
6243 :141,158,027,096,096,160,009
6249 :000,204,186,026,240,248,241
6255 :177,169,048,038,032,061,124
6261 :020,032,021,025,032,210,201
6267 :255,173,161,027,240,010,221
6273 :169,008,032,210,255,169,204
6279 :095,032,210,255,032,225,216
6285 :255,208,005,104,104,076,125
6291 :129,021,200,076,106,024,191
6297 :140,156,027,041,127,032,164
6303 :061,020,201,049,144,017,139
6309 :201,058,176,013,041,015,157
6315 :170,202,189,087,020,032,103
6321 :210,255,076,149,024,201,068
6327 :067,208,026,056,169,080,021
6333 :237,186,026,074,056,237,237
6339 :146,027,168,169,032,032,001
6345 :210,255,136,208,250,172,152
6351 :156,027,076,149,024,201,072
6357 :069,208,017,056,173,147,115
6363 :027,237,186,026,056,237,220
6369 :146,027,168,169,032,076,075
6375 :200,024,201,085,208,008,189
6381 :173,161,027,073,001,141,045
6387 :161,027,201,035,240,003,142
6393 :076,149,024,140,156,027,053
6399 :174,154,027,169,000,160,171
6405 :055,132,001,032,205,189,107
6411 :160,054,132,001,172,156,174
6417 :027,076,149,024,174,160,115
6423 :027,240,026,133,167,041,145
6429 :127,201,065,144,018,201,017
6435 :091,176,014,170,165,167,050
6441 :041,128,073,128,074,074,047
6447 :133,167,138,005,167,096,241
6453 :032,166,009,056,169,000,229
6459 :237,188,026,170,169,207,032
6465 :237,189,026,160,055,132,096
6471 :001,032,205,189,160,054,200
6477 :132,001,169,001,141,184,193
6483 :026,096,014,008,155,211,081
6489 :080,069,069,068,211,067,141
6495 :082,073,080,084,000,032,190
6501 :066,089,032,195,072,065,108
6507 :082,076,069,083,032,194,131
6513 :082,065,078,078,079,078,061
6519 :000,194,085,070,070,069,095
6525 :082,032,195,076,069,065,132
6531 :082,069,068,000,194,085,117
6537 :070,070,069,082,032,198,146
6543 :085,076,076,000,196,069,133
6549 :076,069,084,069,032,040,007
6555 :211,044,215,044,208,041,150
6561 :000,058,032,193,082,069,083
6567 :032,089,079,085,032,083,055
6573 :085,082,069,063,032,040,032
6579 :217,047,206,041,058,000,236
6585 :197,210,193,211,197,032,201
6591 :193,204,204,032,212,197,209
6597 :216,212,000,197,082,065,201
6603 :083,069,032,040,211,044,170
6609 :215,044,208,041,058,032,039
6615 :018,210,197,212,213,210,251
6621 :206,146,032,084,079,032,032
6627 :069,088,073,084,000,203,232
6633 :069,089,058,000,211,065,213
6639 :086,069,058,000,212,065,217
6645 :080,069,032,197,210,210,019
6651 :207,210,000,211,084,079,018
6657 :080,080,069,068,000,214,000
6663 :069,082,073,070,089,032,166
6669 :197,082,082,079,082,000,023
6675 :206,079,032,069,082,082,057
6681 :079,082,083,000,147,032,192
6687 :018,212,146,065,080,069,109
6693 :032,079,082,032,018,196,220
6699 :146,073,083,075,063,000,227
6705 :204,079,065,068,058,000,011
6711 :214,069,082,073,070,089,140
6717 :058,000,208,082,069,083,049
6723 :083,032,018,210,197,212,051
6729 :213,210,206,146,000,036,116
6735 :048,206,079,032,210,079,221
6741 :079,077,000,206,079,032,046
6747 :084,069,088,084,032,073,009
6753 :078,032,066,085,070,070,242
6759 :069,082,046,000,196,069,053
6765 :086,073,067,069,032,035,215
6771 :000,211,069,067,079,078,107
6777 :068,046,032,193,068,068,084
6783 :082,046,032,035,000,208,018
6789 :082,073,078,084,073,078,089
6795 :071,000,206,069,088,084,145
6801 :032,083,072,069,069,084,042
6807 :044,032,018,210,197,212,096
6813 :213,210,206,146,000,200,108
6819 :085,078,084,032,070,079,079
6825 :082,058,000,206,079,084,166
```

```
6831 :032,198,079,085,078,068,203
6837 :000,000,000,000,000,000,181
```

Program 2:
SpeedScript—VIC-20 Version

```
4609 :011,018,010,000,158,052,250
4615 :054,050,049,000,000,000,160
4621 :032,114,019,076,247,019,008
4627 :000,000,000,000,000,000,019
4633 :000,000,165,251,141,059,129
4639 :018,165,252,141,060,018,173
4645 :165,253,141,062,018,165,073
4651 :254,141,063,018,166,181,098
4657 :240,032,169,000,141,129,248
4663 :036,160,000,185,000,000,180
4669 :153,000,000,200,204,129,235
4675 :036,208,244,238,060,018,103
4681 :238,063,018,224,000,240,088
4687 :007,202,208,224,165,180,041
4693 :208,222,096,165,181,170,103
4699 :005,180,208,001,096,024,093
4705 :138,101,252,141,131,018,110
4711 :165,251,141,130,018,024,064
4717 :138,101,254,141,134,018,127
4723 :165,253,141,133,018,232,033
4729 :164,180,208,004,240,013,162
4735 :160,255,185,000,000,153,112
4741 :000,000,136,192,255,208,156
4747 :245,206,131,018,206,134,055
4753 :018,202,208,234,096,169,048
4759 :022,133,195,133,020,169,055
4765 :016,133,196,169,148,133,184
4771 :021,173,125,036,133,155,038
4777 :173,126,036,133,156,173,198
4783 :128,036,032,223,019,162,007
4789 :001,160,000,173,137,036,176
4795 :145,020,177,155,153,139,208
4801 :036,200,041,127,201,031,061
4807 :240,019,192,022,208,235,091
4813 :136,177,155,041,127,201,018
4819 :032,240,005,136,208,245,053
4825 :160,021,200,132,167,136,009
4831 :185,139,036,145,195,136,035
4837 :016,248,164,167,024,152,232
4843 :101,155,133,155,165,156,076
4849 :105,000,133,156,152,157,176
4855 :060,003,192,022,240,008,004
4861 :169,032,145,195,200,076,046
4867 :249,018,024,165,195,105,247
4873 :022,133,195,133,020,144,144
4879 :004,230,196,230,021,232,160
4885 :224,023,240,003,076,182,001
4891 :018,165,155,141,135,036,165
4897 :165,156,141,136,036,096,251
4903 :173,019,018,133,155,141,166
4909 :125,036,141,131,036,133,135
4915 :038,173,020,018,133,156,077
4921 :141,126,036,141,132,036,157
4927 :133,039,056,173,022,018,248
4933 :237,020,018,170,169,032,203
4939 :160,255,198,156,145,155,120
4945 :200,230,156,145,155,200,143
4951 :208,251,230,156,202,208,062
4957 :246,145,155,096,133,167,011
4963 :132,168,160,000,177,167,135
4969 :240,006,032,210,255,200,024
4975 :208,246,096,169,001,141,204
4981 :138,036,032,174,022,169,176
4987 :000,141,019,018,141,021,207
4993 :018,141,023,018,141,025,239
4999 :018,024,173,130,002,105,075
5005 :020,141,020,018,056,173,057
5011 :132,002,233,001,141,026,170
5017 :018,056,233,004,141,024,117
5023 :018,056,233,001,141,022,118
5029 :018,169,000,141,137,036,154
5035 :032,039,019,169,000,141,059
5041 :128,036,169,255,141,138,020
5047 :002,032,121,023,032,203,084
5053 :019,169,073,160,035,032,165
5059 :097,019,169,000,141,127,236
5065 :036,096,162,021,169,160,077
5071 :157,000,016,202,016,250,080
5077 :169,019,032,210,255,169,043
5083 :018,076,210,255,141,134,029
5089 :002,162,021,157,000,148,203
5095 :202,016,250,096,072,041,140
5101 :128,074,133,167,104,041,116
5107 :063,005,167,096,160,000,222
5113 :177,038,133,002,160,000,247
5119 :177,038,073,128,145,038,086
5125 :032,150,018,173,141,002,009
5131 :041,004,240,009,165,197,155
5137 :201,064,240,003,076,161,250
5143 :020,032,228,255,208,013,011
5149 :165,162,041,016,240,229,114
5155 :169,000,133,162,076,253,060
5161 :019,170,160,000,165,002,045
5167 :145,038,224,095,208,012,001
5173 :032,007,022,169,032,160,219
5179 :000,145,038,076,247,019,072
5185 :173,127,036,240,007,138,018
5191 :072,032,187,019,104,170,143
5197 :138,201,013,208,002,162,033
5203 :095,138,041,127,201,032,205
5209 :144,092,224,160,208,002,151
5215 :162,032,138,072,173,128,032
5221 :036,240,003,032,007,025,188
5227 :104,032,235,019,160,000,145
5233 :145,038,032,150,018,056,040
5239 :165,038,237,131,036,133,091
5245 :167,165,039,237,132,036,133
5251 :005,167,144,014,165,038,152
5257 :105,000,141,131,036,165,203
5263 :039,105,000,141,132,036,084
5269 :230,038,208,002,230,039,128
5275 :032,067,021,076,247,019,105
5281 :160,000,165,002,145,038,159
5287 :024,165,197,105,064,170,124
5293 :132,162,165,162,201,006,233
5299 :208,250,132,198,138,174,255
5305 :217,020,221,217,020,240,096
5311 :006,202,208,248,076,247,154
5317 :019,202,138,010,170,169,137
5323 :019,072,169,246,072,189,202
5329 :254,020,072,189,253,020,249
5335 :072,096,035,029,157,137,229
5341 :133,099,085,138,134,020,062
5347 :148,082,019,076,147,135,066
5353 :139,113,136,140,091,145,229
5359 :017,121,074,090,097,077,203
5365 :070,118,072,081,108,107,033
5371 :110,003,252,021,006,022,153
5377 :018,022,076,022,162,022,067
```

```
5383 :193,022,208,022,055,023,018
5389 :094,024,006,025,133,024,063
5395 :203,024,068,025,092,025,200
5401 :122,025,149,025,241,025,100
5407 :255,027,242,026,083,028,180
5413 :208,022,055,023,127,028,244
5419 :120,029,013,030,134,022,135
5425 :098,030,219,027,105,033,049
5431 :121,024,029,030,120,023,146
5437 :208,033,046,035,245,024,140
5443 :032,165,021,056,165,038,032
5449 :237,125,036,133,167,165,168
5455 :039,237,126,036,005,167,177
5461 :176,032,056,173,125,036,171
5467 :237,019,018,133,167,173,070
5473 :126,036,237,020,018,005,027
5479 :167,240,013,165,038,141,099
5485 :125,036,165,039,141,126,229
5491 :036,032,150,018,056,173,068
5497 :135,036,229,038,133,155,079
5503 :173,136,036,229,039,133,105
5509 :156,005,155,240,002,176,099
5515 :024,024,173,125,036,109,118
5521 :061,003,141,125,036,173,172
5527 :126,036,105,000,141,126,173
5533 :036,032,150,018,076,119,076
5539 :021,096,056,173,131,036,164
5545 :237,021,018,133,167,173,150
5551 :132,036,237,022,018,005,113
5557 :167,144,012,173,021,018,204
5563 :141,131,036,173,022,018,196
5569 :141,132,036,056,165,038,249
5575 :237,019,018,133,167,165,170
5581 :039,237,020,018,005,167,179
5587 :176,011,173,019,018,133,229
5593 :038,173,020,018,133,039,126
5599 :096,056,165,038,237,131,178
5605 :036,133,167,165,039,237,238
5611 :132,036,005,167,176,001,240
5617 :096,173,131,036,133,038,080
5623 :173,132,036,133,039,096,088
5629 :230,038,208,002,230,039,232
5635 :032,067,021,096,165,038,166
5641 :208,002,198,039,198,038,180
5647 :032,067,021,096,165,038,178
5653 :133,155,165,039,133,156,034
5659 :198,156,160,255,177,155,104
5665 :201,032,240,004,201,031,230
5671 :208,003,136,208,243,177,246
5677 :155,201,032,240,008,201,114
5683 :031,240,004,136,208,243,145
5689 :096,132,167,056,165,155,060
5695 :101,167,133,038,165,156,055
5701 :105,000,133,039,032,067,189
5707 :021,096,160,000,177,038,055
5713 :201,032,240,008,201,031,026
5719 :240,004,200,208,243,096,054
5725 :200,240,026,177,038,201,207
5731 :032,240,247,201,031,240,066
5737 :243,024,152,101,038,133,028
5743 :038,165,039,105,000,133,079
5749 :039,032,067,021,096,173,033
5755 :131,036,133,038,173,132,254
5761 :036,133,039,076,118,022,041
5767 :169,000,141,125,036,173,011
5773 :132,036,056,233,004,205,039
5779 :020,018,176,003,173,020,045
5785 :018,141,126,036,032,150,144
5791 :018,076,122,022,238,138,005
5797 :036,173,138,036,041,015,092
5803 :141,138,036,010,010,010,004
5809 :010,133,167,173,138,036,066
5815 :041,007,024,105,008,101,213
5821 :167,141,015,144,096,238,222
5827 :137,036,173,137,036,041,243
5833 :007,141,137,036,032,150,192
5839 :018,096,165,038,133,155,044
5845 :165,039,133,156,198,156,036
5851 :160,255,177,155,201,046,189
5857 :240,012,201,033,240,008,191
5863 :201,063,240,004,201,031,203
5869 :208,004,136,208,235,096,100
5875 :177,155,201,046,240,027,065
5881 :201,033,240,023,201,063,242
5887 :240,019,201,031,240,015,233
5893 :136,208,235,198,156,165,079
5899 :156,205,019,018,176,226,043
5905 :076,042,023,132,167,198,143
5911 :167,200,240,010,177,155,204
5917 :201,032,240,247,136,076,193
5923 :058,022,164,167,076,243,253
5929 :022,173,019,018,133,038,188
5935 :173,020,018,133,039,032,206
5941 :067,021,096,160,000,177,062
5947 :038,201,046,240,029,201,046
5953 :033,240,025,201,063,240,099
5959 :021,201,031,240,017,200,013
5965 :208,235,230,039,165,039,225
5971 :205,132,036,240,226,144,042
5977 :224,076,122,022,200,240,205
5983 :250,177,038,201,032,240,009
5989 :247,201,046,240,243,201,255
5995 :033,240,239,201,063,240,099
6001 :235,201,031,240,231,076,103
6007 :106,022,173,023,018,141,090
6013 :073,037,173,024,018,141,079
6019 :074,037,032,203,019,169,153
6025 :088,160,035,032,097,019,056
6031 :169,001,141,127,036,096,201
6037 :056,165,038,237,019,018,170
6043 :133,167,165,039,237,020,148
6049 :018,005,167,208,003,104,154
6055 :104,096,165,038,133,251,186
6061 :165,039,133,252,096,056,146
6067 :165,038,133,253,073,255,072
6073 :101,251,141,077,037,165,189
6079 :039,133,254,073,255,101,022
6085 :252,141,078,037,165,251,097
6091 :141,079,037,165,252,141,250
6097 :080,037,165,253,141,081,198
6103 :037,133,251,165,254,141,172
6109 :082,037,133,252,056,173,186
6115 :078,037,109,074,037,205,255
6121 :026,018,144,020,032,203,164
6127 :019,169,103,160,035,032,245
6133 :097,019,169,001,141,127,031
6139 :036,169,000,133,198,096,115
6145 :173,073,037,133,253,173,075
6151 :074,037,133,254,173,077,243
6157 :037,133,180,024,109,073,057
6163 :037,141,073,037,173,078,046
6169 :037,133,181,109,074,037,084
6175 :141,074,037,032,027,018,104
```

```
6181 :173,079,037,133,251,173,115
6187 :080,037,133,252,173,081,031
6193 :037,133,253,173,082,037,252
6199 :133,254,056,173,131,036,070
6205 :229,253,133,180,173,132,137
6211 :036,229,254,133,181,032,164
6217 :027,018,056,173,131,036,002
6223 :237,077,037,141,131,036,226
6229 :173,132,036,237,078,037,010
6235 :141,132,036,096,032,149,165
6241 :023,032,007,022,032,178,135
6247 :023,056,173,073,037,233,186
6253 :001,141,073,037,173,074,096
6259 :037,233,000,141,074,037,125
6265 :096,032,253,021,032,149,192
6271 :023,032,007,022,076,178,209
6277 :023,032,121,023,169,002,247
6283 :032,223,019,032,203,019,155
6289 :169,115,160,035,032,097,241
6295 :019,032,228,255,240,251,152
6301 :072,032,187,019,104,041,100
6307 :191,201,023,208,009,032,059
6313 :149,023,032,019,022,076,234
6319 :178,023,201,019,208,009,045
6325 :032,149,023,032,209,022,136
6331 :076,178,023,201,016,208,121
6337 :009,032,149,023,032,150,076
6343 :025,076,178,023,096,056,141
6349 :165,038,237,125,036,133,171
6355 :167,165,039,237,126,036,213
6361 :005,167,240,011,173,125,170
6367 :036,133,038,173,126,036,253
6373 :133,039,096,173,019,018,195
6379 :133,038,173,020,018,133,238
6385 :039,032,067,021,096,160,144
6391 :005,140,099,037,032,007,055
6397 :025,172,099,037,136,208,162
6403 :244,076,077,022,024,165,099
6409 :038,133,251,105,001,133,158
6415 :253,165,039,133,252,105,194
6421 :000,133,254,056,173,131,000
6427 :036,229,253,133,180,173,007
6433 :132,036,229,254,133,181,230
6439 :201,255,208,006,169,001,111
6445 :133,180,230,181,032,088,121
6451 :018,160,000,169,032,145,063
6457 :038,238,131,036,208,003,199
6463 :238,132,036,076,118,022,173
6469 :173,128,036,073,006,141,114
6475 :128,036,096,169,130,160,026
6481 :035,032,097,019,032,228,012
6487 :255,240,251,201,089,096,195
6493 :032,203,019,169,141,160,049
6499 :035,032,097,019,169,002,197
6505 :032,223,019,032,078,025,002
6511 :240,004,032,187,019,096,177
6517 :162,255,154,076,013,018,027
6523 :160,000,177,038,201,031,218
6529 :240,015,200,208,247,230,245
6535 :039,165,039,205,132,036,239
6541 :144,238,076,122,022,200,175
6547 :076,106,022,165,038,133,175
6553 :155,165,039,133,156,198,231
6559 :156,160,255,177,155,201,239
6565 :031,240,017,136,192,255,012
6571 :208,245,198,156,165,156,019
6577 :205,020,018,176,236,076,140
6583 :042,023,056,152,101,155,200
6589 :133,155,169,000,101,156,135
6595 :133,156,056,165,155,229,065
6601 :038,133,167,165,156,229,065
6607 :039,005,167,208,018,132,008
6613 :167,024,165,155,229,167,096
6619 :133,155,165,156,233,000,037
6625 :133,156,076,168,025,165,180
6631 :155,133,038,165,156,133,243
6637 :039,032,067,021,096,173,153
6643 :141,002,041,001,208,003,127
6649 :032,121,023,032,203,019,167
6655 :169,152,160,035,032,097,132
6661 :019,160,000,177,038,073,216
6667 :128,145,038,032,150,018,010
6673 :160,000,177,038,073,128,081
6679 :145,038,169,002,032,223,120
6685 :019,032,228,255,240,251,030
6691 :009,064,201,087,208,009,101
6697 :032,079,026,032,077,022,053
6703 :076,094,026,201,083,208,223
6709 :009,032,079,026,032,056,031
6715 :023,076,094,026,201,080,047
6721 :208,009,032,079,026,032,195
6727 :123,025,076,094,026,076,235
6733 :187,019,165,038,133,253,104
6739 :141,244,036,165,039,133,073
6745 :254,141,245,036,096,056,149
6751 :165,038,133,251,237,244,139
6757 :036,141,077,037,165,039,084
6763 :133,252,237,245,036,141,127
6769 :078,037,032,201,023,173,145
6775 :244,036,133,038,173,245,220
6781 :036,133,039,032,150,018,021
6787 :076,006,026,169,022,229,147
6793 :211,141,133,036,169,000,059
6799 :141,101,037,160,000,169,239
6805 :166,032,210,255,169,157,114
6811 :032,210,255,140,134,036,194
6817 :032,228,255,240,251,172,059
6823 :134,036,133,167,169,032,070
6829 :032,210,255,169,157,032,004
6835 :210,255,165,167,201,013,166
6841 :240,046,201,020,208,015,147
6847 :136,016,004,200,076,148,003
6853 :026,169,157,032,210,255,022
6859 :076,148,026,041,127,201,054
6865 :032,144,192,204,133,036,182
6871 :240,187,165,167,153,179,026
6877 :036,032,210,255,169,000,155
6883 :133,212,200,076,148,026,254
6889 :032,210,255,169,000,153,028
6895 :179,036,152,096,032,203,169
6901 :019,169,180,160,035,032,072
6907 :097,019,032,087,027,176,177
6913 :033,173,019,018,133,155,020
6919 :173,020,018,133,156,174,169
6925 :131,036,172,132,036,169,177
6931 :155,032,216,255,176,010,095
6937 :032,183,255,041,191,208,167
6943 :003,076,068,028,240,036,226
6949 :173,086,027,201,008,144,164
6955 :006,032,167,033,076,067,168
6961 :027,173,086,027,201,001,052
6967 :240,249,032,203,019,169,199
6973 :186,160,035,032,097,019,078
6979 :169,001,141,127,036,096,125
```

```
6985 :032,203,019,169,197,160,085
6991 :035,032,097,019,076,067,149
6997 :027,000,032,134,026,240,032
7003 :024,169,228,160,035,032,227
7009 :097,019,032,228,255,240,200
7015 :251,162,008,201,068,240,009
7021 :012,162,001,201,084,240,041
7027 :006,032,187,019,104,104,055
7033 :096,142,086,027,169,001,130
7039 :160,000,032,186,255,160,152
7045 :000,224,001,240,042,185,057
7051 :179,036,201,064,208,007,066
7057 :185,180,036,201,058,240,021
7063 :028,169,048,141,219,036,024
7069 :169,058,141,220,036,185,198
7075 :179,036,153,221,036,200,220
7081 :204,134,036,144,244,240,147
7087 :242,200,076,192,027,185,073
7093 :179,036,153,219,036,200,236
7099 :204,134,036,208,244,140,129
7105 :243,036,032,203,019,169,127
7111 :179,160,036,032,097,019,210
7117 :173,243,036,162,219,160,174
7123 :036,032,189,255,169,013,137
7129 :076,210,255,032,203,019,244
7135 :169,175,160,035,032,097,123
7141 :019,032,228,255,240,251,230
7147 :032,235,019,009,128,072,218
7153 :173,128,036,240,003,032,085
7159 :007,025,032,187,019,104,109
7165 :076,111,020,056,165,038,207
7171 :237,019,018,133,155,165,218
7177 :039,237,020,018,005,155,227
7183 :133,155,240,004,169,006,210
7189 :133,155,032,203,019,169,220
7195 :248,160,035,032,097,019,106
7201 :165,155,032,223,019,032,147
7207 :087,027,165,155,208,003,172
7213 :032,039,019,169,000,166,214
7219 :038,164,039,032,213,255,024
7225 :144,003,076,035,027,142,228
7231 :131,036,140,132,036,032,058
7237 :231,255,032,203,019,169,210
7243 :218,160,035,032,097,019,124
7249 :076,067,027,032,203,019,249
7255 :169,254,160,035,032,097,066
7261 :019,032,087,027,169,001,172
7267 :174,019,018,172,020,018,008
7273 :032,213,255,032,183,255,051
7279 :041,191,240,209,032,203,003
7285 :019,169,205,160,035,032,225
7291 :097,019,076,067,027,169,066
7297 :147,032,210,255,169,013,187
7303 :032,210,255,032,172,028,096
7309 :169,013,032,210,255,169,221
7315 :006,160,036,032,097,019,241
7321 :032,228,255,201,013,208,066
7327 :249,076,187,019,032,204,158
7333 :255,169,001,032,195,255,048
7339 :096,032,231,255,169,001,187
7345 :162,008,160,000,032,186,213
7351 :255,169,002,162,021,160,184
7357 :036,032,189,255,032,192,157
7363 :255,176,221,162,001,032,018
7369 :198,255,032,207,255,032,156
7375 :207,255,032,207,255,032,171
7381 :183,255,208,202,032,207,020
7387 :255,240,197,032,204,255,122
7393 :032,228,255,201,032,208,157
7399 :005,032,228,255,240,251,218
7405 :162,001,032,198,255,032,149
7411 :207,255,072,032,207,255,247
7417 :168,104,170,152,032,205,056
7423 :221,169,032,032,210,255,150
7429 :032,207,255,240,006,032,009
7435 :210,255,076,005,029,169,243
7441 :013,032,210,255,076,209,044
7447 :028,162,000,142,246,036,125
7453 :142,247,036,142,248,036,112
7459 :056,177,155,233,048,144,080
7465 :042,201,010,176,038,014,010
7471 :246,036,046,247,036,014,160
7477 :246,036,046,247,036,014,166
7483 :246,036,046,247,036,014,172
7489 :246,036,046,247,036,013,177
7495 :246,036,141,246,036,200,208
7501 :208,212,230,156,076,035,226
7507 :029,248,173,246,036,013,060
7513 :247,036,240,023,056,173,096
7519 :246,036,233,001,141,246,230
7525 :036,173,247,036,233,000,058
7531 :141,247,036,238,248,036,029
7537 :076,085,029,173,248,036,248
7543 :216,096,056,173,073,037,002
7549 :237,023,018,141,075,037,144
7555 :173,074,037,237,024,018,182
7561 :141,076,037,013,075,037,004
7567 :208,016,032,203,019,169,022
7573 :031,160,036,032,097,019,012
7579 :169,001,141,127,036,096,213
7585 :024,165,038,133,251,109,113
7591 :075,037,133,253,165,039,101
7597 :133,252,109,076,037,133,145
7603 :254,056,173,131,036,229,034
7609 :251,133,180,173,132,036,066
7615 :229,252,133,181,024,101,087
7621 :254,205,022,018,144,016,088
7627 :032,203,019,169,023,160,041
7633 :036,032,097,019,169,001,051
7639 :141,127,036,096,032,088,223
7645 :018,024,173,075,037,133,169
7651 :180,109,131,036,141,131,187
7657 :036,173,076,037,133,181,101
7663 :109,132,036,141,132,036,057
7669 :165,038,133,253,165,039,014
7675 :133,254,173,023,018,133,217
7681 :251,173,024,018,133,252,084
7687 :032,027,018,032,067,021,204
7693 :096,160,000,177,038,170,142
7699 :200,177,038,136,145,038,241
7705 :200,138,145,038,096,160,034
7711 :000,177,038,041,063,240,078
7717 :010,201,027,176,006,177,122
7723 :038,073,064,145,038,076,221
7729 :253,021,133,167,041,064,216
7735 :010,005,167,041,191,133,090
7741 :167,041,032,073,032,010,160
7747 :005,167,096,005,075,066,225
7753 :005,058,002,001,027,014,180
7759 :015,018,000,000,000,000,112
7765 :000,032,203,019,169,075,071
7771 :160,036,076,097,019,076,043
7777 :121,031,169,004,141,102,153
7783 :037,160,007,173,141,002,111
```

```
7789 :Ø41,ØØ1,24Ø,Ø54,Ø32,2Ø3,168
7795 :Ø19,169,Ø5Ø,16Ø,Ø36,Ø32,Ø69
78Ø1 :Ø97,Ø19,Ø32,228,255,24Ø,224
78Ø7 :251,Ø56,233,Ø48,2Ø1,ØØ3,151
7813 :144,217,2Ø1,ØØ8,176,213,Ø68
7819 :141,1Ø2,Ø37,Ø32,2Ø3,Ø19,161
7825 :169,Ø59,16Ø,Ø36,Ø32,Ø97,186
7831 :Ø19,Ø32,228,255,24Ø,251,152
7837 :Ø56,233,Ø48,Ø48,19Ø,2Ø1,165
7843 :Ø1Ø,176,186,168,169,ØØ1,1Ø5
7849 :174,1Ø2,Ø37,Ø32,186,255,187
7855 :169,ØØØ,Ø32,189,255,Ø32,Ø84
7861 :Ø86,Ø3Ø,Ø32,192,255,162,17Ø
7867 :ØØ1,Ø32,2Ø1,255,144,ØØ3,Ø55
7873 :Ø76,121,Ø31,173,Ø19,Ø18,119
7879 :133,155,173,Ø2Ø,Ø18,133,Ø63
7885 :156,162,ØØØ,142,Ø84,Ø37,Ø18
7891 :142,Ø83,Ø37,142,1Ø3,Ø37,243
7897 :142,1Ø4,Ø37,142,Ø97,Ø37,ØØ8
79Ø3 :189,Ø7Ø,Ø3Ø,157,Ø89,Ø37,Ø27
79Ø9 :232,224,ØØ7,2Ø8,245,169,Ø34
7915 :255,141,Ø98,Ø37,16Ø,ØØØ,158
7921 :177,155,Ø16,ØØ3,Ø76,Ø71,227
7927 :Ø32,2Ø1,Ø31,24Ø,Ø34,153,17Ø
7933 :249,Ø36,2ØØ,238,Ø96,Ø37,Ø85
7939 :173,Ø96,Ø37,2Ø5,Ø9Ø,Ø37,129
7945 :144,23Ø,136,14Ø,13Ø,Ø36,Ø57
7951 :177,155,2Ø1,Ø32,24Ø,ØØ9,Ø61
7957 :2Ø6,Ø96,Ø37,136,2Ø8,244,18Ø
7963 :172,13Ø,Ø36,14Ø,13Ø,Ø36,159
7969 :152,Ø56,1Ø1,155,133,155,Ø17
7975 :165,156,1Ø5,ØØØ,133,156,242
7981 :16Ø,ØØØ,173,Ø98,Ø37,2Ø1,2Ø2
7987 :255,2Ø8,ØØ3,Ø32,234,Ø31,Ø46
7993 :Ø32,Ø36,Ø32,173,13Ø,Ø36,24Ø
7999 :141,129,Ø36,169,249,133,152
8ØØ5 :169,169,Ø36,133,17Ø,Ø32,Ø1Ø
8Ø11 :1Ø5,Ø34,Ø32,Ø51,Ø32,173,246
8Ø17 :Ø98,Ø37,2Ø5,Ø93,Ø37,144,183
8Ø23 :ØØ3,Ø32,146,Ø31,Ø56,165,ØØ8
8Ø29 :155,237,131,Ø36,133,167,184
8Ø35 :165,156,237,132,Ø36,ØØ5,Ø62
8Ø41 :167,24Ø,Ø35,144,Ø33,169,125
8Ø47 :ØØØ,141,Ø83,Ø37,141,Ø92,Ø93
8Ø53 :Ø37,Ø32,146,Ø31,Ø32,225,1Ø8
8Ø59 :255,24Ø,251,169,Ø13,Ø32,Ø59
8Ø65 :21Ø,255,Ø32,2Ø4,255,169,23Ø
8Ø71 :ØØ1,Ø32,195,255,Ø32,187,Ø69
8Ø77 :Ø19,Ø96,Ø76,239,Ø3Ø,Ø56,145
8Ø83 :173,Ø91,Ø37,237,Ø98,Ø37,Ø52
8Ø89 :168,136,136,24Ø,Ø1Ø,Ø48,123
8Ø95 :ØØ8,169,Ø13,Ø32,21Ø,255,Ø78
81Ø1 :136,2Ø8,248,173,Ø84,Ø37,Ø27
81Ø7 :24Ø,Ø19,141,129,Ø36,173,141
8113 :Ø87,Ø37,133,169,173,Ø88,Ø96
8119 :Ø37,133,17Ø,Ø32,Ø36,Ø32,111
8125 :Ø32,1Ø5,Ø34,169,Ø13,Ø32,Ø62
8131 :21Ø,255,Ø32,21Ø,255,Ø32,165
8137 :21Ø,255,173,Ø95,Ø37,2Ø8,155
8143 :Ø26,Ø32,2Ø4,255,Ø32,2Ø3,191
8149 :Ø19,169,Ø84,16Ø,Ø36,Ø32,2Ø1
8155 :Ø97,Ø19,Ø32,228,255,24Ø,Ø66
8161 :251,Ø32,Ø86,Ø3Ø,162,ØØ1,Ø19
8167 :Ø32,2Ø1,255,238,Ø97,Ø37,Ø67
8173 :173,Ø83,Ø37,24Ø,Ø19,141,162
8179 :129,Ø36,173,Ø85,Ø37,133,Ø68
8185 :169,173,Ø86,Ø37,133,17Ø,249
8191 :Ø32,Ø36,Ø32,Ø32,1Ø5,Ø34,Ø14
8197 :169,Ø13,Ø32,21Ø,255,172,Ø88
82Ø3 :Ø92,Ø37,14Ø,Ø98,Ø37,136,Ø39
82Ø9 :136,24Ø,Ø15,Ø48,Ø13,169,126
8215 :Ø13,Ø32,21Ø,255,Ø32,225,Ø22
8221 :255,24Ø,Ø18,136,2Ø8,243,1Ø5
8227 :Ø96,169,Ø32,172,Ø89,Ø37,118
8233 :14Ø,Ø96,Ø37,Ø32,21Ø,255,Ø43
8239 :136,2Ø8,25Ø,Ø96,172,Ø94,235
8245 :Ø37,Ø24,152,1Ø9,Ø98,Ø37,254
8251 :141,Ø98,Ø37,169,Ø13,Ø32,Ø37
8257 :21Ø,255,136,2Ø8,25Ø,Ø96,196
8263 :141,1ØØ,Ø37,Ø41,127,Ø32,Ø37
8269 :Ø51,Ø3Ø,2Ø1,Ø49,144,ØØ7,Ø47
8275 :2Ø1,Ø58,176,ØØ3,Ø76,Ø73,158
8281 :Ø33,174,16Ø,Ø32,221,16Ø,1Ø1
8287 :Ø32,24Ø,Ø12,2Ø2,2Ø8,248,Ø13
8293 :2Ø6,Ø96,Ø37,173,1ØØ,Ø37,238
8299 :Ø76,248,Ø3Ø,2Ø2,138,Ø1Ø,Ø43
83Ø5 :17Ø,14Ø,Ø99,Ø37,169,Ø32,248
8311 :Ø72,169,132,Ø72,189,172,157
8317 :Ø32,Ø72,189,171,Ø32,Ø72,181
8323 :Ø96,Ø56,173,Ø99,Ø37,1Ø1,181
8329 :155,133,155,165,156,1Ø5,238
8335 :ØØØ,133,156,Ø76,239,Ø3Ø,ØØ9
8341 :177,155,2Ø1,Ø31,24Ø,ØØ1,186
8347 :136,14Ø,Ø99,Ø37,Ø96,Ø1Ø,161
8353 :Ø87,Ø65,Ø76,Ø82,Ø84,Ø66,1Ø9
8359 :Ø83,Ø78,Ø72,Ø7Ø,19Ø,Ø32,18Ø
8365 :199,Ø32,2Ø8,Ø32,218,Ø32,126
8371 :228,Ø32,238,Ø32,248,Ø32,221
8377 :ØØ2,Ø33,ØØ5,Ø33,Ø43,Ø33,Ø78
8383 :169,ØØØ,141,Ø95,Ø37,2ØØ,Ø65
8389 :Ø76,149,Ø32,2ØØ,169,ØØ1,Ø56
8395 :141,1Ø3,Ø37,Ø76,149,Ø32,229
84Ø1 :2ØØ,Ø32,Ø24,Ø29,141,Ø89,212
84Ø7 :Ø37,Ø76,149,Ø32,2ØØ,Ø32,229
8413 :Ø24,Ø29,141,Ø9Ø,Ø37,Ø76,1Ø6
8419 :149,Ø32,2ØØ,Ø32,Ø24,Ø29,181
8425 :141,Ø92,Ø37,Ø76,149,Ø32,248
8431 :2ØØ,Ø32,Ø24,Ø29,141,Ø93,246
8437 :Ø37,Ø76,149,Ø32,2ØØ,Ø32,ØØ3
8443 :Ø24,Ø29,141,Ø94,Ø37,Ø76,14Ø
8449 :149,Ø32,Ø76,146,Ø31,Ø56,235
8455 :152,1Ø1,155,141,Ø85,Ø37,166
8461 :165,156,1Ø5,ØØØ,141,Ø86,154
8467 :Ø37,Ø32,Ø35,Ø33,Ø56,152,1Ø8
8473 :237,Ø99,Ø37,141,Ø83,Ø37,147
8479 :2ØØ,Ø76,149,Ø32,2ØØ,177,Ø97
8485 :155,2Ø1,Ø31,2Ø8,249,136,249
8491 :Ø96,Ø56,152,1Ø1,155,141,232
8497 :Ø87,Ø37,165,156,1Ø5,ØØØ,Ø87
85Ø3 :141,Ø88,Ø37,Ø32,Ø35,Ø33,165
85Ø9 :Ø56,152,237,Ø99,Ø37,141,Ø15
8515 :Ø84,Ø37,2ØØ,Ø76,149,Ø32,133
8521 :2ØØ,177,155,2Ø1,Ø61,24Ø,Ø83
8527 :ØØ4,136,Ø76,1Ø4,Ø32,2ØØ,119
8533 :Ø32,Ø24,Ø29,Ø72,173,1ØØ,ØØ3
8539 :Ø37,Ø41,Ø15,17Ø,2Ø2,1Ø4,148
8545 :157,Ø77,Ø3Ø,Ø32,149,Ø32,Ø62
8551 :Ø76,132,Ø32,Ø32,231,255,Ø93
8557 :169,ØØØ,Ø32,189,255,169,155
8563 :Ø15,162,ØØ8,16Ø,Ø15,Ø32,251
8569 :186,255,Ø32,192,255,144,161
8575 :ØØ1,Ø96,Ø32,2Ø3,Ø19,169,135
8581 :Ø62,Ø32,21Ø,255,Ø32,134,Ø9Ø
8587 :Ø26,24Ø,Ø25,162,Ø15,Ø32,127
```

```
8593 :201,255,176,012,169,179,113
8599 :160,036,032,097,019,169,152
8605 :013,032,210,255,032,231,162
8611 :255,076,187,019,032,231,195
8617 :255,169,000,032,189,255,045
8623 :169,015,162,008,160,015,192
8629 :032,186,255,032,192,255,109
8635 :176,228,032,203,019,162,239
8641 :015,032,198,255,032,134,091
8647 :026,032,231,255,169,001,145
8653 :141,127,036,096,173,141,151
8659 :002,201,005,240,005,173,069
8665 :101,037,208,040,032,203,070
8671 :019,169,105,160,036,032,232
8677 :097,019,032,134,026,208,233
8683 :004,032,187,019,096,169,230
8689 :001,141,101,037,141,127,021
8695 :036,173,019,018,133,155,013
8701 :173,020,018,133,156,076,061
8707 :018,034,165,038,133,155,034
8713 :165,039,133,156,160,001,151
8719 :076,020,034,160,000,162,211
8725 :000,189,179,036,032,235,180
8731 :019,209,155,240,002,162,046
8737 :255,200,208,011,230,156,069
8743 :165,156,205,132,036,240,205
8749 :002,176,035,232,236,134,092
8755 :036,208,224,024,152,101,028
8761 :155,133,038,165,156,105,041
8767 :000,133,039,056,165,038,238
8773 :237,134,036,133,038,165,044
8779 :039,233,000,133,039,076,083
8785 :067,021,032,203,019,169,080
8791 :115,160,036,032,097,019,034
8797 :169,001,141,127,036,169,224
8803 :000,141,101,037,096,096,058
8809 :160,000,204,129,036,240,106
8815 :248,177,169,048,038,032,055
8821 :051,030,032,015,035,032,056
8827 :210,255,173,104,037,240,118
8833 :010,169,008,032,210,255,045
8839 :169,095,032,210,255,032,160
8845 :225,255,208,005,104,104,018
8851 :076,121,031,200,076,107,246
8857 :034,140,099,037,041,127,119
8863 :032,051,030,201,049,144,154
8869 :017,201,058,176,013,041,159
8875 :015,170,202,189,077,030,086
8881 :032,210,255,076,150,034,166
8887 :201,067,208,026,056,169,142
8893 :080,237,129,036,074,056,033
8899 :237,089,037,168,169,032,159
8905 :032,210,255,136,208,250,012
8911 :172,099,037,076,150,034,007
8917 :201,069,208,017,056,173,169
8923 :090,037,237,129,036,056,036
8929 :237,089,037,168,169,032,189
8935 :076,201,034,201,085,208,012
8941 :008,173,104,037,073,001,121
8947 :141,104,037,201,035,240,233
8953 :003,076,150,034,174,097,015
8959 :037,169,000,032,205,221,151
8965 :160,054,132,001,172,099,111
8971 :037,076,150,034,174,103,073
8977 :037,240,026,133,167,041,149
8983 :127,201,065,144,018,201,011
8989 :091,176,014,170,165,167,044
```

```
8995 :041,128,073,128,074,074,041
9001 :133,167,138,005,167,096,235
9007 :032,203,019,056,173,021,039
9013 :018,237,131,036,170,173,050
9019 :022,018,237,132,036,032,024
9025 :205,221,169,001,141,127,161
9031 :036,096,014,008,144,211,068
9037 :080,069,069,068,211,067,129
9043 :082,073,080,084,000,194,084
9049 :085,070,070,069,082,032,241
9055 :195,076,069,065,082,069,139
9061 :068,000,194,085,070,070,076
9067 :069,082,032,198,085,076,137
9073 :076,000,196,069,076,069,087
9079 :084,069,032,040,211,044,087
9085 :215,044,208,041,000,058,179
9091 :211,085,082,069,063,032,161
9097 :217,047,206,000,197,210,246
9103 :193,211,197,032,212,197,161
9109 :216,212,000,197,082,065,153
9115 :083,069,032,040,211,044,122
9121 :215,044,208,041,058,060,019
9127 :210,197,212,213,210,206,135
9133 :062,000,203,069,089,058,142
9139 :000,211,065,086,069,058,156
9145 :000,212,065,080,069,032,131
9151 :197,210,210,207,210,000,201
9157 :211,084,079,080,080,069,032
9163 :068,000,214,069,082,073,197
9169 :070,089,032,197,082,082,249
9175 :079,082,000,206,079,032,181
9181 :069,082,082,079,082,083,186
9187 :000,147,032,018,212,146,014
9193 :065,080,069,032,079,082,128
9199 :032,018,196,146,073,083,019
9205 :075,063,000,204,079,065,219
9211 :068,058,000,214,069,082,230
9217 :073,070,089,058,000,208,243
9223 :082,069,083,083,032,018,118
9229 :210,197,212,213,210,206,237
9235 :146,000,036,048,206,079,022
9241 :032,210,079,079,077,000,246
9247 :206,079,032,084,069,088,077
9253 :084,032,073,078,032,066,146
9259 :085,070,070,069,082,046,209
9265 :000,196,069,086,073,067,028
9271 :069,032,035,000,211,069,215
9277 :067,079,078,068,046,032,175
9283 :193,068,068,082,046,032,044
9289 :035,000,208,082,073,078,037
9295 :084,073,078,071,000,206,079
9301 :069,088,084,032,083,072,001
9307 :069,069,084,044,032,146,023
9313 :210,197,212,213,210,206,065
9319 :018,000,200,085,078,084,056
9325 :032,070,079,082,058,000,174
9331 :206,079,084,032,198,079,025
9337 :085,078,068,000,000,000,096
```

BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs," "A Beginner's Guide To Typing In Programs," and "The Automatic Proofreader" that appear before the Program Listings.

MLX For VIC And 64

(Article on page 171.)

Program 1: MLX—64 Version

```
100 PRINT"{CLR}{CYN}";CHR$(142);CHR$(8);:
    POKE53281,1:POKE53280,1          :rem 73
101 POKE 788,52:REM DISABLE RUN/STOP
                                    :rem 119
110 PRINT"{RVS}{40 SPACES}";        :rem 176
120 PRINT"{RVS}{15 SPACES}{RIGHT}{OFF}
    {*}£{RVS}{RIGHT} {RIGHT}{2 SPACES}
    {*}{OFF}{*}£{RVS}£{RVS}
    {13 SPACES}{^}";                :rem 250
130 PRINT"{RVS}{15 SPACES}{RIGHT} {G}
    {RIGHT} {2 RIGHT} {OFF}£{RVS}£{*}
    {OFF}{*}{RVS}{13 SPACES}{T}";    :rem 35
140 PRINT"{RVS}{40 SPACES}"         :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK}{3 SPACES}A F
    AILSAFE MACHINE LANGUAGE EDITOR
    {5 DOWN}"                       :rem 130
210 PRINT"{5}{2 UP}STARTING ADDRESS?
    {8 SPACES}{9 LEFT}";            :rem 143
215 INPUTS:F=1-F:C$=CHR$(31+119*F:rem 125
220 IFS<256OR(S>40960ANDS<49152)ORS>53247
    THENGOSUB3000:GOTO210           :rem 235
225 PRINT:PRINT:PRINT               :rem 180
230 PRINT"{5}{2 UP}ENDING ADDRESS?
    {8 SPACES}{9 LEFT}";:INPUTE:F=1-F:C$=
    CHR$(31+119*F)                   :rem 20
240 IFE<256OR(E>40960ANDE<49152)ORE>53247
    THENGOSUB3000:GOTO230           :rem 183
250 IFE<STHENPRINTC$;"{RVS}ENDING < START
    {2 SPACES}":GOSUB1000:GOTO 230
                                    :rem 176
260 PRINT:PRINT:PRINT               :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S:POKEV+21,0
                                    :rem 225
310 PRINTRIGHT$("0000"+MID$(STR$(AD),2),5
    );":";:FORJ=1TO6                :rem 234
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                    :rem 228
390 IFN=-211THEN 710                 :rem 62
400 IFN=-204THEN 790                 :rem 64
410 IFN=-206THENPRINT:INPUT"{DOWN}ENTER N
    EW ADDRESS";ZZ                   :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}OUT OF RANGE":GOSUB1000:GOTO410
                                    :rem 225
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                    :rem 238
420 IF N<>-196 THEN 480             :rem 133
430 PRINT:INPUT"DISPLAY:FROM";F:PRINT,"TO
    ";:INPUTT                       :rem 234
440 IFF<SORF>EORT<SORT>ETHENPRINT"AT LEAS
    T";S;"{LEFT}, NOT MORE THAN";E:GOTO43
    0                               :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":";       :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHT$("00
    "+MID$(STR$(N),2),3);",";        :rem 66
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                                     :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
    T:GOTO310                        :rem 50
480 IFN<0 THEN PRINT:GOTO310        :rem 168
490 A(J)=N:NEXTJ                    :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT     :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20)
                                    :rem 234
515 IFN=CKSUMTHEN530                :rem 255
520 PRINT:PRINT"LINE ENTERED WRONG : RE-E
    NTER":PRINT:GOSUB1000:GOTO310:rem 176
530 GOSUB2000                       :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54
    272,0:POKE54273,0               :rem 227
550 AD=AD+6:IF AD<E THEN 310        :rem 212
560 GOTO 710                        :rem 108
570 N=0:Z=0                          :rem 88
580 PRINT"{+}";                      :rem 79
581 GETA$:IFA$=""THEN581             :rem 95
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN670                   :rem 229
590 IFA>128THENN=-A:RETURN          :rem 137
600 IFA<>20 THEN 630                 :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
    {LEFT} {LEFT}";:GOTO690         :rem 172
620 GOTO570                         :rem 109
630 IFA<48ORA>57THEN580             :rem 105
640 PRINTA$;:N=N*10+A-48            :rem 106
650 IFN>255 THEN A=20:GOSUB1000:GOTO600
                                    :rem 229
660 Z=Z+1:IFZ<3THEN580               :rem 71
670 IFZ=0THENGOSUB1000:GOTO570      :rem 114
680 PRINT",";:RETURN                :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                    :rem 149
691 FORI=1TO3:T=PEEK(S%-I)           :rem 67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                    :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1);:RETURN
                                      :rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***{3 DOWN}"
                                    :rem 236
720 INPUT"{DOWN} FILENAME";F$       :rem 228
730 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
    {RVS}D{OFF}ISK: (T/D)"          :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"THEN740
                                     :rem 36
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                    :rem 158
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                  :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                    :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                               :rem 69
765 POKE254,S/256:POKE253,S-PEEK(254)*256
    :POKE780,253                     :rem 12
766 POKE782,E/256:POKE781,E-PEEK(782)*256
    :SYS65496                       :rem 124
770 IF(PEEK(783)AND1)OR(ST AND191)THEN780
                                    :rem 111
775 PRINT"{DOWN}DONE.":END          :rem 106
780 PRINT"{DOWN}ERROR ON SAVE.{2 SPACES}T
    RY AGAIN.":IFDV=1THEN720        :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO720            :rem 103
790 PRINT"{CLR}{RVS}*** LOAD ***{2 DOWN}"
                                    :rem 212
800 INPUT"{2 DOWN} FILENAME";F$     :rem 244
810 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
    {RVS}D{OFF}ISK: (T/D)"          :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"THEN820
                                     :rem 34
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                    :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                  :rem 2
```

```
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                     :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                               :rem 70
850 POKE780,0:SYS65493               :rem 11
860 IF(PEEK(783)AND1)OR(ST AND191)THEN870
                                     :rem 111
865 PRINT"{DOWN}DONE.":GOTO310       :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.{2 SPACES}T
    RY AGAIN.{DOWN}":IFDV=1THEN800
                                     :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO800             :rem 102
1000 REM BUZZER                      :rem 135
1001 POKE54296,15:POKE54277,45:POKE54278,
     165                             :rem 207
1002 POKE54276,33:POKE 54273,6:POKE54272,
     5                               :rem 42
1003 FORT=1TO200:NEXT:POKE54276,32:POKE54
     273,0:POKE54272,0:RETURN        :rem 202
2000 REM BELL SOUND                  :rem 78
2001 POKE54296,15:POKE54277,0:POKE54278,2
     47                              :rem 152
2002 POKE 54276,17:POKE54273,40:POKE54272
     ,0                              :rem 86
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN
                                     :rem 57
3000 PRINTC$;"{RVS}NOT ZERO PAGE OR ROM":
     GOTO1000                        :rem 89
```

Program 2: MLX—VIC Version

```
100 PRINT"{CLR}{PUR}";CHR$(142);CHR$(8);
                                     :rem 181
101 POKE 788,194:REM DISABLE RUN/STOP
                                     :rem 174
110 PRINT"{RVS}{14 SPACES}"          :rem 117
120 PRINT"{RVS} {RIGHT}?{OFF}[*]£{RVS}
    {RIGHT} {RIGHT}{2 SPACES}[*]{OFF}
    [*]£{RVS}£{RVS} "                :rem 191
130 PRINT"{RVS} {RIGHT} [G]{RIGHT}
    {2 RIGHT} {OFF}£{RVS}£[*]{OFF}
    [*]{RVS} "                       :rem 232
140 PRINT"{RVS}{14 SPACES}"          :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK}A FAILSAFE MA
    CHINE":PRINT"LANGUAGE EDITOR{5 DOWN}"
                                     :rem 141
210 PRINT"{BLK}{3 UP}STARTING ADDRESS":IN
    PUTS:F=1-F:C$=CHR$(31+119*F)     :rem 97
220 IFS<256ORS>32767THENGOSUB3000:GOTO210
                                     :rem 2
225 PRINT:PRINT:PRINT:PRINT          :rem 123
230 PRINT"{BLK}{3 UP}ENDING ADDRESS":INPU
    TE:F=1-F:C$=CHR$(31+119*F)       :rem 158
240 IFE<256ORE>32767THENGOSUB3000:GOTO230
                                     :rem 234
250 IFE<STHENPRINTC$;"{RVS}ENDING < START
    {2 SPACES}":GOSUB1000:GOTO 230
                                     :rem 176
260 PRINT:PRINT:PRINT                :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S       :rem 56
310 PRINTRIGHT$("0000"+MID$(STR$(AD),2),5
    );":";:FORJ=1TO6                 :rem 234
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                     :rem 228
390 IFN=-211THEN 710                 :rem 62
400 IFN=-204THEN 790                 :rem 64
410 IFN=-206THENPRINT:INPUT"{DOWN}ENTER N
    EW ADDRESS";ZZ                   :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}OUT OF RANGE":GOSUB1000:GOTO410
                                     :rem 225
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                     :rem 238
420 IF N<>-196 THEN 480              :rem 133
430 PRINT:INPUT"DISPLAY:FROM";F:PRINT,"TO
    ";:INPUTT                        :rem 234
440 IFF<SORF>EORT<SORT>ETHENPRINT"AT LEAS
    T";S;"{LEFT}, NOT MORE THAN";E:GOTO43
    0                                :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":";       :rem 30
455 FORK=0TO5:N=PEEK(I+K):IFK=3THENPRINTS
    PC(10);                          :rem 34
457 PRINTRIGHT$("00"+MID$(STR$(N),2),3);"
    ,";                              :rem 157
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                                     :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
    T:GOTO310                        :rem 50
480 IFN<0 THEN PRINT:GOTO310         :rem 168
490 A(J)=N:NEXTJ                     :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT      :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20)
                                     :rem 234
515 IFN=CKSUMTHEN530                 :rem 255
520 PRINT:PRINT"LINE ENTERED WRONG":PRINT
    "RE-ENTER":PRINT:GOSUB1000:GOTO310
                                     :rem 129
530 GOSUB2000                        :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:rem 80
550 AD=AD+6:IF AD<E THEN 310         :rem 212
560 GOTO 710                         :rem 108
570 N=0:Z=0                          :rem 88
580 PRINT"[+]";                      :rem 79
581 GETA$:IFA$=""THEN581             :rem 95
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN670                    :rem 229
590 IFA>128THENN=-A:RETURN           :rem 137
600 IFA<>20 THEN 630                 :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
    {LEFT} {LEFT}";:GOTO690          :rem 172
620 GOTO570                          :rem 109
630 IFA<48ORA>57THEN580              :rem 105
640 PRINTA$;:N=N*10+A-48             :rem 106
650 IFN>255 THEN A=20:GOSUB1000:GOTO600
                                     :rem 229
660 Z=Z+1:IFZ<3THEN580               :rem 71
670 IFZ=0THENGOSUB1000:GOTO570       :rem 114
680 PRINT",";:RETURN                 :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                     :rem 149
692 FORI=1TO3:T=PEEK(S%-I)           :rem 68
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                     :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1);:RETURN
                                     :rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***{3 DOWN}"
                                     :rem 236
720 INPUT"{DOWN} FILENAME";F$        :rem 228
730 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
     {RVS}D{OFF}ISK: (T/D)"          :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"THEN740
                                     :rem 36
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                     :rem 158
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                 :rem 3
```

```
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                    :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                               :rem 69
765 POKE254,S/256:POKE253,S-PEEK(254)*256
    :POKE780,253                     :rem 12
766 POKE782,E/256:POKE781,E-PEEK(782)*256
    :SYS65496                       :rem 124
770 IF(PEEK(783)AND1)OR(ST AND191)THEN780
                                    :rem 111
775 PRINT"{DOWN}DONE.":END          :rem 106
780 PRINT"{DOWN}ERROR ON SAVE.{2 SPACES}T
    RY AGAIN.":IFDV=1THEN720        :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO720            :rem 103
782 GOTO720                         :rem 115
790 PRINT"{CLR}{RVS}*** LOAD ***{2 DOWN}"
                                    :rem 212
800 INPUT"{2 DOWN} FILENAME";F$     :rem 244
810 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
     {RVS}D{OFF}ISK: (T/D)"         :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"THEN820
                                     :rem 34
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                    :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                  :rem 2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                    :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                               :rem 70
850 POKE780,0:SYS65493               :rem 11
860 IF(PEEK(783)AND1)OR(ST AND191)THEN870
                                    :rem 111
865 PRINT"{DOWN}DONE.":GOTO310       :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.{2 SPACES}T
    RY AGAIN.{DOWN}":IFDV=1THEN800
                                    :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO800            :rem 102
1000 REM BUZZER                     :rem 135
1001 POKE36878,15:POKE36874,190     :rem 206
1002 FORW=1TO300:NEXTW              :rem 117
1003 POKE36878,0:POKE36874,0:RETURN
                                     :rem 74
2000 REM BELL SOUND                  :rem 78
2001 FORW=15TO0STEP-1:POKE36878,W:POKE368
     76,240:NEXTW                    :rem 22
2002 POKE36876,0:RETURN             :rem 119
3000 PRINTC$;"{RVS}NOT ZERO PAGE OR ROM":
     GOTO1000                        :rem 89
```

64 Electronic Notepad

(Article on page 112.)

BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs," "A Beginner's Guide To Typing In Programs," and "The Automatic Proofreader" that appear before the Program Listings.

```
1 BO=254:SC=246:CH=14               :rem 170
5 POKE53280,BO:POKE53281,SC:POKE646,CH
                                     :rem 17
7 PRINT"{CLR}";TAB(10);"{RVS} ELECTRONIC
  {SPACE}NOTEPAD ",TAB(6)"{DOWN} LOADING
  {SPACE}DATA...PLEASE WAIT"         :rem 83
10 B=885:C=998:FORA=BTOC:READD:POKEA,D:NE
   XT                               :rem 221
12 B=49152:C=49407:FORA=BTOC:READD:POKEA,
   D:NEXT                           :rem 157
14 B=679:C=753:FORA=BTOC:READD:POKEA,D:NE
   XT                               :rem 215
16 PRINT"{CLR}{DOWN} ENTER 2 SECRET CODES
    (0-255):":PRINT"{DOWN} (SEPERATE EACH
    BY A COMMA)"                    :rem 100
17 PRINT"{DOWN} (ENTER 0'S IF NO SECRET C
   ODE){2 DOWN}":POKE646,PEEK(53281):INPU
   TA,B                              :rem 25
19 POKE646,CH:IFA<0ORA>255ORB<0ORB>255THE
   N16                              :rem 196
20 POKE249,A:POKE250,B:A=0:B=0:SYS679
                                    :rem 187
22 PRINT"{CLR}{DOWN} {RVS}D{OFF}ISK OR
   {RVS}T{OFF}APE?"                  :rem 86
23 GETA$:IFA$=""THEN23              :rem 237
24 IFA$="D"THEN30                   :rem 177
25 IFA$="T"THEN29                   :rem 202
26 GOTO22                             :rem 5
29 POKE49303,1:POKE49305,1:POKE49307,255:
   POKE49177,1:POKE49179,1          :rem 163
30 PRINT"{CLR}";TAB(10);"{RVS} ELECTRONIC
    NOTEPAD "                        :rem 15
40 PRINT"{2 DOWN}{3 SPACES}FUNCTION";TAB(
   30);"PRESS"                      :rem 143
45 PRINT"{3 SPACES}{8 T}";TAB(30);"
   {5 T}"                           :rem 198
50 PRINT"{2 DOWN} VIEW NOTEPAD PAGE";TAB(
   31);"F1"                         :rem 120
55 PRINT"{DOWN} CREATE NOTEPAD PAGE";TAB(
   31);"F3"                         :rem 231
60 PRINT"{DOWN} VIEW DISK DIRECTORY";TAB(
   31);"F5"                          :rem 36
65 PRINT"{DOWN} CHANGE PROGRAM OPTIONS";T
   AB(31);"F6"                      :rem 249
70 PRINT"{DOWN} END PROGRAM";TAB(31);"F8"
                                    :rem 252
75 GETA$:IFA$="{F1}"THEN200         :rem 166
80 IFA$="{F3}"THEN600                :rem 40
85 IFA$="{F5}"THEN400                :rem 44
86 IFA$="{F8}"THENPRINT"{CLR}":CLR:POKE24
   9,0:POKE250,0:END                :rem 238
87 IFA$="{F6}"THENRESTORE:GOTO1     :rem 102
90 GOTO75                            :rem 14
200 PRINT"{CLR}"                    :rem 246
210 SYS49152                        :rem 151
215 GOSUB500                        :rem 173
220 GETA$:IFA$=""ORA$="←"THEN220     :rem 49
230 GOTO30                           :rem 49
400 PRINT"{CLR}"                    :rem 248
405 IFPEEK(49303)=1THEN450          :rem 215
410 SYS885                           :rem 57
420 PRINTTAB(7);"{RVS}(PRESS ANY KEY)"
                                    :rem 194
430 GETA$:IFA$=""THEN430             :rem 81
440 GOTO30                           :rem 52
450 PRINT"{DOWN} NO DIRECTORY AVAILABLE":
    PRINT"{DOWN} PROGRAM IN {RVS}TAPE
    {OFF} MODE"                      :rem 12
460 PRINT"{2 DOWN} {RVS} (PRESS ANY KEY)
    {SPACE}"                         :rem 78
470 GETA$:IFA$=""THEN470             :rem 89
480 GOTO30                           :rem 56
500 CLOSE15:OPEN15,8,15:INPUT#15,A,B$,C,D
    :IFA>21THEN510                  :rem 218
505 RETURN                          :rem 122
510 PRINT:PRINT"{DOWN}{3 SPACES}{RVS} *DI
    SK ERROR* ":PRINT"{DOWN}{3 SPACES}
    {RVS} ";B$                       :rem 41
```

```
515 RETURN                              :rem 123
600 PRINT"{CLR}";TAB(9);"{RVS} CREATE NOT
    EPAD PAGE "                          :rem 3
605 PRINT"{2 DOWN} PRESS {RVS}F1{OFF} TO
    {SPACE}SAVE PAGE."                  :rem 83
610 PRINT"{DOWN} PRESS {RVS}F8{OFF} TO AB
    ORT PAGE."                          :rem 142
615 PRINT"{2 DOWN} (PRESS ANY KEY)"
                                        :rem 62
620 GETA$:IFA$=""THEN620                :rem 83
622 PRINT"{CLR}";                       :rem 57
625 GETA$:IFA$=""THENPRINT"{P} {2 LEFT}
    ";:GOTO625                          :rem 197
630 IFA$="{LEFT}"THEN625                :rem 119
635 IFA$="{RIGHT}"THEN625               :rem 252
636 IFA$="{HOME}"THENGOTO625            :rem 44
637 IFA$=CHR$(34)THEN625                :rem 86
638 IFA$="←"THENPRINT" {LEFT}";:GOTO625
                                        :rem 93
640 IFA$="{UP}"THENPRINT" {LEFT}{UP}";:GO
    TO625                               :rem 25
645 IFA$="{DOWN}"THENPRINT" {LEFT}{DOWN}"
    ;:GOTO625                           :rem 30
646 IFA$=CHR$(13)THENPRINT" ";CHR$(13);:G
    OTO625                              :rem 195
650 IFA$="{F1}"THEN680                  :rem 98
655 IFA$="{F8}"THEN30                   :rem 51
675 PRINTA$;" {LEFT}";:GOTO625          :rem 251
680 PRINT" ";:POKE648,60:POKE53272,245:SY
    S49278                              :rem 193
685 GOSUB500:IFA<21THENPOKE648,4:POKE5327
    2,21:GOTO30                         :rem 146
690 PRINT"{DOWN} {RVS} (PRESS ANY KEY) "
                                        :rem 66

695 GETA$:IFA$=""THEN695                :rem 107
697 POKE648,4:POKE53272,21:SYS49374:GOTO6
    25                                  :rem 38
885 DATA 169,001,162,008,160,000        :rem 36
891 DATA 032,186,255,169,002,162        :rem 51
897 DATA 224,160,003,032,189,255        :rem 51
903 DATA 032,192,255,162,001,032        :rem 30
909 DATA 198,255,032,207,255,032        :rem 53
915 DATA 207,255,032,207,255,032        :rem 41
921 DATA 207,255,240,058,032,204        :rem 37
927 DATA 255,032,228,255,201,032        :rem 41
933 DATA 208,005,032,228,255,240        :rem 39
939 DATA 251,162,001,032,198,255        :rem 48
945 DATA 032,207,255,072,032,207        :rem 41
951 DATA 255,168,104,170,152,032        :rem 42
957 DATA 205,189,169,032,032,210        :rem 49
963 DATA 255,032,207,255,240,006        :rem 42
969 DATA 032,210,255,076,196,003        :rem 50
975 DATA 169,013,032,210,255,076        :rem 48
981 DATA 149,003,169,001,032,195        :rem 46
987 DATA 255,032,204,255,096,036        :rem 57
993 DATA 048,013,013,013,013,013        :rem 27
49152 DATA32,54,192, 160, 0, 162, 0, 32
                                        :rem 178
49160 DATA207,255,201,13,240,8,157, 240
                                        :rem 127
49168 DATA193,232,200,76,7,192, 152, 72
                                        :rem 97
49176 DATA169,8,162,8, 160, 1, 32, 186
                                        :rem 206
49184 DATA255,104,162,240,160,193,32,189
                                        :rem 243
49192 DATA255,169,0,162, 0, 160, 4, 32
                                        :rem 188
49200 DATA213,255,96,234,234,234,162, 0
                                        :rem 130
49208 DATA173,134,2,157,0,216,232, 208
                                        :rem 81
49216 DATA250,238,61,192,172,61,192,192
                                        :rem 196
49224 DATA220,208,235,169,216,141,61,192
                                        :rem 237
49232 DATA162,0,189,94,192,32,210, 255
                                        :rem 90
49240 DATA232,224,29,208,245,96,13, 17
                                        :rem 91
49248 DATA69,78,84,69, 82, 32, 78, 79
                                        :rem 192
49256 DATA84,69,80, 65, 68, 32, 80, 65
                                        :rem 172
49264 DATA71,69,32, 78, 65, 77, 69, 32
                                        :rem 175
49272 DATA32,32,13,234,234,234, 32, 187
                                        :rem 84
49280 DATA192,160,0,162,0, 32, 207, 255
                                        :rem 27
49288 DATA201,13,240,8,157,240,194, 232
                                        :rem 138
49296 DATA200,76,133,192,152,72,169, 8
                                        :rem 103
49304 DATA162,8,160,0,32,186,255,104
                                        :rem 31
49312 DATA162,240,160,194,32,189,255,169
                                        :rem 248
49320 DATA0,133,251,169,4,133,252, 169
                                        :rem 83
49328 DATA251,162,255,160,7,32,216, 255
                                        :rem 141
49336 DATA234,234,234,162,0,189,201,192
                                        :rem 188

49344 DATA32,210,255,232,224,18,208,245
                                        :rem 183
49352 DATA96,147,13, 69, 78, 84, 69, 82
                                        :rem 231
49360 DATA32,70,73, 76, 69, 32, 78, 65
                                        :rem 165
49368 DATA77,69,13,234,234,0, 173, 134
                                        :rem 50
49376 DATA2,162,0,157,0, 216, 232, 208
                                        :rem 237
49384 DATA250,172,229,192,192,219,240,7
                                        :rem 201
49392 DATA200,140,229,192,76,222,192,169
                                        :rem 248
49400 DATA216,141,229,192,96,234,234,0
                                        :rem 136
60679 DATA120,169,188,141,20, 3, 169, 2
                                        :rem 43
60687 DATA141,21,3,88,169, 0, 133, 253
                                        :rem 246
60695 DATA169,4,133,254,96,165,197, 201
                                        :rem 157
60703 DATA57,240,3,76,49,234, 160, 0
                                        :rem 189
60711 DATA177,253,69,249,145,253,200,177
                                        :rem 249
60719 DATA253,69,250,145,253,200,234,208
                                        :rem 238
60727 DATA239,230,254,165,254,201,8,208
                                        :rem 190
60735 DATA229,169,4,133,254,160, 0, 162
                                        :rem 86
60743 DATA0,232,208,253,200,208,250, 76
                                        :rem 125
60751 DATA 49, 234, 0, 0, 0, 0, 0, 0
                                        :rem 119
```

Canyon Cruiser

(Article on page 96.)

Program 1:

Canyon Cruiser—64 Version

```
100 RESTORE                                :rem 181
110 GOTO150                                :rem 97
120 WX=INT(255-W):IFWX<160THENWX=160
                                           :rem 216
130 POKE53250,WX:POKE53251,ABS(W-25)
                                           :rem 68
140 POKE53286,7:POKE53286,2:RETURN:rem 24
150 PRINT"{CLR}":POKE53280,6:POKE53281,6
                                           :rem 149
160 FORW=ØTO1Ø:READR$(W):NEXTW             :rem 138
170 DATA"STARTER"                          :rem 27
180 DATA"BEGINNER"                         :rem 65
190 DATA"LEARNER"                          :rem 1
200 DATA"FLYER"                            :rem 114
210 DATA"LEADER"                           :rem 158
220 DATA"ACE"                              :rem 187
230 DATA"LIEUTENANT"                       :rem 236
240 DATA"GENERAL"                          :rem 242
250 DATA"COMMODORE"                        :rem 154
260 DATA"FLEET COMMANDER"                  :rem 252
270 DATA"HAN SOLO ?????????"               :rem 66
280 PRINT"{HOME}{5 DOWN}";SPC(15);"{WHT}6
    4-CANYON"                              :rem 142
290 PRINT                                  :rem 40
300 PRINT"{HOME}{9 DOWN}";SPC(11);"{RVS}I
    NSTRUCTIONS (Y/N)"                     :rem 81
310 FORW=1TO2ØØ:NEXTW                      :rem 69
320 PRINT"{HOME}{9 DOWN}";SPC(11);"INSTRU
    CTIONS (Y/N)"                          :rem 65
330 FORW=1TO2ØØ:NEXTW                      :rem 71
340 GETZ$:IFZ$=""THEN3ØØ                   :rem 127
350 POKE5328Ø,Ø:POKE53281,Ø                :rem 237
360 IFZ$="Y"THEN154Ø                       :rem 121
370 PRINT"{HOME}{5 DOWN}";SPC(15);"⁅4⁆6
    4-CANYON"                              :rem 32
380 PRINT                                  :rem 40
390 PRINT"{HOME}{9 DOWN}";SPC(11);"
    {18 SPACES}"                           :rem 108
400 FORW=1TO5Ø:P=INT(RND(TI)*1ØØØ)+1024:I
    FPEEK(P)=32THENPOKEP,46:POKEP+54272,1
                                           :rem 180
410 NEXTW                                  :rem 43
420 POKE53271,Ø:POKE53277,Ø                :rem 240
430 FORW=12288TO1235Ø:READN:POKEW,N:NEXTW
                                           :rem 99
440 FORW=12352TO12414:READN:POKEW,N:NEXTW
                                           :rem 93
450 POKE53276,PEEK(53276)OR2↑Ø+2↑1:REM SE
    T MULTI COLOR MODE                     :rem 254
460 POKE53285,8:POKE53286,2:POKE53287,6:R
    EM SET COLORS                          :rem 179
470 POKE53248,Ø:POKE53249,Ø                :rem 248
480 POKE53264,PEEK(53264)AND(255-(2↑Ø+2↑1
    ))                                     :rem 95
490 POKE2Ø4Ø,192:POKE2Ø41,193              :rem 83
500 POKE53269,PEEK(53269)OR(2↑Ø+2↑1)
                                           :rem 22
510 POKE53275,Ø                            :rem 39
520 FORW=2ØTO16ØSTEP2:POKE53248,W:POKE532
    49,W:GOSUB12Ø:NEXTW                    :rem 30
530 FORW=16ØTO18ØSTEP2:POKE53249,PEEK(532
    49)+3:GOSUB12Ø:NEXTW                   :rem 232
540 PRINT"{HOME}{19 DOWN}{YEL}{RVS}";SPC(
    13);"GET READY..."                     :rem 164
550 FORW=181TO255STEP2:L=PEEK(53249)+2:PO
    KE53249,(L)*((L>255)+1):GOSUB12Ø:NEXT
                                           :rem 70
560 FORW=235TO255:POKE53251,W:GOSUB14Ø:NE
    XTW                                    :rem 250
570 GOTO74Ø                                :rem 112
580 REM ** DATA FOR THE SHIP **            :rem 63
590 DATA 3, Ø, 192, 15, 195, 24Ø, 67, Ø,
    {SPACE}193                             :rem 236
600 DATA 17Ø, 17Ø, 17Ø, 171, 85, 234, 42,
    15Ø, 168                               :rem 18
610 DATA 42, 15Ø, 168, 26, 17Ø, 164, 34,
    {SPACE}17Ø, 136                        :rem 224
620 DATA 32, 17Ø, 8, 32, 4Ø, 8, 48, 4Ø, 1
    2                                      :rem 123
630 DATA Ø, 4Ø, Ø, Ø, 4Ø, Ø, Ø, 6Ø, Ø
                                           :rem 97
640 DATA Ø, 2Ø, Ø, Ø, 4Ø, Ø, Ø, Ø, Ø
                                           :rem 42
650 DATA Ø, Ø, Ø, Ø, Ø, Ø, Ø, Ø, Ø
                                           :rem 197
660 REM ** DATA FOR THE ASTEROID **
                                           :rem 101
670 DATA Ø, Ø, Ø, Ø, Ø, Ø, 1, Ø, Ø
                                           :rem 200
680 DATA 15, 92, 2Ø8, Ø, 253, 252, 63, 2Ø
    7, 253                                 :rem 130
690 DATA 63, 223, 255, 12, 252, 252, 63,
    {SPACE}245, 6Ø                         :rem 180
700 DATA 7, 127, 112, 15, 2Ø8, 245, 12, 1
    24, 127                                :rem 165
710 DATA 19, 255, 252, 55, 255, 22Ø, 63,
    {SPACE}197, 244                        :rem 239
720 DATA 13, 63, 2Ø4, 15, 31, 76, 3, 61,
    {SPACE}24Ø                             :rem 221
730 DATA Ø, 3, 192, Ø, Ø, Ø, Ø, Ø, Ø
                                           :rem 51
740 GOTO91Ø:REM ** RUN THE GAME **:rem 36
750 DATA"⁅6⁆{RVS} {OFF}⁅K⁆{3 SPACES}
    {RVS}⁅K⁆ {OFF}"                        :rem 29
760 DATA"⁅5⁆{RVS} {OFF}⁅K⁆{3 SPACES}
    {RVS}⁅K⁆ {OFF}"                        :rem 29
770 DATA"⁅4⁆{RVS} {OFF}⁅K⁆{4 SPACES}
    {RVS}⁅K⁆ {OFF}"                        :rem 29
780 DATA"⁅3⁆{RVS} {OFF}⁅K⁆{5 SPACES}
    {RVS}⁅K⁆ {OFF}"                        :rem 29
790 DATA"⁅2⁆{RVS} {OFF}⁅K⁆{6 SPACES}
    {RVS}⁅K⁆ {OFF}"                        :rem 29
800 DATA"⁅1⁆{RVS} {OFF}⁅K⁆{7 SPACES}
    {RVS}⁅K⁆ {OFF}"                        :rem 1
810 DATA"{YEL}{RVS} {OFF}⁅K⁆{8 SPACES}
    {RVS}⁅K⁆ {OFF}"                        :rem 31
820 DATA"{BLU}{RVS} {OFF}⁅K⁆{9 SPACES}
    {RVS}⁅K⁆ {OFF}"                        :rem 161
830 DATA"⁅7⁆{RVS} ⁅L⁆{OFF}{3 SPACES}
    {RVS}⁅K⁆ {OFF}"                        :rem 50
840 DATA"⁅5⁆{RVS} ⁅L⁆{OFF}{3 SPACES}
    {RVS}⁅K⁆ {OFF}"                        :rem 49
850 DATA"⁅4⁆{RVS} ⁅L⁆{OFF}{4 SPACES}
    {RVS}⁅K⁆ {OFF}"                        :rem 49
860 DATA"⁅3⁆{RVS} ⁅L⁆{OFF}{5 SPACES}
    {RVS}⁅K⁆ {OFF}"                        :rem 49
870 DATA"⁅2⁆{RVS} ⁅L⁆{OFF}{6 SPACES}
    {RVS}⁅K⁆ {OFF}"                        :rem 49
880 DATA"⁅1⁆{RVS} {OFF}⁅K⁆{7 SPACES}
    {RVS}⁅H⁆ {OFF}"                        :rem 28
890 DATA"{YEL}{RVS} {OFF}⁅K⁆{8 SPACES}
    {RVS}⁅H⁆ {OFF}"                        :rem 58
900 DATA"{BLU}{RVS} {OFF}⁅K⁆{9 SPACES}
    {RVS}⁅H⁆ {OFF}"                        :rem 179
```

```
910 FORW=53248TO53264:POKEW,Ø:NEXTW
                                   :rem 174
920 PRINT"{CLR}{2 DOWN}";SPC(15);"{RED}
    {RVS}[A]+++++[S]"         :rem 75
930 POKE53271,2:POKE53277,2        :rem 250
940 POKE53275,253                  :rem 152
950 FORW=1TO1Ø                      :rem 78
960 PRINTSPC(15);"{RED}{RVS}-[Q]{OFF}
    {3 SPACES}{RVS}[W]{RED}{RVS}-{OFF}"
                                   :rem 242
970 NEXTW                           :rem 54
980 PRINT"{RED}{RVS}{4 SPACES}{PUR}{OFF}T
    IME: 5{RED}{RVS}{2 SPACES}[A][R]+
    [X]{OFF}{3 SPACES}{RVS}[Z]+[R]
    [S]{16 SPACES}"              :rem 32
990 PRINT"{UP}";:FORW=1TO1Ø:PRINTSPC(12);
    "{RED}{RVS} [Q]{OFF}{9 SPACES}{RVS}
    [W] {OFF}":NEXTW              :rem 155
1000 POKE53248,160:POKE53249,74  :rem 192
1010 PRINT"{HOME}"                :rem 166
1020 TI$="ØØØØØ5"                  :rem 42
1030 PRINT"{HOME}{13 DOWN}{9 RIGHT}{PUR}"
     ;                             :rem 97
1040 A1=INT(1Ø-VAL(TI$)):PRINTA1 :rem 108
1050 IFA1<>ØTHEN1Ø3Ø              :rem 100
1060 LDP=53248                    :rem 234
1070 FORSYP=74TO140:POKE53249,SYP:GOSUB14
     Ø:NEXTSYP                    :rem 231
1080 PRINT"{HOME}{23 DOWN}";      :rem 111
1090 FORW=ØTO7:READC$(W):NEXTW    :rem 132
1100 FORW=ØTO7:READBC$(W):NEXTW   :rem 190
1110 TL=12:C$=C$(7):BC$=BC$(7)     :rem 33
1120 GOSUB1440:TI$="ØØØØØØ"       :rem 169
1130 C$=C$(ABS(LEN(C$)-11))       :rem 110
1140 E=PEEK(53279)AND2↑Ø:IFE=1THEN129Ø
                                  :rem 210
1150 IFTI>8ØØTHENC$=C$((LEN(C$)-12)):BC$=
     BC$((LEN(C$)-11)):GOTO146Ø  :rem 155
1160 IFINT(RND(TI)*1ØØ)<2ØTHENC$=BC$
                                   :rem 58
1170 IFINT(RND(TI)*1ØØ)=4THENGOSUB144Ø
                                   :rem 10
1180 TL=ABS(TL+((INT(RND(TI)*3)-1)))
                                    :rem 4
1190 TL=TL-(1*((TL<(4Ø-LEN(C$)))+1))
                                  :rem 116
1200 TL=TL*((TL<Ø)+1)             :rem 116
1210 PRINT:PRINTSPC(TL);C$;       :rem 204
1220 IFAS=1THEN147Ø                :rem 81
1230 P=PEEK(2Ø3):IFP=45THEN128Ø    :rem 23
1240 IFP=5ØTHEN127Ø                :rem 65
1250 P=PEEK(5632Ø)AND12:IFP=8THEN128Ø
                                  :rem 137
1260 IFP<>4THEN129Ø                :rem 81
1270 GOSUB14Ø:POKELD,PEEK(LD)+2:POKELD,PE
     EK(LD)+3:GOTO113Ø             :rem 40
1280 GOSUB14Ø:POKELD,PEEK(LD)-2:POKELD,PE
     EK(LD)-3:GOTO113Ø             :rem 45
1290 E=PEEK(53279)AND2↑Ø:IFE=ØTHEN115Ø
                                  :rem 210
1300 PRINT"{UP}{YEL}":PRINTSPC(TL);"{RVS}
     YOU CRASHED!"                 :rem 11
1310 R=R+INT(TI/4ØØ):IFR>1ØTHENR=1Ø
                                   :rem 70
1320 FORW=ØTO15ØSTEP2:GOSUB14Ø:POKE2Ø4Ø,W
     :POKE2Ø4Ø,192:GOSUB14Ø        :rem 95
1325 POKE2Ø41,W:NEXT              :rem 189
1330 POKE5328Ø,Ø:POKE53281,Ø       :rem 28
1340 PRINTSPC(6)"{RVS}YOUR RATING: ";R$(R
     )                            :rem 186
1350 FORW=1TO1Ø:GETZ$:NEXTW       :rem 225
1360 GETZ$:IFZ$=":"ORZ$=";"ORZ$=""THEN136
     Ø                            :rem 158
1370 POKE53251,Ø                   :rem 86
1380 POKE2Ø4Ø,192:FORW=PEEK(53249)TO12ØST
     EP-1:POKE53249,W:GOSUB14Ø:NEXTW
                                  :rem 157
1390 POKE53248,ABS(PEEK(53248)-1Ø):POKE53
     275,Ø                         :rem 42
1400 POKE53277,1:POKE53271,1:FORW=PEEK(53
     249)TOØSTEP-1:POKE53249,W:GOSUB14Ø
                                    :rem 2
1405 NEXT                           :rem 9
1410 RUN                          :rem 187
1420 POKE53287,PEEK(53287)+1        :rem 6
1430 TI$="ØØØØØØ":R=R+2:IFR>1ØTHENR=1Ø
                                  :rem 123
1440 IFAS=1THEN121Ø                :rem 77
1450 AS=1:POKE5325Ø,ABS(PEEK(53248)-2Ø):P
     OKE53251,255:RETURN          :rem 221
1460 GOSUB142Ø:GOTO121Ø            :rem 73
1470 POKE53251,PEEK(53251)-5      :rem 255
1480 POKE53251,PEEK(53251)-5        :rem Ø
1490 POKE53251,PEEK(53251)-5        :rem 1
1500 POKE53251,PEEK(53251)-5      :rem 249
1510 POKE5325Ø,PEEK(5325Ø)-(5-INT(RND(TI)
     *1Ø))                         :rem 15
1520 IFPEEK(53251)<2ØTHENPOKE53251,Ø:AS=Ø
                                   :rem 96
1530 GOTO123Ø                     :rem 200
1540 PRINT"{CLR}{DOWN} YOU ARE THE PILOT
     {SPACE}OF A NEW SPACESHIP."  :rem 11
1550 PRINT"{DOWN}YOU MUST TEST THE CRAFT
     {SPACE}TO ITS LIMITS."        :rem 23
1560 PRINT"{DOWN}YOU CAN FLY USING THE :
     {SPACE}AND ; KEYS,"          :rem 127
1570 PRINT"{DOWN}OR USE A JOYSTICK (PORT
     {SPACE}2)."                  :rem 228
1580 PRINT"{DOWN} THE SHIP WILL CHANGE CO
     LOR TO WARN OF"              :rem 146
1590 PRINT"{DOWN}CHANGES IN THE CANYON SI
     ZE."                          :rem 83
1610 PRINT"{DOWN} AT THE END OF YOUR FLIG
     HT YOU WILL BE"               :rem 85
1620 PRINT"{DOWN}RANKED. YOU MUST NOT HIT
      THE SIDES OF"               :rem 152
1630 PRINT"{DOWN}THE CANYON, BUT ASTEROID
     S WILL FLY BY."                :rem 6
1640 PRINT"{DOWN} {WHT}GOOD LUCK![5] (H
     IT A KEY TO RESTART)"        :rem 235
1650 WAIT198,1:GETZ$:PRINT"{CLR}";:RUN
                                  :rem 160
```

Program 2:
Canyon Cruiser—VIC Initialization

If using tape rather than disk, line 30 should read:

30 PRINT"LO";CHR$(34);"VIC-CANYON.PRG"

```
10 POKE52,28:POKE56,28:CLR          :rem 18
15 FORI=7168TO7679:POKEI,PEEK(256ØØ+I):NE
   XTI                             :rem 176
20 FORX=ØTO19                       :rem 27
21 READC                           :rem 194
22 FORD=ØTO7:READE:POKE7168+C*8+D,E:NEXTD
                                    :rem 29
23 NEXTX                           :rem 252
30 PRINT"LO";CHR$(34);"VIC-CANYON.PRG";CH
   R$(34);",8"                     :rem 135
40 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT"RU
   N"                               :rem 13
```

```
50 PRINT"{9 UP}";                          :rem 138
60 END                                     :rem 61
63000 DATA033,231,231,231,000,000,231,231
      ,231                                 :rem 173
63001 DATA034,231,231,231,231,231,231,231
      ,231                                 :rem 187
63002 DATA035,000,003,007,055,127,127,127
      ,063                                 :rem 200
63003 DATA036,000,192,248,248,252,124,238
      ,254                                 :rem 223
63004 DATA037,123,127,063,061,063,015,003
      ,000                                 :rem 195
63005 DATA038,254,238,124,252,248,248,192
      ,000                                 :rem 227
63006 DATA042,214,124,254,186,186,146,016
      ,016                                 :rem 223
63007 DATA043,000,000,000,000,000,000,000
      ,000                                 :rem 145
63008 DATA044,255,255,255,255,255,255,255
      ,255                                 :rem 243
63009 DATA047,224,224,224,224,224,224,224
      ,224                                 :rem 215
63010 DATA048,240,240,240,240,240,240,240
      ,240                                 :rem 192
63011 DATA049,231,231,231,007,007,231,231
      ,231                                 :rem 196
63012 DATA050,231,231,231,224,224,231,231
      ,231                                 :rem 191
63013 DATA051,231,231,231,224,224,255,255
      ,255                                 :rem 211
63014 DATA052,231,231,231,007,007,255,255
      ,255                                 :rem 211
63015 DATA053,255,255,255,000,000,231,231
      ,231                                 :rem 199
63016 DATA054,255,255,255,255,255,000,000
      ,000                                 :rem 207
63017 DATA055,255,255,255,224,224,231,231
      ,231                                 :rem 219
63018 DATA056,255,255,255,007,007,231,231
      ,231                                 :rem 219
63019 DATA057,007,007,007,007,007,007,007
      ,007                                 :rem 209
```

Program 3:
Canyon Cruiser—VIC Main Program

This program should be SAVEd as VIC-CANYON.PRG (see line 30 of Program 2).

```
90 POKE36869,240                           :rem 106
100 PRINT"{WHT}"                           :rem 103
110 GOTO150                                :rem 97
120 WX=INT(255-W):IFWX<160THENWX=160
                                           :rem 216
150 PRINT"{CLR}"                           :rem 250
160 FORW=0TO10:READR$(W):NEXTW             :rem 138
170 DATA"STARTER"                          :rem 27
180 DATA"BEGINNER"                         :rem 65
190 DATA"LEARNER"                          :rem 1
200 DATA"FLYER"                            :rem 114
210 DATA"LEADER"                           :rem 158
220 DATA"ACE"                              :rem 187
230 DATA"LIEUTENANT"                       :rem 236
240 DATA"GENERAL"                          :rem 242
250 DATA"COMMODORE"                        :rem 154
260 DATA"FLEET COMMANDER"                  :rem 252
270 DATA"HAN SOLO ?????????"               :rem 66
350 POKE36879,8                            :rem 60
370 PRINT"{HOME}{4 DOWN}{6 SPACES}VIC-CAN
    YON"                                   :rem 153
380 PRINT                                  :rem 40
400 FORW=1TO50:P=INT(RND(TI)*506)+7680:IF
    PEEK(P)=32THENPOKEP,46:POKEP+30720,1
                                           :rem 148
410 NEXTW                                  :rem 43
540 PRINT"{BLU}{HOME}{18 DOWN}{RVS}";SPC(
    5);"GET READY..."                      :rem 229
580 TI$="000005"                           :rem 4
590 PRINT"{HOME}{11 DOWN}{RED}{8 SPACES}
    {RVS}TIME 5{2 LEFT}";                  :rem 68
600 A1=INT(10-VAL(TI$)):PRINTA1;"{3 LEFT}
    ";                                     :rem 206
610 IFA1<>0THEN600                         :rem 7
740 GOTO920:REM ** RUN THE GAME **:rem 37
750 DATA"{WHT},/{4 SPACES}9,{3 SPACES}"
                                           :rem 191
760 DATA"{CYN},/{4 SPACES}9,{3 SPACES}"
                                           :rem 90
770 DATA"{PUR},/{5 SPACES}9,{3 SPACES}"
                                           :rem 88
780 DATA"{GRN},/{6 SPACES}9,{3 SPACES}"
                                           :rem 219
790 DATA"{BLU},/{7 SPACES}9,{3 SPACES}"
                                           :rem 221
800 DATA"{YEL},/{8 SPACES}9,{3 SPACES}"
                                           :rem 84
810 DATA"{RED},/{9 SPACES}9,{3 SPACES}"
                                           :rem 211
820 DATA"{WHT},/{10 SPACES}9,{3 SPACES}"
                                           :rem 189
830 DATA"{WHT},/{4 SPACES}9,{3 SPACES}"
                                           :rem 190
840 DATA"{CYN},/{4 SPACES}9,{3 SPACES}"
                                           :rem 89
850 DATA"{PUR},/{5 SPACES}9,{3 SPACES}"
                                           :rem 87
860 DATA"{GRN},/{6 SPACES}9,{3 SPACES}"
                                           :rem 218
870 DATA"{BLU},/{7 SPACES}9,{3 SPACES}"
                                           :rem 220
880 DATA"{YEL},/{8 SPACES}9,{3 SPACES}"
                                           :rem 92
890 DATA"{RED},/{9 SPACES}9,{3 SPACES}"
                                           :rem 219
900 DATA"{WHT},/{10 SPACES}9,{3 SPACES}"
                                           :rem 188
920 PRINT"{CLR}":PRINT"{UP}";:POKE36869,2
    55                                     :rem 23
925 PRINTSPC(9);"{RED}71118"               :rem 10
950 FORW=1TO10                             :rem 78
960 PRINTSPC(9);"{RED}";CHR$(34);"2 1";CH
    R$(34)                                 :rem 1
970 NEXTW                                  :rem 54
980 PRINT"{RED},,,,,,,7514 3158,,,,,,";
                                           :rem 135
990 FORW=1TO10:PRINTSPC(6);"{RED},2
    {7 SPACES}1,":NEXTW                    :rem 172
1000 POKE7713,42:POKE38433,4               :rem 34
1020 TI$="000005"                          :rem 42
1070 FOR SYP=55TO297STEP22:POKE38400+SYP-
     22,0:POKE7680+SYP-22,32               :rem 58
1075 POKE38400+SYP,4:POKE7680+SYP,42:NEXT
     SYP:SP=297:AB=32:CB=0                 :rem 53
1080 PRINT"{HOME}{20 DOWN}";               :rem 60
1090 FORW=0TO7:READC$(W):NEXTW             :rem 132
1100 FORW=0TO7:READBC$(W):NEXTW            :rem 190
1110 TL=6:C$=C$(7):BC$=BC$(7)              :rem 244
1120 GOSUB1440:TI$="000000"                :rem 169
1130 C$=C$(ABS(LEN(C$)-11))                :rem 110
1150 IFTI>800THENC$=C$((LEN(C$)-12)):BC$=
     BC$((LEN(C$)-11)):GOTO1460            :rem 155
1160 IFINT(RND(TI)*100)<20THENC$=BC$
                                           :rem 58
```

```
1170 IFINT(RND(TI)*100)=4THENGOSUB1440
                                   :rem 10
1180 TL=ABS(TL+((INT(RND(TI)*3)-1)))
                                    :rem 4
1190 TL=TL-(1*((TL<(22-LEN(C$)))+1))
                                  :rem 116
1200 TL=TL*((TL<0)+1)             :rem 116
1205 POKESP+7680,AB:POKESP+38400,CB
                                   :rem 48
1210 PRINT:PRINTSPC(TL);C$;       :rem 204
1220 IFAS=1THEN1470                :rem 81
1230 P=PEEK(203):IF P=64 THEN 1285:rem 29
1235 IF P=45 THEN 1280             :rem 74
1240 IF P=22 THEN 1270             :rem 64
1260 GOTO 1285                    :rem 210
1270 SP=SP+1:IF PEEK(7680+SP)<>32 THEN 13
     00                            :rem 93
1271 AB=PEEK(7680+SP):CB=PEEK(38400+SP)
                                  :rem 227
1272 POKESP+38400,4:POKESP+7680,42:GOTO 1
     130                          :rem 254
1280 SP=SP-1:IF PEEK(7680+SP)<>32 THEN 13
     00                            :rem 96
1281 AB=PEEK(7680+SP):CB=PEEK(38400+SP)
                                  :rem 228
1282 POKESP+38400,4:POKESP+7680,42:GOTO 1
     130                          :rem 255
1285 IF PEEK(7680+SP)<>32AND PEEK(7680+SP
     )<>42THEN 1300                :rem 22
1286 AB=PEEK(7680+SP):CB=PEEK(38400+SP)
                                  :rem 233
1287 POKESP+38400,4:POKESP+7680,42:GOTO 1
     130                            :rem 4
1300 POKE36869,240:PRINT"{CLR}":PRINTSPC(
     5);"{RVS}YOU CRASHED!"        :rem 63
1330 POKE36879,8                  :rem 107
1340 PRINTSPC(5);"{RVS}YOUR RATING:"
                                  :rem 160
1345 PRINTSPC(11-LEN(R$(R))/2);"{RVS}";R$
     (R)                          :rem 175
1350 FORW=1TO10:GETZ$:NEXTW       :rem 225
1360 GETZ$:IFZ$=":"ORZ$=";"ORZ$=""THEN136
     0                            :rem 158
1410 RUN                          :rem 187
1430 TI$="000000":R=R+1:IFR>10THENR=10
                                  :rem 122
1440 IFAS=1THEN1205                :rem 81
1450 AS=1:RETURN                  :rem 230
1460 GOSUB1430:GOTO1205            :rem 78
1470 AS=0                         :rem 205
1475 FORQ=1TOLEN(C$)/2:PRINT"{LEFT}";:NEX
     TQ                           :rem 191
1480 PRINT"{2 UP}#${DOWN}{2 LEFT}%&"
                                  :rem 157
1530 GOTO1230                     :rem 200
```

Tetracrystals Of Veluria

(Article on page 86.)

BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs," "A Beginner's Guide To Typing In Programs," and "The Automatic Proofreader" that appear before the Program Listings.

Program 1: Tetracrystals—VIC Version

```
10 POKE52,25:POKE56,25:CLR:GOSUB5000:CLR
                                  :rem 166
13 S7$="{RVS}{3 SPACES}{OFF}{2 SPACES}":C
   7$="BWRCPGBY"                  :rem 240
20 G1$=" -- Y VALUE --{2 SPACES}":G2$="AB
   CDEFGHIJKLMNOP":G3$="+Ɛ+Ǝ+Ɛ+Ǝ+
   Ɛ+Ǝ+Ɛ+Ǝ+Ɛ+Ǝ+Ɛ+Ǝ+Ɛ+Ǝ+Ɛ+Ǝ+":
   CJ$="XYPC"                     :rem 101
60 TG=1:GOSUB1100                 :rem 235
100 GOSUB8200:GOSUB6000:GOSUB8200:rem 223
160 PRINT"{CLR}SPEED":INPUTHF     :rem 196
180 PRINT"{DOWN}CHR?{DOWN}"       :rem 168
190 GOSUB8000                     :rem 226
195 IF(G$<"1")OR(G$>"4")THEN190   :rem 119
197 G=VAL(G$)*4-1:POKE7167,G       :rem 49
255 PRINT"{CLR}{3 SPACES}** X VALUE
    **":PRINT"{2 SPACES}";G2$:SYS6401
                                  :rem 188
260 H=0:FORG=1TO16:PRINTMID$(G1$,G,1);MID
    $(G2$,G,1);MID$(G3$,H+1,16):H=ABS(H-1
    ):NEXTG                       :rem 125
280 FORG=0TO3                      :rem 11
282 PRINT"{HOME}"MID$(CJ$,G+1,1) :rem 172
285 GOSUB8000                     :rem 231
290 IFG$="{F1}"THEN280            :rem 100
295 IFG$="{F7}"THEN510            :rem 104
300 H=ASC(G$)-49:ONG+1GOTO325,345,365,385
                                  :rem 240
301 GOTO285                       :rem 108
305 NEXTG                          :rem 30
325 H=H-16:IF((H<0)OR(H>15))THEN285
                                  :rem 198
330 PRINT"{HOME}{DOWN}{2 RIGHT}";G2$;:POK
    E7704+H,81:XV=7726+H:GOTO305 :rem 148
345 H=H-16:IF((H<0)OR(H>15))THEN285
                                  :rem 200
350 FORJ=1TO16:POKE7703+J*22,J:NEXTJ:POKE
    7725+H*22,81:YV=H:GOTO305       :rem 4
365 IF((H<0)OR(H>1))THEN285       :rem 250
370 PV=H+49:SB=2↑H:GOTO305         :rem 76
385 IF((H<0)OR(H>7))THEN285         :rem 2
390 POKEXV+YV*22+30720,H:POKEXV+YV*16+302
    08,H:POKEXV+YV*22,PV:POKEXV+YV*16-558
    ,PV                            :rem 14
395 POKEXV+YV*16-302,((H*16)+SB):GOTO280
                                   :rem 31
510 POKE36879,SS:TG=1:CP=0:GOTO750:rem 52
600 SYS6406:SYS6431               :rem 108
650 SYS6577                       :rem 115
655 FORG=1TOHF:NEXTG               :rem 45
656 GETG$:IFG$<>""THENGOSUB7000    :rem 84
657 IFCPTHENGOSUB7310              :rem 62
660 SYS6475:SYS6542               :rem 123
690 GOSUB1000:TG=1                 :rem 35
695 GOSUB1100                     :rem 230
700 SYS6401:SYS6424               :rem 106
750 SYS6573                       :rem 112
755 FORG=1TOHF:NEXTG               :rem 46
756 GETG$:IFG$<>""THENGOSUB7000    :rem 85
757 IFCPTHENGOSUB7310              :rem 63
760 SYS6468:SYS6535               :rem 128
790 GOSUB1000:TG=0                 :rem 35
795 GOSUB1100:GOTO600             :rem 240
1000 J=INT(RND(1)*16):H=INT(RND(1)*3):POK
     E36874+H,PEEK(6993+J):POKE36878,(JOR
     8):RETURN                     :rem 70
1100 POKE36866,128*TG+16:POKE36867,32:POK
     E36864,10:POKE648,2*TG+28:Z=TG*2+156
                                  :rem 119
```

```
1120 FORH=217TO228:POKEH,Z:POKEH+12,Z+1:N
     EXTH:PRINT"{HOME}";:RETURN   :rem 149
5000 OPEN1,1,0,"CRYSTALS T3/JAN"   :rem 85
5020 FORG=0TO39:FORH=1TO16         :rem 82
5030 INPUT#1,Y:X=G*16+H+6400:POKEX,Y:PRIN
     TY;                          :rem 225
5060 NEXTH,G                      :rem 197
5075 INPUT#1,L:IFL<>99999THENSTOP :rem 78
5080 CLOSE1:RETURN                :rem 142
6000 SS=PEEK(36879):PRINT"{CLR}"SPC(66)S7
     $"F1 BDR":PRINTS7$"F3 SCN   :rem 166
6010 PRINTS7$"F5 REV":PRINTS7$"F7 FIN
     {DOWN}"                      :rem 210
6012 FORG=1TO8:PRINTS7$;G;MID$(C7$,G,1);S
     7$:NEXTG                     :rem 216
6020 FORG=0TO7:FORH=0TO7:POKE38583+G*22+H
     ,G:NEXTH:NEXTG               :rem 200
6030 POKE36879,SS                 :rem 219
6050 GOSUB8000:IFG$="{F1}"THENG=(SS+1)AND
     7:SS=(SSAND248)ORG           :rem 202
6052 IFG$="{F3}"THENSS=(SS+16)AND255
                                  :rem 216
6054 IFG$="{F5}"THENG=(SS+8)AND15:SS=(SSA
     ND240)ORG                    :rem 124
6056 IFG$="{F7}"THENRETURN        :rem 227
6060 GOTO6030                     :rem 206
7000 IF(G$<"{F1}")OR(G$>"{F6}")THENRETURN
                                  :rem 144
7010 G=ASC(G$)-132:ONGGOTO7100,7200,7210,
     7300,7400,7450,7600          :rem 142
7100 CP=0:RETURN                  :rem 226
7200 H=25:GOTO7220                :rem 243
7210 H=-25                        :rem 227
7220 GOSUB8000:G=ASC(G$)-48:IF(G<0)OR(G>9
     )THEN7220                    :rem 116
7240 HF=HF+G*H:RETURN              :rem 36
7300 CP=1                         :rem 203
7310 GOSUB8000:IFG$="{F7}"THENRETURN
                                   :rem 95
7320 IFG$="{F1}"THENCP=0:RETURN    :rem 21
7330 GOTO7000                     :rem 205
7400 PV=INT(RND(1)*8)*16+1:GOTO7500
                                  :rem 222
7450 GOSUB8000:G=ASC(G$)-49:IF(G<0)OR(G>7
     )THEN7450                    :rem 125
7460 PV=G*16+1                    :rem 232
7500 XV=INT(RND(1)*16):YV=INT(RND(1)*16):
     QV=XV+YV*16+7424             :rem 118
7510 POKEQV,PV:POKEQV+512,PV:RETURN
                                   :rem 52
7600 RUN13                         :rem 38
8000 GETG$:IFG$=""THEN8000        :rem 191
8010 RETURN                       :rem 169
8200 POKE36866,150:POKE36867,46:POKE36864
     ,5:POKE36879,27:RETURN       :rem 211
```

Program 2:
Tetracrystals—VIC DATA Maker

```
2 REM PREPARES{14 SPACES}DATAFILE FOR
  {10 SPACES}TETRACRYSTALS        :rem 142
5 DIMCS(40)                        :rem 90
8 FORG=1TO40:READH:CS(G)=H:NEXTG   :rem 38
10 PRINT"{CLR} CHECKSUM";          :rem 83
13 FORJ=1TO40                      :rem 10
15 TL=0                           :rem 115
18 FORK=1TO16                      :rem 19
20 READL:TL=TL+L                  :rem 248
23 NEXTK                          :rem 239
25 IFTL<>CS(J)THENPRINT:PRINT"TYPO IN LIN
   E";100+J*10:STOP                :rem 95
28 NEXTJ                          :rem 243
30 READL:IFL<>99999THENPRINT"INSUFFICIENT
   DATA":STOP                     :rem 152
32 PRINT" OK{2 DOWN}"             :rem 242
35 PRINT"GET DATA TAPE READY{3 SPACES}HIT
   A KEY TO CONTINUE              :rem 199
36 GETG$:IFG$=""THEN36              :rem 1
37 PRINT:PRINT                    :rem 190
38 CLR                             :rem 76
40 OPEN1,1,1,"CRYSTALS T3/JAN"    :rem 245
43 FORJ=1TO40:READL:NEXTJ         :rem 114
45 FORJ=1TO40                      :rem 15
50 FORK=1TO16                      :rem 15
53 READL:PRINT#1,L;CHR$(13);      :rem 143
55 NEXTK                          :rem 244
58 NEXTJ                          :rem 246
60 READL:IFL<>99999THENPRINT"CHECK DATA":
   GOTO65                         :rem 208
63 PRINT#1,L                      :rem 194
65 CLOSE1                          :rem 18
99 END                             :rem 73
101 DATA2175,2423,2474,1942,2180   :rem 98
102 DATA2431,2113,1935,2660,2567   :rem 98
103 DATA2317,1710,2352,2061,1928   :rem 93
104 DATA2461,2246,1533,2072,1693 :rem 100
105 DATA1626,1883,1556,1844,2604 :rem 113
106 DATA1715,2310,2659,1771,1908 :rem 109
107 DATA2411,1971,2168,2142,1770 :rem 101
108 DATA2510,1251,3302,0,0         :rem 28
110 DATA169,29,208,3,234,169,31,133,252,1
    69,0,133,251,234,160,0         :rem 68
120 DATA145,251,200,208,251,96,234,169,29
    ,162,31,208,5,234,169,31      :rem 179
130 DATA162,29,133,252,134,254,169,0,133,
    251,133,253,234,160,0,177     :rem 216
140 DATA253,10,9,240,133,1,177,253,9,15,3
    7,1,145,251,200,208           :rem 166
150 DATA238,96,234,169,28,162,29,208,5,23
    4,169,30,162,31,133,252       :rem 143
160 DATA134,254,169,0,133,251,133,253,234
    ,172,255,27,162,4,185,65      :rem 182
170 DATA27,149,0,136,202,208,247,169,32,1
    33,5,234,234,160,0,177         :rem 75
180 DATA253,162,0,9,16,24,232,106,176,3,1
    44,250,234,181,0,145          :rem 220
190 DATA251,200,208,235,96,234,169,148,16
    2,29,208,5,234,169,150,162     :rem 38
200 DATA31,133,252,134,254,169,0,133,251,
    133,253,234,160,0,177,253     :rem 208
210 DATA74,74,74,74,145,251,200,208,245,9
    6,234,234,169,29,208,2         :rem 94
220 DATA169,31,133,2,133,252,160,0,132,1,
    234,177,1,41,4,240            :rem 101
230 DATA5,32,204,25,164,254,200,208,242,9
    6,234,132,254,169,0,133       :rem 116
240 DATA253,152,41,240,208,4,169,17,133,2
    53,152,41,15,208,6,169         :rem 75
250 DATA34,5,253,133,253,200,152,41,15,20
    8,6,169,68,5,253,133          :rem 229
260 DATA253,165,254,9,15,168,200,208,6,16
    9,136,5,253,133,253,234       :rem 138
270 DATA76,11,26,5,253,133,253,96,234,234
    ,164,254,169,48,37,253        :rem 101
280 DATA208,19,152,56,233,17,168,177,1,41
    ,12,240,5,169,3,32            :rem 137
290 DATA4,26,164,254,234,169,80,37,253,20
    8,19,152,56,233,15,168        :rem 100
300 DATA177,1,41,12,240,5,169,5,32,4,26,1
    64,254,234,169,160            :rem 128
310 DATA37,253,208,19,152,24,105,15,168,1
    77,1,41,12,240,5,169          :rem 230
```

```
320 DATA10,32,4,26,164,254,234,169,192,37
    ,253,208,19,152,24,105          :rem 76
330 DATA17,168,177,1,41,12,240,5,169,12,3
    2,4,26,164,254,234             :rem 129
340 DATA169,1,37,253,208,50,165,254,56,23
    3,16,168,177,1,41,15           :rem 244
350 DATA234,234,234,234,208,34,165,254,56
    ,233,16,170,41,240,240,11      :rem 215
360 DATA138,56,233,16,168,177,1,41,12,208
    ,13,164,254,177,1,56           :rem 243
370 DATA233,3,134,251,164,251,145,1,169,2
    ,37,253,208,41,164,254          :rem 76
380 DATA136,177,1,41,15,234,234,234,234,2
    34,234,234,208,25,164,254      :rem 228
390 DATA136,152,41,15,240,7,136,177,1,41,
    12,208,10,164,254,177           :rem 22
400 DATA1,56,233,3,136,145,1,169,4,37,253
    ,208,44,164,254,200            :rem 176
410 DATA177,1,41,15,234,234,234,234,234,2
    34,234,208,28,169,1,133        :rem 121
420 DATA251,164,254,200,200,152,41,15,240
    ,6,177,1,41,12,208,9           :rem 208
430 DATA164,254,177,1,56,233,3,145,251,16
    9,8,37,253,208,44,165           :rem 45
440 DATA254,24,105,16,168,177,1,41,15,234
    ,234,234,234,208,28,169        :rem 134
450 DATA16,133,251,165,254,24,105,32,176,
    7,168,177,1,41,12,208           :rem 27
460 DATA10,164,254,177,1,56,233,3,145,251
    ,234,164,254,96,234,234        :rem 135
470 DATA87,81,90,0,219,102,160,42,65,90,8
    3,88,43,37,36,28                :rem 57
480 DATA163,175,175,187,195,201,207,209,2
    15,215,219,221,225,228,232,235
                                   :rem 235
490 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                    :rem 75
500 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                    :rem 67
50000 DATA99999                     :rem 44
```

Program 3: Tetracrystals—64 version

```
10 GOTO9000:REM MACHINE LANGUAGE LOADER
                                   :rem 113
13 S7$="{RVS}{7 SPACES}{OFF}{4 SPACES}":C
   7$="BLKWHTREDCYNPURGBLUYEL":PS=53280
   :RV=49760                        :rem 17
20 G1$=" -- Y VALUE --{2 SPACES}":G2$="AB
   CDEFGHIJKLMNOP":G3$="+E+3+E+3+
   E+3+E+3+E+3+E+3+E+3+E+3+"
                                   :rem 181
30 CJ$="XVALUEYVALUEPLANT#COLOR"  :rem 97
60 GOSUB1100:REM NORMAL COLORS     :rem 97
100 GOSUB6000:REM SET UP SCREEN/BORDER
                                    :rem 51
110 GOSUB1100                      :rem 212
160 PRINT"{CLR}SPEED? (THE LOWER THE NUMB
    ER THE FASTER)":INPUTHF         :rem 14
180 PRINT"{DOWN}WHICH CHARACTER SET?
    {SHIFT-SPACE}(1-4){DOWN}"       :rem 58
190 GETG$:IFG$=""THEN190            :rem 99
195 IF(G$<"1")OR(G$>"4")THEN190   :rem 119
197 G=VAL(G$)*4-1:POKE49664,G     :rem 105
255 PRINT"{CLR}"SPC(130)"{3 SPACES}** X
    VALUE **":PRINTSPC(52);G2$:SYS4967
    0                              :rem 100
260 H=0:FORG=1TO16:PRINTSPC(9)MID$(G1$,G,
    1)" "MID$(G2$,G,1);             :rem 90
270 PRINTMID$(G3$,H+1,16):H=ABS(H-1):NEXT
    G                              :rem 194
280 FORG=0TO3                       :rem 11
282 PRINT"{HOME}"MID$(CJ$,G*6+1,6):PRINT"
    *-F1 TO CANCEL-*-F7 TO BEGIN GAME-*"
                                   :rem 173
285 GETG$:IFG$=""THEN285           :rem 109
290 IFG$="{F1}"THEN280             :rem 100
295 IFG$="{F7}"THEN510             :rem 104
300 H=ASC(G$)-49:ONG+1GOTO325,345,365,385
                                   :rem 240
301 GOTO285                        :rem 108
305 NEXTG                           :rem 30
325 H=H-16:IF((H<0)OR(H>15))THEN285
                                   :rem 198
330 PRINT"{HOME}{5 DOWN}"SPC(12);G2$;:POK
    E1236+H,81:XV=1276+H:GOTO305   :rem 44
345 H=H-16:IF((H<0)OR(H>15))THEN285
                                   :rem 200
350 FORJ=1TO16:POKE1235+J*40,J:NEXTJ:POKE
    1275+H*40,81:YV=H:GOTO305      :rem 248
365 IF((H<0)OR(H>1))THEN285       :rem 250
370 PV=H+49:SB=2↑H:GOTO305          :rem 76
385 IF((H<0)OR(H>7))THEN285          :rem 2
390 POKEXV+YV*40+54272,H:POKEXV+YV*40,PV
                                   :rem 213
395 POKEXV+YV*16+48132,((H*16)+SB):GOTO28
    0                              :rem 138
510 POKEPS,RB:POKEPS+1,SC:CP=0    :rem 140
520 PRINT"{CLR}{DOWN}"SPC(7);"T E T R A C
    R Y S T A L S":GOTO700         :rem 248
600 SYS49670:SYS49695              :rem 233
650 SYS49841                       :rem 164
655 FORG=1TOHF:NEXTG                :rem 45
656 GETG$:IFG$<>""THENGOSUB7000     :rem 84
657 IFCPTHENGOSUB7310               :rem 62
660 SYS49806:SYS49739              :rem 239
690 GOSUB1000:REM MUSIC            :rem 127
700 SYS49665:SYS49688              :rem 240
750 SYS49837                       :rem 170
755 FORG=1TOHF:NEXTG                :rem 46
756 GETG$:IFG$<>""THENGOSUB7000     :rem 85
757 IFCPTHENGOSUB7310               :rem 63
760 SYS49799:SYS49732              :rem 244
790 GOSUB1000:REM MUSIC            :rem 128
795 GOTO600                        :rem 116
1000 REM MUSIC                      :rem 38
1050 VM=(INT(RND(1)*3))*16:VM=VM+(INT(RND
     (1)*16))                      :rem 243
1060 POKE50688,VM                  :rem 208
1070 SYS50689                      :rem 211
1080 RETURN                        :rem 169
1099 RETURN                        :rem 179
1100 POKE53280,254:POKE53281,246:RETURN
                                     :rem 8
6000 RB=PEEK(PS):SC=PEEK(PS+1):PRINT"
     {CLR}"SPC(131)S7$"F1 BORDER":rem 195
6005 PRINTSPC(11)S7$"F3 SCREEN    :rem 250
6008 PRINTSPC(11)S7$"F5 REVERSE    :rem 91
6010 PRINTSPC(11)S7$"F7 CONTINUE{DOWN}
                                   :rem 176
6012 FORG=1TO8:PRINTSPC(11)S7$;G;MID$(C7$
     ,G*3-2,3):NEXTG                :rem 70
6020 FORG=0TO7:FORH=0TO7:POKE55627+G*40+H
     ,G:NEXTH:NEXTG                :rem 198
6030 POKEPS,RB:POKEPS+1,SC         :rem 133
6050 GETG$:IFG$=""THEN6050         :rem 197
6051 IFG$="{F1}"THENRB=(RB+1)AND15:rem 70
6052 IFG$="{F3}"THENSC=(SC+1)AND15:rem 76
6054 IFG$="{F5}"THENPOKERV,((PEEK(RV)+128
     )AND255)                      :rem 248
6055 IFG$="{F5}"THENFORX=1344TO1663:POKEX
     ,((PEEK(X)+128)AND255):NEXTX :rem 31
```

```
6056 IFG$="{F7}"THENRETURN           :rem 227
6060 GOTO6030                        :rem 206
7000 IF(G$<"{F1}")OR(G$>"{F6}")THENRETURN
                                     :rem 144
7010 G=ASC(G$)-132:ONGGOTO7100,7200,7210,
     7300,7400,7450,7600             :rem 142
7100 CP=0:RETURN                     :rem 226
7200 H=25:GOTO7220                   :rem 243
7210 H=-25                           :rem 227
7220 GOSUB8000:G=ASC(G$)-48:IF(G<0)OR(G>9
     )THEN7220                       :rem 116
7240 HF=HF+G*H:RETURN                 :rem 36
7300 CP=1:RETURN                     :rem 229
7310 GOSUB8000:IFG$="{F7}"THENRETURN
                                      :rem 95
7320 IFG$="{F1}"THENCP=0:RETURN       :rem 21
7330 GOTO7000                        :rem 205
7400 PV=INT(RND(1)*8)*16+1:GOTO7500
                                     :rem 222
7450 GOSUB8000:G=ASC(G$)-49:IF(G<0)OR(G>7
     )THEN7450                       :rem 125
7460 PV=G*16+1                       :rem 232
7500 XV=INT(RND(1)*16):YV=INT(RND(1)*16):
     QV=XV+YV*16+49152               :rem 170
7510 POKEQV,PV:POKEQV+256,PV:RETURN
                                      :rem 57
7600 RUN13                            :rem 38
8000 GETG$:IFG$=""THEN8000           :rem 191
8010 RETURN                          :rem 169
9000 DIMCS(54)                       :rem 243
9010 FORG=1TO54:READCS(G):NEXTG      :rem 182
9020 PRINT"{CLR}"SPC(6);"T E T R A C R Y
     {SPACE}S T A L S{2 DOWN}":PRINT"-- C
     HECKSUM:";                       :rem 95
9030 FORJ=1TO54                      :rem 119
9040 TL=0                            :rem 218
9050 FORK=1TO16                      :rem 120
9060 READL:TL=TL+L                   :rem 101
9070 NEXTK                            :rem 90
9080 IFTL<>CS(J)THENPRINT:PRINT"TYPO IN L
     INE";9340+J*10:STOP               :rem 8
9090 NEXTJ                            :rem 91
9100 READL:IFL<>99999THENSTOP         :rem 83
9110 PRINT" OK{DOWN}":PRINT"-- LOADING MA
     CHINE LANGUAGE."                 :rem 17
9120 CLR:M=49664                     :rem 126
9130 POKEM,3                         :rem 168
9150 FORJ=1TO54:READK:NEXTJ          :rem 222
9155 FORJ=1TO40                      :rem 122
9160 FORK=1TO16:M=M+1                :rem 231
9165 READL:POKEM,L                   :rem 107
9170 NEXTK                            :rem 91
9175 NEXTJ                            :rem 95
9179 M=50688                         :rem 111
9180 FORJ=41TO46                     :rem 178
9185 FORK=1TO16:M=M+1                :rem 238
9190 READL:POKEM,L                   :rem 105
9195 NEXTK                            :rem 98
9200 NEXTJ                            :rem 84
9204 M=50943                          :rem 94
9205 FORJ=47TO54                     :rem 181
9210 FORK=1TO16:M=M+1                :rem 227
9215 READL:POKEM,L                   :rem 103
9220 NEXTK                            :rem 87
9225 NEXTJ                            :rem 91
9230 READL:IFL<>99999THENSTOP         :rem 87
9240 RUN13                            :rem 40
9270 DATA2500,2910,2637,1944,2015:rem 155
9272 DATA2122,2553,2150,2153,2209:rem 148
9274 DATA1612,1875,2521,2061,1928:rem 166
9276 DATA2461,2415,1534,2241,1863:rem 163
9278 DATA1627,2052,1726,1845,2604:rem 171
9280 DATA1717,2311,2660,1772,1910:rem 158
9282 DATA2412,1972,2169,2143,1771:rem 167
9284 DATA2511,1251,2546,2115,1718:rem 158
9286 DATA1762,2481,2247,1939,2497:rem 189
9288 DATA1436,546,282,778,929    :rem 248
9290 DATA1246,1495,1077,1002     :rem 164
9350 DATA169,192,208,3,234,169,193,133,25
     2,169,0,133,251,234,160,0   :rem 237
9360 DATA145,251,200,208,251,96,234,169,1
     92,162,193,208,5,234,169,193:rem 149
9370 DATA162,192,133,252,134,254,169,0,13
     3,251,133,253,234,160,0,177  :rem 72
9380 DATA253,10,9,240,133,2,177,253,9,15,
     37,2,145,251,200,208        :rem 231
9390 DATA238,96,234,169,4,32,81,196,240,5
     ,169,4,32,85,196,234         :rem 20
9400 DATA172,0,194,162,4,185,65,196,149,1
     ,136,202,208,247,169,32     :rem 146
9410 DATA133,6,234,234,234,234,234,234,23
     4,160,15,177,253,162,0,9    :rem 176
9420 DATA16,24,232,106,176,3,144,250,234,
     181,1,145,251,136,16,235    :rem 170
9430 DATA32,101,196,144,228,96,169,216,32
     ,81,196,240,5,169,216,32    :rem 201
9440 DATA85,196,234,160,15,177,253,74,74,
     74,74,145,251,136,16,245    :rem 213
9450 DATA32,101,196,144,238,96,234,0,0,0,
     0,0,169,192,208,2            :rem 77
9460 DATA169,193,133,3,133,252,160,0,132,
     2,234,177,2,41,4,240        :rem 224
9470 DATA5,32,204,194,164,254,200,208,242
     ,96,234,132,254,169,0,133   :rem 234
9480 DATA253,152,41,240,208,4,169,17,133,
     253,152,41,15,208,6,169     :rem 138
9490 DATA34,5,253,133,253,200,152,41,15,2
     08,6,169,68,5,253,133        :rem 36
9500 DATA253,165,254,9,15,168,200,208,6,1
     69,136,5,253,133,253,234    :rem 192
9510 DATA76,11,195,5,253,133,253,96,234,2
     34,164,254,169,48,37,253    :rem 210
9520 DATA208,19,152,56,233,17,168,177,2,4
     1,12,240,5,169,3,32         :rem 192
9530 DATA4,195,164,254,234,169,80,37,253,
     208,19,152,56,233,15,168    :rem 209
9540 DATA177,2,41,12,240,5,169,5,32,4,195
     ,164,254,234,169,160        :rem 247
9550 DATA37,253,208,19,152,24,105,15,168,
     177,2,41,12,240,5,169        :rem 38
9560 DATA10,32,4,195,164,254,234,169,192,
     37,253,208,19,152,24,105    :rem 194
9570 DATA17,168,177,2,41,12,240,5,169,12,
     32,4,195,164,254,234        :rem 248
9580 DATA169,1,37,253,208,50,165,254,56,2
     33,16,168,177,2,41,15        :rem 52
9590 DATA234,234,234,234,208,34,165,254,5
     6,233,16,170,41,240,240,11   :rem 22
9600 DATA138,56,233,16,168,177,2,41,12,20
     8,13,164,254,177,2,56        :rem 43
9610 DATA233,3,134,251,164,251,145,2,169,
     2,37,253,208,41,164,254     :rem 131
9620 DATA136,177,2,41,15,234,234,234,234,
     234,234,234,208,25,164,254   :rem 27
9630 DATA136,152,41,15,240,7,136,177,2,41
     ,12,208,10,164,254,177       :rem 77
9640 DATA2,56,233,3,136,145,2,169,4,37,25
     3,208,44,164,254,200        :rem 241
9650 DATA177,2,41,15,234,234,234,234,234,
     234,234,208,28,169,1,133    :rem 185
```

```
9660 DATA251,164,254,200,200,152,41,15,24
     0,6,177,2,41,12,208,9           :rem 16
9670 DATA164,254,177,2,56,233,3,145,251,1
     69,8,37,253,208,44,165         :rem 109
9680 DATA254,24,105,16,168,177,2,41,15,23
     4,234,234,234,208,28,169       :rem 198
9690 DATA16,133,251,165,254,24,105,32,176
     ,7,168,177,2,41,12,208          :rem 91
9700 DATA10,164,254,177,2,56,233,3,145,25
     1,234,164,254,96,234,234       :rem 190
9710 DATA87,81,90,0,219,102,160,42,65,90,
     83,88,43,37,36,28              :rem 111
9720 DATA162,192,208,2,162,193,133,252,13
     4,254,169,132,133,251,169,0     :rem 79
9730 DATA133,253,96,234,24,169,16,101,253
     ,144,1,96,133,253,169,40       :rem 193
9740 DATA101,251,144,2,230,252,133,251,24
     ,96,234,0,0,0,0,0               :rem 50
9749 REM -- MACHINE LANGUAGE MUSIC --
     {18 SPACES}COULD BE OPTIONAL:rem 147
9750 DATA173,0,198,10,170,74,41,240,74,72
     ,74,74,74,133,251,104           :rem 40
9760 DATA56,229,251,168,189,32,199,153,0,
     199,232,200,189,32,199,153      :rem 69
9770 DATA 0,199,136,132,252,162,7,185,0,1
     99,153,0,212,200,202,208       :rem 128
9780 DATA246,169,1,141,3,212,169,10,141,2
     4,212,164,252,169,1,25          :rem 79
9790 DATA4,199,153,4,212,162,5,160,0,234,
     234,234,200,208,250,238        :rem 128
9800 DATA3,212,238,24,212,202,208,241,96,
     0,0,0,0,0,0,0                   :rem 99
9809 REM -- MUSIC DATA STATEMENTS --
                                     :rem 21
9810 DATA0,0,0,5,64,140,48,0,0,0,0,32,152
     ,105,0,0                        :rem 99
9820 DATA0,0,16,57,194,0,0,0,15,0,0,0,0,0
     ,0,0                           :rem 152
9830 DATA134,3,180,4,71,5,71,6,12,7,12,7,
     104,9,143,10                    :rem 80
9840 DATA143,10,143,12,24,14,24,14,195,16
     ,209,18,31,21,30,25            :rem 168
9850 DATA49,28,165,31,135,33,162,37,62,42
     ,193,44,60,50,99,56            :rem 216
9860 DATA75,63,15,67,15,67,69,75,125,84,2
     14,94,121,100,199,112           :rem 63
9870 DATA12,7,97,8,143,10,143,10,218,11,1
     43,12,24,14,210,15             :rem 118
9880 DATA195,16,209,18,31,21,30,25,49,28,
     49,28,162,37,62,42             :rem 161
10000 DATA99999                      :rem 40
```

Hardhat Climber

(Article on page 82.)

BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs," "A Beginner's Guide To Typing In Programs," and "The Automatic Proofreader" that appear before the Program Listings.

Program 1:
Hardhat Climber—VIC Version

```
0 POKE51,192:POKE52,29:POKE55,192:POKE56,
  29:POKE36869,255:POKE36878,15:POKE36879
  ,25                                 :rem 83
1 D=37154:P1=37151:P2=37152:DO(0)=-1:DO(1
  )=1:DI=DO(INT(RND(1)*2))            :rem 54
2 A$=">>>>>>>>>>>>>>>>>>>>>{LEFT}{INST}>"
  :DIMB(11):G=30720:E2=0             :rem 126
3 SC=1:CH=2:E1=0:D(0)=4:D(1)=2:D(4)=7:Z=5
  7:E3=1:Q=10000:J=56                 :rem 32
10 GOSUB100:POKE36869,255:E4=0:E5=2
                                     :rem 246
15 GOSUB70:H=0:Y=0                    :rem 60
16 S=8143+INT(RND(1)*20):IFPEEK(S+22)=62O
   RPEEK(S)=59THEN16                 :rem 145
17 T=PEEK(S):POKES,58:POKES+G,0      :rem 161
19 V=7712+B(Y):W=62:DO=DO(INT(RND(1)*2))
                                       :rem 7
20 SYS828:ONPEEK(1)GOTO35,26,28,31,33
                                     :rem 110
21 FORN=1TO23:NEXT:GOTO41             :rem 95
26 IFPEEK(S+22)=ZTHENPOKES,T:POKES+G,D(T-
   J):S=S+22:GOTO40                   :rem 84
27 GOTO41                              :rem 7
28 DI=-1:IFPEEK(S+21)<62THENPOKES,T:POKES
   +G,D(T-J):S=S-1:GOTO40            :rem 147
29 IFT<>ZTHENPOKES,T:POKES+G,D(T-J):S=S+D
   I:T=PEEK(S):GOTO55                 :rem 83
30 GOTO41                              :rem 1
31 IFT=ZTHENPOKES,T:POKES+G,D(T-J):S=S-22
   :GOTO40                            :rem 78
32 GOTO41                              :rem 3
33 DI=1:IFPEEK(S+23)<62THENPOKES,T:POKES+
   G,D(T-J):S=S+1:GOTO40              :rem 98
34 GOTO29                             :rem 11
35 POKE36876,240:POKES,T:POKES+G,D(T-J):S
   =S-22+DI:T=PEEK(S):POKES,58:IFT=60THEN
   55                                  :rem 5
36 IFPEEK(S+22)=60THENSS=SS+1000:PRINT"
   {HOME}{RVS}"TAB(8-LEN(STR$(SS)))SS
                                      :rem 62
37 FORN=1TO5:NEXT:POKES,T:POKES+G,D(T-J):
   S=S+22+DI:T=PEEK(S):POKES,58:IFPEEK(S+
   22)>61THEN55
38 POKES+G,0:POKE36876,0:GOTO41     :rem 102
40 POKE36876,200:POKE36876,0:T=PEEK(S):PO
   KES,58:POKES+G,0                  :rem 165
41 IFT=61THENSS=SS+150:PRINT"{HOME}{RVS}"
   TAB(8-LEN(STR$(SS)))SS:H=H+1:T=62:IFH=
   16THEN64                          :rem 240
42 IFT=60THEN55                      :rem 133
43 GOSUB98                            :rem 88
45 POKEV,W:POKEV+G,D(W-J):V=V+DO:W=PEEK(V
   ):POKEV,60:POKEV+G,7              :rem 145
46 IFDO=22ANDPEEK(V+22)=56THENDO=DO(INT(R
   ND(1)*2)):GOTO48                  :rem 195
47 IFW=63THENDO=22                    :rem 90
48 IFW=58THEN55                      :rem 149
49 IFV<8164THEN20                    :rem 242
50 Y=Y+1:IFY=12THEN55                 :rem 11
51 POKEV,62:GOTO19                    :rem 92
55 POKE36876,0:SO=250:IFT=60THENT=W
                                     :rem 138
56 POKE36874,SO:IFPEEK(S+22)<>56ANDS<8164
   THENPOKES,T:POKES+G,D(T-J):S=S+22:T=PE
   EK(S):POKES,58:POKES+G,0          :rem 185
57 FORN=1TO17:NEXT:SO=SO-5:IFSO>150THEN56
                                     :rem 137
58 POKE36874,0:CH=CH-1:IFCH=-1THENPOKED,2
   55:POKE36869,240:PRINT"{CLR}{BLK}"SS:E
   ND                                :rem 104
59 PRINT"{HOME}{RVS}"TAB(14)CH::POKEV,W:P
   OKEV+G,D(W-J):Y=Y+1:IFW=58THENPOKEV,T:
   POKEV+G,D(T-J)                    :rem 248
60 IFY>10THEN15                      :rem 130
61 IFS>8163THENPOKES,T:GOTO16        :rem 100
```

```
62 POKES,58:POKES+G,Ø:GOTO19        :rem 234
64 IFY=11THEN67                     :rem 141
65 FORN=Y+1TO11:POKE7712+B(N),62:SS=SS+1Ø
   Ø:PRINT"{HOME}{RVS}"TAB(8-LEN(STR$(SS)
   ))SS:GOSUB98                     :rem 194
66 POKE36877,25Ø:FORM=24ØTO25Ø:POKE36876,
   M:NEXT:POKE36876,Ø:POKE36877,Ø:NEXT
                                    :rem 113
67 E2=E2+.Ø5:SC=SC+1:E1=E1+1:IFE1>8THENE1
   =8                               :rem 227
68 GOTO15                           :rem 13
69 GOTO69                           :rem 23
7Ø PRINT"{CLR}{PUR}";:FORN=1TO21:PRINTA$:
   NEXT:PRINTA$"{HOME}":B$=">888888888888
   88888888"                        :rem 1Ø8
71 PRINT"{2 DOWN}"TAB(6)"?>>>>>>>>?{RED}"
   :PRINTTAB(6)"9{PUR}88888888{RED}9":PRI
   NTTAB(6)"9>>>>>>>>9":PRINTTAB(6)"9>>>>
   >>>>9{PUR}"                      :rem 12
72 FORN=1TO3:PRINTB$"{4 DOWN}":NEXT:PRINT
   B$"{HOME}";:POKE8185,62          :rem 162
73 PRINT"{RVS}"TAB(8-LEN(STR$(SS)))SS;TAB
   (14)CH;TAB(17)SC:POKE7697,163:FORN=ØTO
   11:POKE7712+B(N),6Ø              :rem 61
74 POKE7712+B(N)+G,7:NEXT:FORN=7834TO8164
   STEP11Ø:IFN=8164THEN8Ø           :rem 73
75 FORO=1TO3                        :rem 23Ø
76 R=N+1+INT(RND(1)*2Ø):IFPEEK(R)<>56THEN
   76                               :rem 17
77 FORM=RTOR+88STEP22:POKEM,57:POKEM+G,2:
   NEXT:IFO>1ANDRND(1)<E2THENPOKER+(INT(R
   ND(1)*2)+2)*22,63                :rem 2Ø3
78 IFRND(1)<.5ANDPEEK(R-22)=62THENPOKER-2
   2,63                             :rem 17Ø
79 NEXT                             :rem 175
8Ø FORO=1TOE1                       :rem 37
81 R=N+3+INT(RND(1)*16):IFPEEK(R)<>56ORPE
   EK(R-22)<>62ORPEEK(R+1)=62ORPEEK(R-1)=
   62THEN85                         :rem 198
84 POKER,62:POKER-22,63             :rem 98
85 NEXT                             :rem 172
86 FORO=1TO4                        :rem 233
87 R=N-21+INT(RND(1)*2Ø):IFPEEK(R)<>62ORP
   EEK(R+22)=62THEN87               :rem 227
88 POKER,61:POKER+G,Ø:NEXT:NEXT     :rem 255
89 POKE771Ø,63:POKE7715,63:POKE7731,63:PO
   KE7738,63                        :rem 125
9Ø FORN=7812TO8142STEP11Ø:POKEN,63:NEXT:F
   ORN=7833TO8163STEP11Ø:POKEN,63:NEXT:RE
   TURN                             :rem 85
98 IFSS>=Q*E3THENCH=CH+1:E3=E3+1:PRINT"
   {HOME}{RVS}"TAB(14)CH            :rem 253
99 RETURN                           :rem 82
1ØØ DATA,1,21,22,23,24,42,43,44,45,46,47
                                    :rem 197
1Ø1 DATA255,255,153,1Ø2,1Ø2,153,255,255,1
    95,255,255,195,195,255,255,195,6Ø,6Ø,
    25,255,188                      :rem 64
1Ø2 DATA6Ø,36,231,3,4,24,24,6Ø,126,126,6Ø
    ,6Ø,66,165,153,153,165,66,6Ø,,24,36,1
    26,126,126                      :rem 11
1Ø3 DATA126,,,,,,,,,,,,,,,,         :rem 51
1Ø6 DATA169,,133,1,169,255,141,34,145,169
    ,32,44,31,145,2Ø8,5,169,1,133,1,96,16
    9,8,44                          :rem 98
1Ø7 DATA31,145,2Ø8,5,169,2,133,1,96,169,1
    6,44,31,145,2Ø8,5,169,3,133,1,96,169,
    4,44,31                         :rem 142
1Ø8 DATA145,2Ø8,3,133,1,96,169,127,141,34
    ,145,169,128,44,32,145,2Ø8,4,169,5,13
    3,1,96                          :rem 1Ø2
1Ø9 FORN=ØTO11:READB(N):NEXT:FORN=7616TO7
    679:READM:POKEN,M:NEXT          :rem 61
11Ø FORN=828TO899:READM:POKEN,M:NEXT:RETU
    RN                              :rem 89
```

Program 2:
Hardhat Climber—64 Version

```
1 PRINT"{CLR}{7 DOWN}{12 RIGHT}HARDHAT CL
  IMBER":POKE5328Ø,1:POKE53281,1:GOSUB78Ø
                                    :rem 76
2 POKE52,48:POKE56,48:CLR           :rem 231
5 PRINT"{CLR}"                      :rem 153
6 PRINT"{7 DOWN}{12 RIGHT}HARDHAT CLIMBER
  "                                 :rem 212
1Ø POKE5328Ø,1:POKE53281,1          :rem 184
2Ø DO(Ø)=-1:DO(1)=1:DI=DO(INT(RND(1)*2))
                                    :rem 18Ø
3Ø A$=">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
   >>>>>{LEFT}{INST}>":DIMB(11):G=54272:E
   2=Ø                              :rem 19
4Ø SC=1:CH=2:E1=Ø:D(Ø)=4:D(1)=2:D(4)=7:Z=
   57:E3=1:Q=1ØØØØ:J=56             :rem 81
44 FORI=GTOG+24:POKEI,Ø:NEXT        :rem 3Ø
45 POKEG+24,15:POKEG+5,17:POKEG+6,24Ø:POK
   EG,1ØØ                           :rem 23Ø
5Ø GOSUB71Ø:E4=Ø:E5=2               :rem 192
6Ø GOSUB5ØØ:H=Ø:Y=Ø                 :rem 1Ø6
7Ø S=19Ø5+INT(RND(1)*38):IFPEEK(S+4Ø)=62O
   RPEEK(S)=59THEN7Ø                :rem 153
8Ø T=PEEK(S):POKES,58:POKES+G,Ø     :rem 161
9Ø V=1123+B(Y):W=62:DO=DO(INT(RND(1)*2))
                                    :rem 252
1ØØ JV=255-PEEK(56321):GETQ$        :rem 61
1Ø1 IFJV=1THENJV=4:GOTO1Ø9          :rem 124
1Ø2 IFJV=4THENJV=3:GOTO1Ø9          :rem 127
1Ø3 IFJV=8THENJV=5:GOTO1Ø9          :rem 134
1Ø4 IFJV=2THEN1Ø9                   :rem 252
1Ø5 IFJVAND16=16THENJV=1:GOTO1Ø9    :rem 237
1Ø8 JV=Ø                            :rem 166
1Ø9 ONJVGOTO21Ø,12Ø,14Ø,17Ø,19Ø     :rem 173
11Ø FORN=1TO23:NEXT:GOTO26Ø         :rem 193
12Ø IFPEEK(S+4Ø)=ZTHENPOKES,T:POKES+G,D(T
    -J):S=S+4Ø:GOTO25Ø              :rem 178
13Ø GOTO26Ø                         :rem 1Ø1
14Ø DI=-1:IFPEEK(S+39)<62THENPOKES,T:POKE
    S+G,D(T-J):S=S-1:GOTO25Ø        :rem 25Ø
15Ø IFT<>ZTHENPOKES,T:POKES+G,D(T-J):S=S+
    DI:T=PEEK(S):GOTO36Ø            :rem 173
16Ø GOTO26Ø                         :rem 1Ø4
17Ø IFT=ZTHENPOKES,T:POKES+G,D(T-J):S=S-4
    Ø:GOTO25Ø                       :rem 181
18Ø GOTO26Ø                         :rem 1Ø6
19Ø DI=1:IFPEEK(S+41)<62THENPOKES,T:POKES
    +G,D(T-J):S=S+1:GOTO25Ø         :rem 2Ø1
2ØØ GOTO15Ø                         :rem 97
21Ø POKEG+1,17:POKEG+4,17           :rem 156
215 POKES,T:POKES+G,D(T-J):S=S-4Ø+DI:T=PE
    EK(S):POKES,58:IFT=6ØTHEN36Ø    :rem 43
22Ø IFPEEK(S+4Ø)=6ØTHENSS=SS+1ØØØ:PRINT"
    {HOME}{RVS}"TAB(8-LEN(STR$(SS)))SS
                                    :rem 1Ø5
23Ø FORN=1TO5:NEXT:POKES,T:POKES+G,D(T-J)
    :S=S+4Ø+DI:T=PEEK(S):POKES,58:rem 2Ø4
235 IFS>1943ORPEEK(S+4Ø)>61THEN36Ø
                                    :rem 241
24Ø POKEG+4,16:POKES+G,Ø:GOTO26Ø    :rem 147
25Ø POKEG+1,9:POKEG+4,17:POKEG+4,16:T=PEE
    K(S):POKES,58:POKES+G,Ø         :rem 134
```

```
260 IFT<>61THEN270                   :rem 36
265 SS=SS+150:PRINT"{HOME}{RVS}"TAB(8-LEN
    (STR$(SS)))SS:H=H+1:T=62:IFH=16THEN44
    0                                :rem 160
270 IFT=60THEN360                    :rem 231
280 GOSUB690                         :rem 185
290 POKEV,W:POKEV+G,D(W-J):V=V+DO:W=PEEK(
    V):POKEV,60:POKEV+G,7            :rem 195
300 IFDO=40ANDPEEK(V+40)=56THENDO=DO(INT(
    RND(1)*2)):GOTO320               :rem 21
310 IFW=63THENDO=40                  :rem 131
320 IFW=58THEN360                    :rem 237
330 IFV<1944THEN100                  :rem 73
340 Y=Y+1:IFY=12THEN60               :rem 57
350 POKEV,62:GOTO90                  :rem 141
360 POKEG+4,16:SO=250:IFT=60THENT=W
                                     :rem 136
361 FORQQ=1TO30:NEXT                 :rem 16
365 T1=T                             :rem 180
370 POKEG+1,SO:POKEG+4,33:IFPEEK(S+40)<>5
    6ANDS<1944THEN375                :rem 227
372 GOTO 380                         :rem 112
375 POKES,T:POKES+G,D(T-J):S=S+40:T=PEEK(
    S):POKES,58:POKES+G,0            :rem 122
380 FORN=1TO17:NEXT:SO=SO-5:IFSO>150THEN3
    70                               :rem 231
390 POKEG+4,16:CH=CH-1:IFCH=-1THENPOKE532
    72,21:GOTO1000                   :rem 2
400 PRINT"{HOME}{RVS}"TAB(25)CH:POKEV,W:P
    OKEV+G,D(W-J):Y=Y+1              :rem 43
405 IFW=58THENPOKEV,T1:POKEV+G,D(T1-J)
                                     :rem 124
410 IFY>10THEN60                     :rem 177
420 IFS>1943THENPOKES,T:GOTO70       :rem 146
430 POKES,58:POKES+G,0:GOTO90        :rem 24
440 IFY=11THEN470                    :rem 233
450 FORN=Y+1TO11:POKE1123+B(N),62:SS=SS+1
    00:PRINT"{HOME}{RVS}"TAB(8-LEN(STR$(S
    S)))SS                           :rem 187
455 GOSUB690:NEXT                    :rem 54
460 POKEG+4,33:FORM=9TO176STEP2:POKEG+1,M
    :NEXT:POKEG+4,32                 :rem 50
470 E2=E2+.05:SC=SC+1:E1=E1+1:IFE1>8THENE
    1=8                              :rem 17
480 GOTO60                           :rem 59
490 GOTO490                          :rem 115
500 PRINT"{CLR}{PUR}";:FORN=1TO22:PRINTA$
    :NEXT:PRINTA$"{HOME}"            :rem 220
505 B$=">888888888888888888888888888888888
    888888"                          :rem 15
510 PRINT"{3 DOWN}"TAB(15)"?>>>>>>>>?
    {RED}":PRINTTAB(15)"9{PUR}88888888
    {RED}9"                          :rem 121
515 PRINTTAB(15)"9>>>>>>>>9":PRINTTAB(15)
    "9>>>>>>>>9{PUR}"                :rem 243
520 FORN=1TO3:PRINTB$"{4 DOWN}":NEXT:PRIN
    TB$"{HOME}";:POKE1983,62         :rem 207
530 PRINT"{RVS}"TAB(8-LEN(STR$(SS)))SS;TA
    B(25)CH;TAB(31)SC:POKE1054,163
                                     :rem 157
535 FORN=0TO11:POKE1123+B(N),60      :rem 18
540 POKE1123+B(N)+G,7:NEXT:FORN=1344TO194
    4STEP200:IFN=1944THEN600         :rem 143
550 FORO=1TO3                        :rem 20
560 R=N+1+INT(RND(1)*38):IFPEEK(R)<>56THE
    N560                             :rem 118
570 FORM=RTOR+160STEP40:POKEM,57:POKEM+G,
    2:NEXT                           :rem 3
575 IFO>1ANDRND(1)<E2THENPOKER+(INT(RND(1
    )*2)+2)*40,63                    :rem 132
580 IFRND(1)<.5ANDPEEK(R-40)=62THENPOKER-
    40,63                            :rem 216
590 NEXT                             :rem 221
600 FORO=1TOE1                       :rem 83
610 R=N+3+INT(RND(1)*36)             :rem 50
615 IFPEEK(R)<>56ORPEEK(R-40)<>62ORPEEK(R
    +1)=62ORPEEK(R-1)=62THEN630      :rem 82
620 POKER,62:POKER-40,63             :rem 142
630 NEXT                             :rem 216
640 FORO=1TO4                        :rem 21
650 R=N-39+INT(RND(1)*38):IFPEEK(R)<>62OR
    PEEK(R+40)=62THEN650             :rem 77
660 POKER,61:POKER+G,0:NEXT:NEXT     :rem 43
670 POKE1122,63:POKE1125,63:POKE1161,63:P
    OKE1166,63                       :rem 129
680 FORN=1304TO1904STEP200:POKEN,63:NEXT:
    FORN=1343TO1943STEP200:POKEN,63:NEXT
                                     :rem 90
685 RETURN                           :rem 131
690 IFSS>=Q*E3THENCH=CH+1:E3=E3+1:PRINT"
    {HOME}{RVS}"TAB(25)CH            :rem 45
700 RETURN                           :rem 119
710 DATA,1,39,40,41,42,78,79,80,81,82,83
                                     :rem 231
720 DATA255,255,153,102,102,153,255,255,1
    95,255,255,195,195,255,255,195,60,60
                                     :rem 31
725 DATA25,255,188                   :rem 180
730 DATA60,36,231,3,4,24,24,60,126,126,60
    ,60,66,165,153,153,165,66,60,,24,36
                                     :rem 195
735 DATA126,126,126                  :rem 220
740 DATA126,,,,,,,,,,,,,,,,,         :rem 58
750 DATA169,,133,1,169,255,141,34,145,169
    ,32,44,31,145,208,5,169,1,133,1,96
                                     :rem 163
755 DATA169,8,44                     :rem 83
760 DATA31,145,208,5,169,2,133,1,96,169,1
    6,44,31,145,208,5,169,3,133,1,96,169
                                     :rem 15
765 DATA4,44,31                      :rem 20
770 DATA145,208,3,133,1,96,169,127,141,34
    ,145,169,128,44,32,145,208,4,169,5
                                     :rem 176
775 DATA133,1,96                     :rem 76
776 FORN=0TO11:READB(N):NEXT         :rem 251
777 RETURN                           :rem 133
780 FORI=0TO11:READX:NEXT:PRINT"{9 DOWN}
    {9 RIGHT}REDEFINING CHARACTERS"
                                     :rem 204
782 POKE56334,PEEK(56334)AND254:POKE1,PEE
    K(1)AND251                       :rem 194
783 FORI=12288TO12288+256*8:POKEI,PEEK(I+
    40960):NEXTI                     :rem 115
784 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334
    )OR1                             :rem 146
785 FORI=12288+56*8TO12288+63*8+7:READM:P
    OKEI,M:NEXTI                     :rem 158
786 POKE53272,(PEEK(53272)AND240)+12
                                     :rem 196
800 RETURN                           :rem 120
1000 PRINT"{CLR}YOUR SCORE IS ";SS
                                     :rem 109
1010 PRINT"{6 DOWN}DO YOU WISH TO PLAY AG
     AIN (Y/N) ?";                   :rem 158
1020 GETA$:IFA$=""THEN1020           :rem 169
1030 IFA$="Y"THENPOKE53272,(PEEK(53272)AN
     D240)+12:GOTO2                  :rem 133
1040 IFA$<>"N"THEN1020               :rem 183
1050 SYS2048                         :rem 147
```

Cave-In For 64

(Article on page 80.)

BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs," "A Beginner's Guide To Typing In Programs," and "The Automatic Proofreader" that appear before the Program Listings.

```
1 POKE56,28:CLR:DEFFNW(X)=PEEK(37151)AND32:B=36865:GOSUB6 :rem 186
2 PRINT"{CLR}{WHT}"SPC(73)"CAVE-IN"SPC(101)"{BLK}F1 FOR HELP":GOTO56 :rem 101
3 A$="+01-22-01+22+01":F=A+VAL(MID$(A$,D*3+1,3))*V :rem 248
4 X=VAL(MID$(A$,D*3+4,3)):L=F+X:R=F-X:RETURN :rem 118
5 PRINTSPC(230)"V":RETURN :rem 194
6 Y=30:POKEB+14,42:POKEB+1,150:GOTO8 :rem 112
7 Y=28:POKEB+14,107:POKEB+1,22:POKEB,25:POKEA,VAL(MID$("235241243242",D*3+1,3)) :rem 146
8 POKE648,Y:IFFNW(W)THENRETURN :rem 249
9 GOTO8 :rem 170
10 PRINTSPC(207)"M{DOWN}{LEFT}[M]{DOWN}{LEFT}N":RETURN :rem 186
11 PRINTSPC(161)"M{DOWN}M{DOWN}{LEFT}[M]{DOWN}{LEFT}[M]{DOWN}{LEFT}[M]{2 DOWN}{2 LEFT}N{UP}N":RETURN :rem 79
12 PRINTSPC(92)W$MID$(X$,37)"{2 DOWN}{3 LEFT}N{UP}N{UP}N":RETURN :rem 81
13 PRINT"{DOWN}{RIGHT}"W$MID$(X$,19)"{2 DOWN}{3 LEFT}N{UP}N{UP}N":RETURN :rem 34
14 PRINT"M"X$"{LEFT}N":RETURN :rem 72
15 PRINTSPC(209)"N{DOWN}{LEFT}[G]{DOWN}{LEFT}M":RETURN :rem 191
16 PRINTSPC(188)"N{UP}N{2 DOWN}{2 LEFT}[G]{DOWN}{LEFT}[G]{DOWN}{LEFT}[G]{DOWN}{LEFT}M{DOWN}M":RETURN :rem 87
17 PRINTSPC(146)"N{UP}N{UP}N{3 DOWN}{3 LEFT}[G]"MID$(Y$,40)W$:RETURN :rem 55
18 PRINTSPC(83)"N{UP}N{UP}N{3 DOWN}{3 LEFT}[G]"MID$(Y$,22)W$:RETURN :rem 8
19 PRINTSPC(20)"N"Y$"M{HOME}":RETURN :rem 93
20 PRINTSPC(229)"P{DOWN}{LEFT}[T]":RETURN :rem 66
21 PRINTSPC(205)"[T]P{DOWN}{LEFT}[M]{DOWN}{2 LEFT}[@]@":RETURN :rem 141
22 PRINTSPC(158)"[2 T]P"MID$(X$,40)"{3 LEFT}[3 T]":RETURN :rem 153
23 PRINTSPC(89)"[2 T]P"MID$(X$,22)"{3 LEFT}[3 T]":RETURN :rem 109
24 PRINT"{DOWN}P"MID$(X$,4)"{LEFT}[T]":RETURN :rem 225
25 PRINTSPC(231)"O{DOWN}{LEFT}[T]":RETURN :rem 63
26 PRINTSPC(210)"O[T]{DOWN}{2 LEFT}[G]{DOWN}{LEFT}L[@]":RETURN :rem 157
27 PRINTSPC(168)"O[2 T]{DOWN}{3 LEFT}[G]"MID$(Y$,43)"[3 T]":RETURN :rem 88
28 PRINTSPC(105)"O[2 T]{DOWN}{3 LEFT}[G]"MID$(Y$,25)"[3 T]":RETURN :rem 80
29 PRINTSPC(42)"O"MID$(Y$,4)"[T]":RETURN :rem 213
30 PRINTSPC(230)"[T]{DOWN}{LEFT}[T]":RETURN :rem 14
31 PRINTSPC(207)"[3 T]{2 DOWN}{3 LEFT}[3 @]":RETURN :rem 237
32 PRINTSPC(161)MID$(Z$,13)SPC(147)MID$(Z$,13):RETURN :rem 104
33 PRINTSPC(92)MID$(Z$,7)"{DOWN}"SPC(251)MID$(Z$,7):RETURN :rem 51
34 PRINTSPC(23)Z$SPC(154)SPC(245)Z$"{HOME}":RETURN :rem 160
35 PRINT"{CLR}{WHT}":FORV=0TO5:GOSUB3:IFPEEK(F)=32THENPRINT"{HOME}":ONVGOSUB34,33,32,31,30:GOTO41 :rem 226
36 PRINT"{HOME}":IFPEEK(L)=32THENONV+1GOSUB14,13,12,11,10,5:GOTO38 :rem 86
37 ONV+1GOSUB24,23,22,21,20,5 :rem 62
38 PRINT"{HOME}":IFPEEK(R)=32THENONV+1GOSUB19,18,17,16,15,5:GOTO40 :rem 112
39 ONV+1GOSUB29,28,27,26,25,5 :rem 89
40 NEXT :rem 163
41 GOSUB77:W=PEEK(37151):IFY=30GOTO44 :rem 251
42 IF(WAND32)=0THENPOKEB,Z:GOSUB6:POKEA,X :rem 134
43 GOTO41 :rem 5
44 IFK=0ANDP<TITHENX=PEEK(B)+1:POKEB,X:P=TI+40:IFX=122THENRETURN :rem 167
45 IF(WAND4)GOTO51 :rem 70
46 V=1:GOSUB3:IFPEEK(F)<>32THENA=F:POKE30720+A,1:GOTO35 :rem 123
47 IFA=7397ANDK=0THENRETURN :rem 236
48 IFPEEK(A)<>13GOTO51 :rem 47
49 K=K-1:PRINTSPC(116)"MAN FOUND"SPC(34)"MEN LEFT="K:POKEA,160 :rem 178
50 IFK=0THENPRINTSPC(72)"CAVE-IN":IFGTHENO=1 :rem 43
51 IF(WAND16)=0THEND=VAL(MID$(D$,D+3,1)):GOTO35 :rem 229
52 POKE37154,127:X=PEEK(37152)AND128:POKE37154,255:IFX=0THEND=VAL(MID$(D$,D+1,1)):GOTO35 :rem 119
53 IF(WAND8)=0THEND=VAL(MID$(D$,D+4,1)):GOTO35 :rem 185
54 IF(WAND32)=0ANDO=0THENZ=PEEK(B):X=PEEK(A):GOSUB7 :rem 201
55 GOTO41 :rem 8
56 D=3:D$="+02-44-02+44":PRINTSPC(91)"{WHT}MINE BEING DUG :rem 131
57 POKE648,28:A=7397:X$="{21 SPACES}":PRINT"{CLR}{RVS}{CYN} "X$;:FORW=1TO21 :rem 142
58 PRINT"{OFF}{BLU}"X$"{CYN}{RVS} ";:NEXT:PRINTX$"{HOME}":POKE7673,160:POKE38393,3:PRINTSPC(141)"{WHT}E :rem 131
59 GOSUB77:X=INT(RND(1)*4):Y=X :rem 83
60 W=A+VAL(MID$(D$,X*3+1,3)) :rem 237
61 IFPEEK(W)=32THENZ=0:POKEW,X:POKEA+VAL(MID$(D$,X*3+1,3))/2,160:A=W:GOTO59 :rem 15
62 X=(X+1)*-(X<3):IFX<>YGOTO60 :rem 4
63 X=PEEK(A):POKEA,160:IFZ=0THENPOKEA,13:Z=1:K=K+1 :rem 76
```

```
64 IFX<>5THENA=A-VAL(MID$(D$,X*3+1,3)):GO
   TO59                              :rem 131
65 W$="M{DOWN}M{DOWN}M":X$="{DOWN}{LEFT}
   ƸMƷ{DOWN}{LEFT}ƸMƷ{DOWN}{LEFT}
   ƸMƷ{DOWN}{LEFT}ƸMƷ{DOWN}{LEFT}
   ƸMƷ{DOWN}{LEFT}ƸMƷ{DOWN}{LEFT}
   ƸMƷ{DOWN}{LEFT}ƸMƷ{DOWN}{LEFT}
   ƸMƷ{DOWN}{LEFT}ƸMƷ{DOWN}{LEFT}
   ƸMƷ{DOWN}{LEFT}ƸMƷ{DOWN}{LEFT}
   ƸMƷ{DOWN}{LEFT}ƸMƷ{DOWN}{LEFT}
   ƸMƷ{DOWN}{LEFT}ƸMƷ{DOWN}{LEFT}
   ƸMƷ{DOWN}{LEFT}ƸMƷ{DOWN}{LEFT}
   ƸMƷ{DOWN}                     :rem 101
66 Y$="{DOWN}{LEFT}ƸGƷ{DOWN}{LEFT}ƸGƷ
   {DOWN}{LEFT}ƸGƷ{DOWN}{LEFT}ƸGƷ
   {DOWN}{LEFT}ƸGƷ{DOWN}{LEFT}ƸGƷ
   {DOWN}{LEFT}ƸGƷ{DOWN}{LEFT}ƸGƷ
   {DOWN}{LEFT}ƸGƷ{DOWN}{LEFT}ƸGƷ
   {DOWN}{LEFT}ƸGƷ{DOWN}{LEFT}ƸGƷ
   {DOWN}{LEFT}ƸGƷ{DOWN}{LEFT}ƸGƷ
   {DOWN}{LEFT}ƸGƷ{DOWN}{LEFT}ƸGƷ
   {DOWN}{LEFT}ƸGƷ{DOWN}{LEFT}ƸGƷ
   {DOWN}{LEFT}ƸGƷ{DOWN}{LEFT}   :rem 31
67 D$="3012301":GOSUB6:Z$="Ƹ19 TƷ":POKE
   A,209:GOSUB35                     :rem 133
68 GOSUB7:POKE217,156:POKE218,156:rem 188
69 PRINT"{HOME}{RVS}{CYN}PRESS THE FIRE B
   UTTON{OFF}{RIGHT}{WHT}TWICE=PLAY - ONC
   E=END                             :rem 91
70 FORW=37933TO38329STEP22:FORX=0TO18:POK
   EW+X,1:NEXT:NEXT                  :rem 187
71 GOSUB77:IFFNW(X)GOTO71             :rem 84
72 GOSUB6:PRINT"{CLR}                 :rem 155
73 IFFNW(X)=0GOTO73                   :rem 157
74 FORW=0TO30:IFFNW(X)=0GOTO88        :rem 143
75 NEXT                               :rem 171
76 POKE56,30:CLR:END                  :rem 194
77 GETA$:IFA$<>CHR$(133)THENRETURN:rem 83
78 POKEB,25:GOSUB6:PRINT"{CLR}PICK ONE
   {BLK}":PRINT"{DOWN} F1=NOVICE":PRINT"
   {DOWN} F3=ADVANCED                 :rem 20
79 PRINT"{DOWN} F5=OLD MAP":PRINT"{DOWN}
   {SPACE}F7=END":PRINT"{DOWN}{WHT}*CURRE
   NT LEVEL                           :rem 56
80 PRINT"{3 DOWN}GOAL-{BLK}FIND THE MINER
   S"SPC(7)"AND GET BACK             :rem 112
81 PRINT"{WHT}{2 DOWN}JOYSTICK-{BLK}MOVE"
   :PRINT"SEE MAP Q↑                 :rem 152
82 PRINT"SEE LEFT<W>SEE RIGHT"SPC(11)"V"S
   PC(18)"SEE BACK{HOME}{WHT}":IFGTHENPRI
   NT"{3 DOWN}*":GOTO84              :rem 88
83 PRINT"{DOWN}*                      :rem 85
84 GETA$:IFA$<>""THENW=ASC(A$)-132:ONABS(
   W)GOTO86,87,68,72                 :rem 118
85 GOTO84                             :rem 18
86 G=0:GOTO88                         :rem 5
87 G=1                                :rem 36
88 O=0:K=0:PRINT"{CLR}":GOTO56        :rem 154
```

Alpha-Shoot

(Article on page 118.)

Program 1: Alpha-Shoot — VIC Version

```
5 PRINT"{CLR}":Y=7900:SV=36878:SS=36876:C
  L=36879:POKECL,78:POKESV,14       :rem 225
7 H=8108:CL=30720:J=37137:POKE650,128:POK
  E651,1                            :rem 130
8 DIM AB(26):FORX=1TO26:AB(X)=32:NEXTX
                                     :rem 25
10 GOSUB200                          :rem 115
12 GOSUB40                           :rem 71
14 GOSUB215:FORX=8120TO8141:POKEX,67:POKE
   X+30720,3:NEXTX                   :rem 225
16 GOSUB250:GOSUB265                 :rem 213
30 GOSUB435:GOTO16                   :rem 89
40 PRINT"{CLR}":RESTORE:Y=7900       :rem 198
42 READL:IFL=0THEN90                 :rem 232
43 POKE Y,L                          :rem 103
50 POKEY,L:READ P:POKESS,P:READ D :rem 48
60 FORX=1TOD:NEXTX:POKESS,0          :rem 54
70 Y=Y+1:FORX=1TO10:NEXTX:IFL=32THENY=Y-1
                                     :rem 161
80 IFY=7922THENY=7952                :rem 253
85 GOTO 42                           :rem 12
90 READA$:IFA$="0"THEN100            :rem 137
92 READP:READD                       :rem 113
95 PRINT"{4 RIGHT}"A$:POKESS,P       :rem 163
97 FORX=1TOD:NEXTX:POKESS,0:FORX=1TO10:NE
   XTX:PRINT"{HOME}":GOTO90          :rem 246
100 FORX=1TO1500:NEXTX:RETURN        :rem 146
170 DATA 1,135,310,2,135,310,3,175,310,4,
    175,310,5,183,310,6,183,310   :rem 239
173 DATA 7,175,615,8,163,310,9,163,310,10
    ,159,310,11,159,310             :rem 131
175 DATA 12,147,120,13,147,120,14,147,120
    ,15,147,120,16,135,602          :rem 5
177 DATA17,175,310,18,175,310,19,163,601,
    20,159,310
178 DATA21,159,310,22,147,601,23,175,121
                                    :rem 221
179 DATA 32,175,121,32,175,231,24,163,601
    ,25,159,310,32,159,310          :rem 26
181 DATA 26,147,605,0               :rem 7
183 DATA"NOW",135,310,"{4 RIGHT}I",135,31
    0,"{6 RIGHT}KNOW",175,310       :rem 11
184 DATA"{11 RIGHT}MY",175,310      :rem 105
185 DATA"{DOWN}{4 RIGHT}A",183,310,"
    {DOWN}{5 RIGHT}B",183,310,"{DOWN}
    {6 RIGHT}C'S",175,610           :rem 158
187 DATA"{2 DOWN}NEXT",163,310,"{2 DOWN}
    {5 RIGHT}TIME",163,310,"{2 DOWN}
    {10 RIGHT}WON'T",159,310        :rem 107
189 DATA"{3 DOWN}YOU",159,310,"{3 DOWN}
    {4 RIGHT}SING",147,310,"{3 DOWN}
    {9 RIGHT}WITH",147,310          :rem 247
190 DATA"{5 DOWN}{5 RIGHT}ME",135,630,"0"
                                    :rem 154
200 PRINTTAB(5):PRINT"ALPHA-SHOOT"
                                    :rem 167
202 LE=-1:KR=0                      :rem 4
204 PRINT"{2 DOWN}WHICH GAME- 1,2,3,OR 4"
                                    :rem 50
205 POKE198,0:WAIT198,1:GETA$:      :rem 235
206 IFA$="1"THEN211                 :rem 1
207 IFA$="2"THENLE=0:GOTO 211       :rem 116
208 IFA$="3"THENLE=1:GOTO 211       :rem 119
209 IFA$="4"THENLE=2:GOTO 211       :rem 122
210 GOTO205                         :rem 99
211 RETURN                          :rem 116
215 R$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
                                    :rem 110
225 PRINT"{CLR}":POKEH,83:POKE36879,10:GO
    SUB228:POKE7703+V,R:RETURN      :rem 245
228 R=INT(LEN(R$)*RND(1)+1):P=ASC(MID$(R$
    ,R,1))-64                       :rem 31
229 IFLE=1THEN232                   :rem 241
```

```
230 R$=LEFT$(R$,R-1)+RIGHT$(R$,LEN(R$)-R)
                                    :rem 31
232 R=P:V=INT(RND(1)*350+1)         :rem 3
236 IFLE=0THENKR=KR+1:R=KR          :rem 144
237 IFLE=1THENWAIT198,1:GETB$:R=ASC(B$)-6
    4                               :rem 251
238 IFR>26ORR<1THENR=1              :rem 115
239 RETURN                          :rem 126
250 POKE37139,0:X=(PEEK(37137)AND60)/4
                                    :rem 96
252 POKE37154,127:J=PEEK(37152)AND128:POK
    E37154,255                      :rem 110
255 IFX=11THEND=-1:GOSUB275         :rem 136
257 IFJ=0 THEND= 1:GOSUB275         :rem 29
259 IFX=7THENGOSUB300               :rem 61
260 RETURN                          :rem 120
265 GETA$:IFA$=""THENGOTO270        :rem 146
266 IFA$="C"THEND=-1:GOSUB275       :rem 188
267 IFA$="B"THEND=+1:GOSUB275       :rem 186
268 IFA$=" "THENGOSUB300            :rem 87
270 RETURN                          :rem 121
275 X=H+D:IFX<8098ORX>8119THENRETURN
                                    :rem 57
276 POKESS,130:POKEH,32:POKEX,83:H=X
                                    :rem 229
278 POKESS,0:RETURN                 :rem 236
300 G=H:FORU=1TO19:G=G-22:IFPEEK(G)<>32TH
    ENPOKEG,32:POKEG+22,32:GOTO350:rem 92
305 POKESS,U+220:POKEG,30:IFU>1THENPOKEG+
    22,32                           :rem 62
306 GOSUB435:NEXTU:POKESS,0:POKEG,32:RETU
    RN                              :rem 73
350 POKESS,0:POKE36877,220:FORL=13TO0STEP
    -1:POKE36878,L:POKE36879,40:GOSUB375
                                    :rem 208
355 NEXTL:POKE36877,0:POKE36878,14:GOSUB3
    90                              :rem 92
357 POKE36879,10:GOSUB228:POKE7703+V,R:RE
    TURN                            :rem 21
375 POKEG,90:POKEG+22,42:POKEG-22,42:POKE
    G+1,42:POKEG-1,42               :rem 140
377 POKEG+23,77:POKEG-23,77:POKEG-21,78:P
    OKEG+21,78                      :rem 209
379 POKEG,32:POKEG+22,32:POKEG-22,32:POKE
    G-1,32:POKEG+1,32               :rem 136
381 POKEG-23,32:POKEG+23,32:POKEG-21,32:P
    OKEG+21,32:RETURN               :rem 192
390 AB(R)=R:FORX=1TO22:POKE8141+X,AB(X):P
    OKE8141+X+30720,7               :rem 149
392 NEXTX:FORX=23TO26:POKE8150+X,AB(X):PO
    KE8150+X+30720,7:NEXT           :rem 42
394 FORX=1TO26:IFAB(X)=32THENRETURN
                                    :rem 254
395 NEXTX:FORX=1TO26:AB(X)=32:NEXTX:POKE3
    6879,78:FORW=1TO1000:NEXTW:GOSUB40:RU
    N                               :rem 127
435 IFLE<2THENRETURN                :rem 57
436 Q=V+7703:IFPEEK(162)<41 THEN RETURN
                                    :rem 5
440 IFQ>8074THENPOKEQ,32:V=2:RETURN
                                    :rem 226
442 POKEQ,32:POKEQ+1,R:V=V+1:POKE162,0
                                    :rem 28
445 RETURN                          :rem 125
```

Program 2: Alpha-Shoot — 64 Version

```
4 POKE53280,6:POKE53281,1:FORT=54272TO542
  96:POKET,O:NEXTT                  :rem 248
5 PRINT"{CLR}":Y=1424:HF=54273:LF=54272:C
  L=54272:POKE54296,15:POKE54277,66
                                    :rem 100
6 POKE54278,68:POKE54284,17:POKE54285,250
  :POKE54279,100:POKE54280,100      :rem 155
7 H=1798:POKE650,128:POKE651,1      :rem 194
8 DIM AB(26):FORX=1TO26:AB(X)=32:NEXTX
                                    :rem 25
10 GOSUB200                         :rem 115
12 GOSUB40                          :rem 71
14 GOSUB215:FORX=1824TO1863:POKEX,67:POKE
   X+54272,3:NEXTX                  :rem 241
16 GOSUB250:GOSUB265                :rem 213
30 GOSUB435:GOTO16                  :rem 89
40 PRINT"{CLR}":RESTORE:POKE53281,1:Y=155
   1                                :rem 139
42 READL:IFL=0THEN90                :rem 232
43 POKE Y,L:POKEY+CL,2              :rem 65
50 READHP:POKEHF,HP:READ LP:POKELF,LP:REA
   DD:POKE54276,17                  :rem 40
60 FORX=1TOD:NEXTX:POKE54276,16     :rem 207
70 Y=Y+1:FORX=1TO10:NEXTX:IFL=32THENY=Y-1
                                    :rem 161
80 IFY=1464THENY=1517               :rem 239
85 GOTO 42                          :rem 12
90 READA$:IFA$="0"THEN100           :rem 137
92 READHP:READLP:READD              :rem 171
95 PRINT"{13 RIGHT}"A$:POKE54276,17:POKEH
   F,HP:POKELF,LP                   :rem 160
97 FORI=1TOD+7:NEXT:POKE54276,16:FORX=1TO
   10:NEXT:PRINT"{HOME}":GOTO90     :rem 50
100 FORX=1TO1500:NEXTX:RETURN       :rem 146
170 DATA 1,34,75,310,2,34,75,310,3,51,97,
    310,4,51,97,310,5,57,172,310    :rem 49
171 DATA6,57,172,310                :rem 7
173 DATA 7,51,97,615,8,45,198,310,9,45,19
    8,310,10,43,52,310,11,43,52,310
                                    :rem 216
175 DATA 12,38,126,120,13,38,126,120,14,3
    8,126,120,15,38,126,120,16,34,75,602
                                    :rem 187
177 DATA17,51,97,310,18,51,97,310,19,45,1
    98,601,20,43,52,310,21,43,52,310
                                    :rem 55
178 DATA22,38,126,601,23,51,97,121
                                    :rem 188
179 DATA 32,51,97,121,32,51,97,231,24,45,
    198,601,25,43,52,310,32,43,52,310
                                    :rem 55
181 DATA 26,38,126,605,0            :rem 155
182 DATA"NOW"                       :rem 237
183 DATA34,75,310,"{4 RIGHT}I",34,75,310,
    "{6 RIGHT}KNOW",51,97,310,"{11 RIGHT}
    MY"                             :rem 45
184 DATA51,97,310                   :rem 121
185 DATA"{DOWN}{4 RIGHT}A",57,172,310,"
    {DOWN}{5 RIGHT}B",57,172,310,"{DOWN}
    {6 RIGHT}C'S",51,97,610         :rem 47
187 DATA"{2 DOWN}NEXT",45,198,310,"
    {2 DOWN}{5 RIGHT}TIME",45,198,310,"
    {2 DOWN}{10 RIGHT}WON'T"        :rem 26
188 DATA43,52,310                   :rem 117
189 DATA"{3 DOWN}YOU",43,52,310,"{3 DOWN}
    {4 RIGHT}SING",38,126,310,"{3 DOWN}
    {9 RIGHT}WITH"                  :rem 94
190 DATA38,126,310,"{5 DOWN}{5 RIGHT}ME",
    34,75,630,"0"                   :rem 28
200 PRINT"{HOME}{10 DOWN}{14 RIGHT}{RVS}
    {BLU}ALPHA SHOOT"               :rem 218
202 LE=-1:KR=0                      :rem 4
```

```
204 PRINT"{HOME}{12 DOWN}{9 RIGHT}WHICH G
    AME-1,2,3,OR 4"                  :rem 244
205 POKE198,Ø:WAIT198,1:GETA$:       :rem 235
206 IFA$="1"THEN211                    :rem 1
207 IFA$="2"THENLE=Ø:GOTO 211        :rem 116
208 IFA$="3"THENLE=1:GOTO 211        :rem 119
209 IFA$="4"THENLE=2:GOTO 211        :rem 122
210 GOTO2Ø5                           :rem 99
211 RETURN                           :rem 116
215 R$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
                                     :rem 11Ø
225 PRINT"{CLR}":POKEH,83:POKE53281,Ø:POK
    E5328Ø,2:GOSUB228:POKE1Ø65+V,R:RETURN
                                     :rem 122
228 R=INT(LEN(R$)*RND(1)+1):P=ASC(MID$(R$
    ,R,1))-64                         :rem 31
229 IFLE=1THEN232                    :rem 241
23Ø R$=LEFT$(R$,R-1)+RIGHT$(R$,LEN(R$)-R)
    :R=P                              :rem 56
232 V=INT(RND(1)*35Ø+1)              :rem 234
236 IFLE=ØTHENKR=KR+1:R=KR           :rem 144
237 IFLE=1THENPOKE198,Ø:WAIT198,1:GETB$:R
    =ASC(B$)-64                       :rem 98
239 IFR>26ORR<ØTHENPOKE198,Ø:GOTO237
                                     :rem 239
24Ø RETURN                           :rem 118
25Ø XV=(PEEK(5632Ø)AND15)            :rem 13Ø
252 JV=15-XV:KV=(PEEK(5632Ø)AND16)
                                     :rem 2Ø8
255 IFJV=4THEND=-1:GOSUB274          :rem 161
257 IFJV=8THEND=1:GOSUB274           :rem 122
259 IFKV=ØTHENGOSUB3ØØ               :rem 127
26Ø RETURN                           :rem 12Ø
265 GETA$:IFA$=""THEN27Ø              :rem 89
266 IFA$="C"THEND=-1:GOSUB274        :rem 187
267 IFA$="B"THEND=1:GOSUB274         :rem 142
268 IFPEEK(197)=6ØTHENPOKE198,Ø:POKE197,Ø
    :GOSUB3ØØ                        :rem 248
27Ø RETURN                           :rem 121
274 H=H+D:IFH<1784THENH=1784         :rem 13Ø
275 IFH>1823THENH=1823                 :rem 3
276 POKEHF,1Ø:POKELF,7Ø:POKEH-D,32:POKEH,
    83:POKEH+CL,1                     :rem 58
277 POKE54276,33:POKE54276,32:RETURN
                                     :rem 133
3ØØ G=H:FORU=1TO19:G=G-4Ø:IFPEEK(G)<>32TH
    ENPOKEG,32:POKEG+4Ø,32:GOTO35Ø:rem 92
3Ø5 POKE54276,17:POKEHF,U+6Ø:POKELF,U+125
    :POKEG,3Ø:IFU>1THENPOKEG+4Ø,32:rem 6Ø
3Ø7 GOSUB435:NEXTU:POKEG,32:POKE54276,16:
    RETURN                           :rem 227
35Ø POKE54283,129:GOSUB374:POKE5328Ø,2:PO
    KE53281,Ø                        :rem 126
355 POKE54283,128:GOSUB39Ø           :rem 239
357 GOSUB228:POKE1Ø65+V,R:RETURN       :rem 9
374 POKE5328Ø,Ø:POKE53281,2          :rem 245
375 POKE54276,16:POKEG,9Ø:POKEG+4Ø,42:POK
    EG-4Ø,42:POKEG+1,42:POKEG-1,42
                                     :rem 144
377 POKEG+41,77:POKEG-41,77:POKEG-39,78:P
    OKEG+39,78:FORL=ØTO3ØØ:NEXT   :rem 1Ø8
379 POKEG,32:POKEG+4Ø,32:POKEG-4Ø,32:POKE
    G-1,32:POKEG+1,32                :rem 136
381 POKEG-41,32:POKEG+41,32:POKEG-39,32:P
    OKEG+39,32:RETURN                :rem 21Ø
39Ø AB(R)=R:FORX=1TO22:POKE1873+X,AB(X):P
    OKE1873+X+54272,7                :rem 167
392 NEXTX:FORX=23TO26:POKE1938+X,AB(X):PO
    KE1938+X+54272,7:NEXT             :rem 64
394 FORX=1TO26:IFAB(X)=32THENRETURN
                                     :rem 254
395 NEXTX:FORX=1TO26:AB(X)=32:NEXTX:FORW=
    1TO1ØØØ:NEXTW:GOSUB4Ø:RUN        :rem 1Ø6
435 IFLE<2THENRETURN                  :rem 57
436 Q=V+1Ø65:IFPEEK(162)<41 THEN RETURN
44Ø IFQ>1742THENPOKEQ,32:V=2:RETURN
                                     :rem 221
442 POKEQ,32:POKEQ+1,R:V=V+1:POKE162,Ø
                                      :rem 28
445 RETURN                           :rem 125
```

Machine Language For Beginners

(Article on page 150.)

BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs," "A Beginner's Guide To Typing In Programs," and "The Automatic Proofreader" that appear before the Program Listings.

Program 1: VIC Version

```
12288   LDY    # Ø
12290   LDA    # 8
12292   STA    37888 ,Y
12295   STA    38144 ,Y
12298   STA    38656 ,Y
12301   INY
12302   BNE           12292
12304   LDY    # Ø
12306   LDA    # 224
12308   STA    4096 ,Y
12311   STA    4580 ,Y
12314   INY
12315   CPY    # 22
12317   BNE           12308
12319   LDA    # 21
12321   STA    71
12323   LDA    # 16
12325   STA    72
12327   LDX    # 24
12329   LDY    # Ø
12331   LDA    # 224
12333   STA  ( 71 ),Y
12335   INY
12336   STA  ( 71 ),Y
12338   DEX
12339   BEQ           12357
12341   CLC
12342   LDA    71
12344   ADC    # 22
12346   STA    71
12348   LDA    72
1235Ø   ADC    # Ø
12352   STA    72
12354   JMP      --->   12329
12357   RTS
```

Program 3: VIC BASIC Loader

```
1ØØØ FORI=12288TO12359:READ DA:POKEI,DA:N
     EXT                             :rem 145
12288 DATA 16Ø,ØØØ,169,ØØ8,153,ØØØ
                                     :rem 131
```

```
12294 DATA 148,153,ØØØ,149,153,ØØØ
                                 :rem 133
123ØØ DATA 151,2ØØ,2Ø8,244,16Ø,ØØØ
                                 :rem 112
123Ø6 DATA 169,224,153,ØØØ,Ø16,153
                                 :rem 131
12312 DATA 228,Ø17,2ØØ,192,Ø22,2Ø8
                                 :rem 127
12318 DATA 245,169,Ø21,133,Ø71,169
                                 :rem 146
12324 DATA Ø16,133,Ø72,162,Ø24,16Ø
                                 :rem 127
1233Ø DATA ØØØ,169,224,145,Ø71,2ØØ
                                 :rem 123
12336 DATA 145,Ø71,2Ø2,24Ø,Ø16,Ø24
                                 :rem 126
12342 DATA 165,Ø71,1Ø5,Ø22,133,Ø71
                                 :rem 127
12348 DATA 165,Ø72,1Ø5,ØØØ,133,Ø72
                                 :rem 131
12354 DATA Ø76,Ø41,Ø48,Ø96,Ø13,Ø13
                                 :rem 138
```

Program 4: 64 BASIC Loader

```
1ØØØ FORI=49152TO49229:READ DA: POKEI,DA:
     NEXT                        :rem 151
49152 DATA 16Ø,ØØØ,169,ØØ8,153,ØØØ
                                 :rem 131
49158 DATA 216,153,ØØØ,217,153,ØØØ
                                 :rem 134
49164 DATA 218,153,ØØØ,219,2ØØ,2Ø8
                                 :rem 138
4917Ø DATA 241,16Ø,ØØØ,169,224,153
                                 :rem 138
49176 DATA ØØØ,ØØ4,153,192,ØØ7,2ØØ
                                 :rem 131
49182 DATA 192,Ø4Ø,2Ø8,245,169,Ø39
                                 :rem 159
49188 DATA 133,Ø71,169,ØØ4,133,Ø72
                                 :rem 151
49194 DATA 162,Ø24,16Ø,ØØØ,169,224
                                 :rem 143
492ØØ DATA 145,Ø71,2ØØ,145,Ø71,2Ø2
                                 :rem 127
492Ø6 DATA 24Ø,Ø16,Ø24,165,Ø71,1Ø5
                                 :rem 136
49212 DATA Ø4Ø,133,Ø71,165,Ø72,1Ø5
                                 :rem 134
49218 DATA ØØØ,133,Ø72,Ø76,Ø44,192
                                 :rem 143
49224 DATA Ø96,Ø13,Ø13,Ø13,Ø13,Ø13
                                 :rem 126
```

Graph Plotter

(Article on page 145.)

Program 1: Graph Plotter—64 Version

```
1ØØ PRINT"{CLR}":POKE53281,Ø:POKE5328Ø,6
                                 :rem 138
11Ø FORI=123ØTO1261:POKEI,114:POKEI+54272
    ,5:NEXT:REM TOP              :rem 114
12Ø FORI=127ØTO179ØSTEP4Ø:POKEI,1Ø7:POKEI
    +54272,5:NEXT:REM LEFT       :rem 88
13Ø FORI=13Ø1TO1821STEP4Ø:POKEI,115:POKEI
    +54272,5:NEXT:REM RIGHT      :rem 161
14Ø FORI=183ØTO1861:POKEI,113:POKEI+54272
    ,5:NEXT:REM BOTTOM           :rem 98
15Ø FORI=1351TO138Ø:POKEI,67:POKEI+54272,
    5:NEXT                       :rem 66
16Ø FORI=1471TO15ØØ:POKEI,67:POKEI+54272,
    5:NEXT                       :rem 64
17Ø FORI=1591TO162Ø:POKEI,67:POKEI+54272,
    5:NEXT                       :rem 71
18Ø FORI=1711TO174Ø:POKEI,67:POKEI+54272,
    5:NEXT:PRINT:PRINT:PRINT     :rem 154
19Ø PRINT: PRINTSPC(4)"15":PRINTSPC(4)"14
    ":PRINTSPC(4)"13":PRINTSPC(4)"12"
                                 :rem 145
2ØØ PRINTSPC(4)"11":PRINTSPC(4)"1Ø":PRINT
    SPC(5)"9":PRINTSPC(5)"8"     :rem 1Ø2
21Ø PRINTSPC(5)"7":PRINTSPC(5)"6":PRINTSP
    C(5)"5":PRINTSPC(5)"4":PRINTSPC(5)"3"
                                 :rem 181
22Ø PRINTSPC(5)"2":PRINTSPC(5)"1":PRINTSP
    C(5)"Ø"                      :rem 82
23Ø PRINTTAB(9)"A"SPC(4)"B"SPC(4)"C"SPC(4
    )"D"SPC(4)"E"SPC(4)"F"       :rem 199
24Ø IFZ$="Y"THEN31Ø              :rem 64
25Ø DATAA,B,C,D,E,F              :rem 34
26Ø FORI=1TO6:READA$(I):NEXTI    :rem 38
27Ø DATA7,6,4,13,8,14            :rem 49
28Ø FORI=1TO6:READD(I):NEXTI     :rem 7
29Ø DATA1792,1797,18Ø2,18Ø7,1812,1817
                                 :rem 116
3ØØ FORI=1TO6:READA(I):NEXTI     :rem 253
31Ø C=54272:FORJ=1TO6            :rem 2Ø2
32Ø FORK=1971TO1975:POKEK,32:NEXTK:PRINT"
    {HOME}":FORL=1TO22:PRINT:NEXTL
                                 :rem 228
33Ø PRINT"VALUE FOR COLUMN ";A$(J);" (Ø-1
    5) ";                        :rem 162
34Ø INPUTA$:IFVAL(A$)=ØTHEN32Ø   :rem 31
35Ø AA=VAL(A$):AA=INT(AA+.5):D=D(J):X=A(J
    )                            :rem 122
36Ø IFAA<ØORAA>15THEN32Ø         :rem 161
37Ø A=A(J)-(AA*4Ø)+8Ø:GOSUB43Ø:NEXTJ
                                 :rem 41
38Ø PRINT"{HOME}":FORI=1TO22:PRINT:NEXT:F
    ORI=1971TO1983:POKEI,32:NEXT :rem 75
39Ø PRINT"WISH TO DO IT AGAIN? (Y/N)"
                                 :rem 65
4ØØ GETZ$:IFZ$=""THEN4ØØ         :rem 125
41Ø IFZ$="Y"THEN1ØØ              :rem 6Ø
42Ø END                          :rem 1Ø9
43Ø IFAA=1THEN53Ø                :rem 221
44Ø IFAA=ØTHEN RETURN            :rem 37
45Ø POKEX,16Ø:POKEX+1,231:POKEX+2,1Ø5
                                 :rem 162
46Ø POKEX+C,D:POKE(X+1)+C,D:POKE(X+2)+C,D
    :IFAA=2THEN49Ø               :rem 3Ø
47Ø FORI=X-4ØTOASTEP-4Ø:POKEI,16Ø:POKEI+1
    ,231:POKEI+2,16Ø             :rem 185
48Ø POKEI+C,D:POKE(I+1)+C,D:POKE(I+2)+C,D
    :NEXTI:GOTO51Ø               :rem 56
49Ø POKEA,247:POKEA+1,2Ø8:POKEA+2,1Ø5
                                 :rem 1Ø7
5ØØ POKEA+C,D:POKE(A+1)+C,D:POKE(A+2)+C,D
    :GOTO53Ø                     :rem 89
51Ø POKEA,247:POKEA+1,2Ø8:POKEA+2,224
                                 :rem 1Ø2
52Ø POKEA+C,D:POKE(A+1)+C,D:POKE(A+2)+C,D
                                 :rem 8Ø
53Ø POKEA-4Ø,233:POKEA-39,16Ø:POKEA-38,2Ø
    6                            :rem 1Ø5
54Ø POKE(A-4Ø)+C,D:POKE(A-39)+C,D:POKE(A-
    38)+C,D                      :rem 172
55Ø RETURN                       :rem 122
```

Program 2:
Graph Plotter—VIC Version

```
100 PRINT"{CLR}":POKE36879,11     :rem 253
110 FORI=7726TO7745:POKEI,114:POKEI+3Ø72Ø
    ,5:NEXTI                      :rem 191
120 FORI=7748TO8Ø34STEP22:POKEI,1Ø7:POKEI
    +3Ø72Ø,5:NEXTI                :rem 94
130 FORI=7767TO8Ø53STEP22:POKEI,115:POKEI
    +3Ø72Ø,5:NEXTI                :rem 96
14Ø FORI=8Ø56TO8Ø75:POKEI,113:POKEI+3Ø72Ø
    ,5:NEXTI                      :rem 187
15Ø FORI=7793TO781Ø:POKEI,64:POKEI+3Ø72Ø,
    5:NEXTI                       :rem 148
16Ø FORI=7859TO7876:POKEI,64:POKEI+3Ø72Ø,
    5:NEXTI                       :rem 164
17Ø FORI=7925TO7942:POKEI,64:POKEI+3Ø72Ø,
    5:NEXTI                       :rem 153
18Ø FORI=7991TO8ØØ8:POKEI,64:POKEI+3Ø72Ø,
    5:NEXTI:PRINT                 :rem 94
19Ø PRINTSPC(Ø)"15":PRINTSPC(Ø)"14":PRINT
    SPC(Ø)"13":PRINTSPC(Ø)"12":PRINTSPC(Ø
    )"11"                         :rem 142
2ØØ PRINTSPC(Ø)"1Ø":PRINTSPC(1)"9":PRINTS
    PC(1)"8":PRINTSPC(1)"7"       :rem 44
21Ø PRINTSPC(1)"6":PRINTSPC(1)"5":rem 17Ø
22Ø PRINTSPC(1)"4":PRINTSPC(1)"3":PRINTSP
    C(1)"2":PRINTSPC(1)"1":PRINTSPC(1)"Ø"
                                  :rem 147
23Ø PRINTTAB(4)"A"SPC(2)"B"SPC(2)"C"SPC(2
    )"D"SPC(2)"E"SPC(2)"F"        :rem 184
24Ø IFZ$="Y"THEN31Ø               :rem 64
25Ø DATAA,B,C,D,E,F               :rem 34
26Ø FORI=1TO6:READA$(I):NEXTI     :rem 38
27Ø DATA7,6,4,3,5,1               :rem 2Ø1
28Ø FORI=1TO6:READD(I):NEXTI      :rem 7
29Ø DATA8Ø35,8Ø38,8Ø41,8Ø44,8Ø47,8Ø5Ø
                                  :rem 113
3ØØ FORI=1TO6:READA(I):NEXTI      :rem 253
31Ø C=3Ø72Ø:FORJ=1TO6             :rem 194
32Ø FORK=8138TO8141:POKEK,32:NEXTK:PRINT"
    {HOME}{19 DOWN}"              :rem 179
33Ø PRINT"COLUMN ";A$(J);" (Ø-15) ";
                                  :rem 62
34Ø INPUT Y$:IFVAL(Y$)=ØTHEN32Ø   :rem 79
35Ø Y=VAL(Y$):Y=INT(Y+.5):D=D(J):X=A(J)
                                  :rem 23
36Ø IFY<ØORY>15THEN32Ø            :rem 79
37Ø A=A(J)-(Y*22)+44:GOSUB43Ø:NEXTJ
38Ø PRINT"{HOME}{2Ø DOWN}"        :rem 211
39Ø PRINT"DO IT AGAIN? (Y/N)"     :rem 99
4ØØ GETZ$:IFZ$=""THEN4ØØ          :rem 125
41Ø IFZ$="Y"THEN1ØØ               :rem 6Ø
42Ø END                           :rem 1Ø9
43Ø IFY=1THEN53Ø                  :rem 18Ø
44Ø IFY=ØTHENRETURN               :rem 252
45Ø POKEX,16Ø:POKEX+1,231:POKEX+2,1Ø5
                                  :rem 162
46Ø POKEX+C,D:POKE(X+1)+C,D:POKE(X+2)+C,D
    :IFY=2THEN49Ø                 :rem 245
47Ø FORI=X-22TOASTEP-22:POKEI,16Ø:POKEI+1
    ,231:POKEI+2,16Ø              :rem 185
48Ø POKEI+C,D:POKE(I+1)+C,D:POKE(I+2)+C,D
    :NEXTI:GOTO51Ø                :rem 56
49Ø POKEA,227:POKEA+1,2Ø8:POKEA+2,1Ø5
                                  :rem 1Ø5
5ØØ POKEA+C,D:POKE(A+1)+C,D:POKE(A+2)+C,D
    :GOTO53Ø                      :rem 89
51Ø POKEA,227:POKEA+1,2Ø8:POKEA+2,224
                                  :rem 1ØØ
52Ø POKEA+C,D:POKE(A+1)+C,D:POKE(A+2)+C,D
                                  :rem 8Ø
53Ø POKEA-22,233:POKEA-21,16Ø:POKEA-2Ø,2Ø
    6                             :rem 87
54Ø POKE(A-22)+C,D:POKE(A-21)+C,D:POKE(A-
    2Ø)+C,D                       :rem 154
55Ø RETURN                        :rem 122
```

64 BASIC Aid

(Article on page 156.)

BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs," "A Beginner's Guide To Typing In Programs," and "The Automatic Proofreader" that appear before the Program Listings.

```
39852 :173,254,159,133,Ø55,173,Ø95
39858 :255,159,133,Ø56,169,Ø76,ØØ2
39864 :133,124,173,217,155,133,Ø95
3987Ø :125,173,218,155,133,126,Ø96
39876 :Ø76,143,156,24Ø,ØØ3,Ø76,122
39882 :ØØ8,175,169,2Ø1,133,124,244
39888 :169,Ø58,133,125,169,176,Ø14
39894 :133,126,Ø96,219,155,133,Ø52
399ØØ :139,134,151,186,189,ØØ1,252
399Ø6 :ØØ1,2Ø1,14Ø,24Ø,Ø16,2Ø8,ØØ8
39912 :ØØ2,164,14Ø,166,151,165,252
39918 :139,2Ø1,Ø58,176,ØØ3,Ø76,123
39924 :128,ØØØ,Ø96,189,ØØ2,ØØ1,148
3993Ø :2Ø1,164,2Ø8,237,165,139,Ø84
39936 :Ø16,ØØ2,23Ø,122,132,14Ø,13Ø
39942 :162,ØØØ,134,165,2Ø2,232,133
39948 :164,122,185,ØØØ,ØØ2,Ø56,Ø29
39954 :253,217,159,24Ø,Ø19,2Ø1,Ø83
3996Ø :128,24Ø,Ø19,23Ø,165,232,Ø14
39966 :189,216,159,Ø16,25Ø,189,Ø25
39972 :217,159,2Ø8,228,24Ø,191,255
39978 :232,2ØØ,2Ø8,224,132,122,136
39984 :165,165,Ø1Ø,17Ø,189,245,224
3999Ø :159,Ø72,189,244,159,Ø72,181
39996 :Ø32,233,155,Ø76,115,ØØØ,159
4ØØØ2 :Ø32,178,157,165,Ø95,166,Ø91
4ØØØ8 :Ø96,133,Ø36,134,Ø37,Ø32,Ø28
4ØØ14 :Ø19,166,165,Ø95,166,Ø96,Ø17
4ØØ2Ø :144,Ø1Ø,16Ø,ØØ1,177,Ø95,159
4ØØ26 :24Ø,ØØ4,17Ø,136,177,Ø95,144
4ØØ32 :133,122,134,123,165,Ø36,Ø41
4ØØ38 :Ø56,229,122,17Ø,165,Ø37,113
4ØØ44 :229,123,168,176,Ø3Ø,138,2Ø4
4ØØ5Ø :Ø24,1Ø1,Ø45,133,Ø45,152,1Ø2
4ØØ56 :1Ø1,Ø46,133,Ø46,16Ø,ØØØ,Ø94
4ØØ62 :177,122,145,Ø36,2ØØ,2Ø8,246
4ØØ68 :249,23Ø,123,23Ø,Ø37,165,142
4ØØ74 :Ø46,197,Ø37,176,239,Ø32,Ø97
4ØØ8Ø :Ø51,165,165,Ø34,166,Ø35,248
4ØØ86 :Ø24,1Ø5,ØØ2,133,Ø45,144,Ø91
4ØØ92 :ØØ1,232,134,Ø46,Ø32,Ø89,178
4ØØ98 :166,Ø76,131,164,Ø32,124,Ø87
4Ø1Ø4 :165,Ø32,115,ØØØ,133,139,24Ø
4Ø11Ø :162,ØØØ,134,Ø73,Ø32,14Ø,2Ø3
4Ø116 :157,165,165,2Ø1,ØØØ,2Ø8,Ø52
4Ø122 :ØØ7,162,ØØ2,134,Ø73,Ø32,Ø84
4Ø128 :14Ø,157,Ø32,115,ØØØ,24Ø,1Ø8
```

```
40134 :003,032,253,174,032,178,102
40140 :157,165,095,166,096,133,248
40146 :122,134,123,032,215,170,238
40152 :208,011,200,152,024,101,144
40158 :122,133,122,144,002,230,207
40164 :123,032,202,159,240,005,221
40170 :032,220,157,176,003,076,130
40176 :143,156,132,085,230,085,047
40182 :164,085,166,049,165,050,157
40188 :133,139,177,122,240,216,255
40194 :221,000,002,208,237,232,134
40200 :200,198,139,208,241,136,106
40206 :132,011,132,151,165,073,166
40212 :240,091,032,240,157,165,177
40218 :052,056,229,050,133,167,201
40224 :240,040,200,240,202,177,107
40230 :122,208,249,024,152,101,126
40236 :167,201,002,144,064,201,055
40242 :075,176,060,165,167,016,197
40248 :002,198,139,024,101,011,019
40254 :133,151,176,005,032,036,083
40260 :158,240,003,032,012,158,159
40266 :165,151,056,229,052,168,127
40272 :200,165,052,240,015,133,117
40278 :140,166,051,189,000,002,122
40284 :145,122,232,200,198,140,105
40290 :208,245,024,165,045,101,118
40296 :167,133,045,165,046,101,249
40302 :139,133,046,165,122,166,113
40308 :123,133,095,134,096,166,095
40314 :067,165,068,032,061,158,161
40320 :032,225,255,169,000,133,174
40326 :198,164,151,076,242,156,097
40332 :164,122,200,148,049,169,224
40338 :000,149,050,185,000,002,020
40344 :240,021,197,139,240,005,226
40350 :246,050,200,208,242,132,212
40356 :122,096,201,171,240,004,230
40362 :201,045,208,001,096,076,029
40368 :008,175,144,005,240,003,239
40374 :032,166,157,032,107,169,077
40380 :032,019,166,032,121,000,046
40386 :240,011,032,166,157,032,064
40392 :115,000,032,107,169,208,063
40398 :224,165,020,005,021,208,081
40404 :006,169,255,133,020,133,160
40410 :021,096,032,202,159,133,093
40416 :067,032,202,159,133,068,117
40422 :056,165,020,229,067,165,164
40428 :021,229,068,096,165,122,169
40434 :133,034,165,123,133,035,097
40440 :165,045,133,036,165,046,070
40446 :133,037,096,165,034,197,148
40452 :036,208,004,165,035,197,137
40458 :037,096,164,011,200,177,183
40464 :034,164,151,200,145,034,232
40470 :032,001,158,208,001,096,006
40476 :230,034,208,236,230,035,233
40482 :208,232,164,011,177,036,094
40488 :164,151,145,036,032,001,057
40494 :158,208,001,096,165,036,198
40500 :208,002,198,037,198,036,219
40506 :076,036,158,160,000,132,108
40512 :165,132,015,032,205,189,034
40518 :169,032,164,165,041,127,000
40524 :032,210,255,201,034,208,248
40530 :006,165,015,073,255,133,217
40536 :015,200,177,095,240,025,072
40542 :016,236,201,255,240,232,250
```

```
40548 :036,015,048,228,132,165,212
40554 :032,124,158,200,177,174,203
40560 :048,214,032,210,255,208,055
40566 :246,032,215,170,056,096,165
40572 :160,157,132,174,160,160,043
40578 :132,175,056,233,127,170,255
40584 :160,000,202,240,238,230,182
40590 :174,208,002,230,175,177,084
40596 :174,016,246,048,241,032,137
40602 :107,169,165,020,133,053,033
40608 :165,021,133,054,032,253,050
40614 :174,032,107,169,165,020,065
40620 :133,051,165,021,133,052,215
40626 :032,142,166,032,202,159,143
40632 :032,202,159,208,033,032,082
40638 :172,159,032,202,159,032,178
40644 :202,159,208,003,076,143,219
40650 :156,032,202,159,165,099,247
40656 :145,122,032,202,159,165,009
40662 :098,145,122,032,183,159,185
40668 :240,226,032,202,159,032,087
40674 :202,159,032,202,159,201,157
40680 :034,208,011,032,202,159,110
40686 :240,197,201,034,208,247,085
40692 :240,238,170,240,188,016,056
40698 :233,162,004,221,212,159,217
40704 :240,005,202,208,248,240,119
40710 :221,165,122,133,059,165,103
40716 :123,133,060,032,115,000,219
40722 :176,211,032,107,169,032,233
40728 :081,159,165,060,133,123,233
40734 :165,059,133,122,160,000,157
40740 :162,000,189,000,001,201,077
40746 :048,144,017,072,032,115,214
40752 :000,144,003,032,130,159,004
40758 :104,160,000,145,122,232,049
40764 :208,232,032,115,000,176,055
40770 :008,032,145,159,032,121,051
40776 :000,144,248,201,044,240,181
40782 :184,208,150,032,172,159,215
40788 :032,202,159,032,202,159,102
40794 :208,008,169,255,133,099,194
40800 :133,098,048,014,032,202,111
40806 :159,197,020,208,015,032,221
40812 :202,159,197,021,208,011,138
40818 :032,209,189,169,032,076,053
40824 :210,255,032,202,159,032,242
40830 :183,159,240,210,032,162,088
40836 :159,230,151,032,036,158,130
40842 :230,045,208,002,230,046,131
40848 :096,032,162,159,198,151,174
40854 :032,012,158,165,045,208,002
40860 :002,198,046,198,045,096,229
40866 :032,240,157,160,000,132,115
40872 :011,132,151,096,165,053,008
40878 :133,099,165,054,133,098,088
40884 :076,142,166,165,099,024,084
40890 :101,051,133,099,165,098,065
40896 :101,052,133,098,032,202,042
40902 :159,208,251,096,160,000,048
40908 :230,122,208,002,230,123,095
40914 :177,122,096,137,138,141,253
40920 :167,067,072,065,078,071,224
40926 :197,068,069,076,069,084,017
40932 :197,070,073,078,196,075,149
40938 :073,076,204,078,085,077,059
40944 :066,069,210,000,165,156,138
40950 :065,156,165,156,198,155,117
40956 :152,158,172,013,013,013,005
```

PRODUCT MART

VIC 20™/COMMODORE 64™

Ridge Runner — for the unexpanded VIC 20 on tape or disk format. 100% machine language. Maneuver your multicolor U.F.O. through a field of blinking mines while blasting spinning asteroids and various enemy ships with laser fire. Features include a horizontally scrolling playfield, hi-resolution/multicolor graphics, excellent sound, high score, pause button, bonus ships, and ever increasing levels of difficulty. Joystick required. Price: $14.95

Dungeons — for the VIC 20 with 16K expansion and tape or disk. Create characters to explore a twelve level dungeon which contains 1200 individual rooms. After you purchase your weapon and armor, you will find vast treasures and do battle with over fifty types of monsters which you must slay for experience points. Your character also has the ability to cast numerous spells and you are given the option of saving the game to tape or disk as your character gains strength and experience. Excellent sound and three dimensional graphics add to the excitement. Price: $14.95

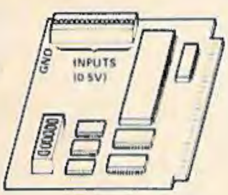
Pak Alien — for the unexpanded VIC 20 with tape or disk. 100% machine language arcade-style game. Custom graphic characters and 100 levels of increasing difficulty. Guide your alien through a maze of interplanetary space particles dodging the seven evil aliens and clear the board before the bonus timer runs out. Joystick or keyboard. Includes pause feature. Price: $14.95

BYTES and BITS
524 E. Canterbury Ln.
Phoenix, AZ 85022
(602) 942-1475
Please specify tape or disk
Add $2.00 for postage and handling

VIC 20 and Commodore 64 are trademarks of Commodore Electronics Ltd.

MULTIPLEXED DIGITAL VOLTMETER FOR THE VIC-20 AND C-64

INTRODUCTORY OFFER
$79
Expires April 84



FEATURES

- 16 channel Analog to Digital converter (0-5V range)
- High accuracy (1/2% of full scale)
- Simply uses BASIC statements (one POKE and PEEK)
- Compatible with other I/O devices (switchable address)
- Can be used in both VIC-20 and C-64
- Can be daisy chained together for up to 256 channels

Great for engineering experiments, Science fair projects, Household monitoring of temperatures, light switches, smoke detectors, alarm systems, or invent your own uses.

Send check or money order
DRIVER ELECTRONICS
194-B Gladys Ave.
Mountain View, CA 94043

California residents add 6.5% sales tax.
Send self-addressed envelope for free brochure.

VIC & 64

AT LAST! The information you need, without always going back to the manual. These durable plastic coated overlays contain program starting locations, function key labeling, commands and additional aids in center cutout.

Please send me the following Leroy's Cheatsheet™ keyboard overlays

20	64		20	64	
□		Programmer's Aid¹	□	□	Graphic printer (1515 & 1525)¹
□		Vicmon¹	□		UMI Wordcraft 20²
□		Super Expander¹	□		HES Vic Forth³
□		Vic Typewriter¹	□	□	HES Writer³
□		Victerm 1¹		□	Wordpro 3 plus
	□	Term 64¹		□	Easy Script¹
□	□	Quick Brown Fox	□	□	Basic

CG184

Send check or money order plus $1.00 (postage and handling)
PA residents add 6% sales tax.

Name ____________
Address ____________
City ________ State ________ Zip ________

(1) Products of Commodore Business Machines, Inc. (2) Product of United Microware Industries, Inc. (3) Products of Human Engineered Software. (4)Cindy is a Trademark of Commodore Business Machines, Inc.

CHEATSHEET PRODUCTS™
P.O. Box 8299 Pittsburgh, PA. 15218 (412) 456-7420

for Commodore 64™ and VIC 20™
Creative and hilarious FUN HOUSE

- Great as a gift and for having fun at parties
- Has color graphics, fun sounds & accepts words
- Find special keys for extra fun

On cassette. Just $22. get $2 of with the word "GAZETTE" in your order. Specify computer. Send check or money order

To: CONTINENTAL CONCEPTS
P.O. Box 7631 NEWPORT BEACH, CA 92660

CA add Sales Tax. Postage extra if fast delivery desired.
VIC 20 version uses 16K expansion. Order Today!
"VIC 20" and "Commodore 64" are trademarks of Commodore Business Machines

VIC 20 C64

1983 TAX RETURN HELPER

Fast and easy income tax preparation.

- Form 1040 and Schedules A,B,C,D,E
- Enter and modify data on a screen copy of the form.
- Works like a spreadsheet - all the lines affected by a change are instantly updated.
- Form 1040 and Schedule A are automatically corelated.
- Price is **tax deductible.** Cassette $23 (+ $1.50 shipping & handling). Specify VIC 20 (16K RAM) or C64.
- Check, MO or credit card.

KSOFT CO.
845 WELLNER RD.
NAPERVILLE, IL 60540
(312) 961-1250

Dealer inquiries welcome

FREE CATALOG!
HOME, EDUCATIONAL, AND BUSINESS SOFTWARE FOR THE VIC AND 64

New Items...
Checkbook/64 (Disk) Handles all checking account data. $16.00
Capitals/64 Teaches U.S. and Foreign capitals. Disk $10.00 Tape $8.00

Over 50 other titles! Low-Priced! Practical! Put your VIC or 64 to work with quality software from

Farthest Fringe S.A.
101 Highway Blvd.
N. Pekin, IL 61554

'PUBLIC DOMAIN'™ — SOFTWARE —

Supporting all COMMODORE computers
Written by users, for users
★ GAMES ★ UTILITIES ★ EDUCATIONAL ★

VIC 20™
collection #1 – collection #2 – collection #3
collection #4 – collection #5 – collection #6
70+ programs per collection – Tape/Disk – $10.00

COMMODORE 64™
64 collection #1 – 64 collection #2 – 64 collection #3
64 collection #4 – 64 collection #5
25+ programs per collection – Tape/Disk – $10.00

PET® / CBM®
5 Utility – Tapes/Disks – $10.00 each
11 Game – Tapes/Disks – $10.00 each
8 Educational – Tapes/Disks – $10.00 each

DINSET™: Reset Switch
Works on Vic 20 or Commodore 64 – $5.00

All prices include shipping and handling.
CHECK, MONEY ORDERS, VISA and MASTERCARD accepted.

For A Free Catalog Write:

Public Domain, Inc.
5025 S. Rangeline Rd., W. Milton, OH 45383
10:00 a.m. - 5:00 p.m. EST – Mon. thru Fri.
(513) 698-5638 or (513) 339-1725

VIC 20™, C64™ and Commodore 64™ are Trademarks of Commodore Electronics Ltd. PET® is a Registered Trademark of Commodore Business Machines, Inc.

VIC-20 USERS
Cartridge Back-Up

- SYSTEM IS AN EASY TO USE PROGRAM AND A HIGH QUALITY CARTRIDGE INTERFACE BOARD
- BACK-UP YOUR CARTRIDGES ONTO CASSETTE OR DISK
- 8K RAM REQUIRED

$49.95 POST PAID

Cassette Back-Up

- EASY TO USE PROGRAMS
- BACK-UP ANY CASSETTE TAPE ONTO TAPE OR DISK
- REQUIRES **NO USER MEMORY**

$19.95 POST PAID

VISA/MASTERCARD ORDERS:
PHONE (215) 269-4803

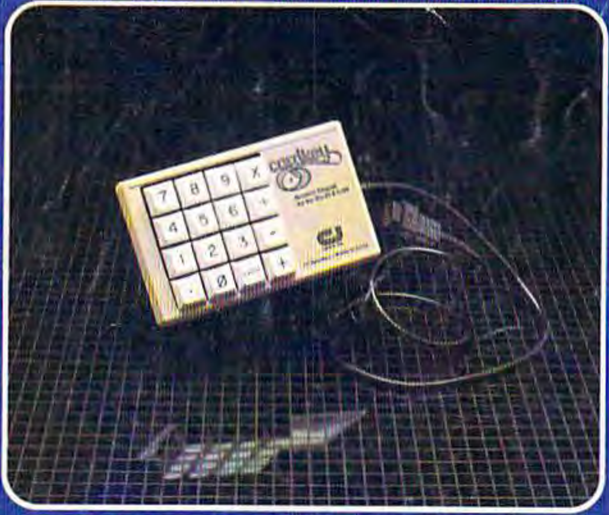
MAIL CHECK OR MONEY ORDER TO:
E-M TECHNOLOGIES
P.O. BOX 185
DOWNINGTOWN, PA. 19335
PA. RESIDENTS ADD 6%
6 MONTH REPLACEMENT GUARANTEE

ADVERTISERS INDEX

Cardco Presents

16 Key Keypad for the VIC-20™ and Commodore 64™

The CARDKEY/1 provides an additional programable 16 keys to either the VIC or 64. This is a separate pad that plugs into the joystick port of your computer. It comes with tape programs (transferable to disk) that allow the user to define the 16 keys as any number, letter or character string value. It additionally allows the keys to be defined as strings. Another function available is the direct calculator mode in which you use your computer as a calculator. For example you press 5 + 5 (enter) and the computer prints 10 on the screen.



Suggested retail: $39.95

See us at
COMDEX™
Fall '83
November 28-December 2, 1983
Las Vegas Convention Center
Las Vegas, Nevada

The CARDKEY features:

- 16 keys laid out in a calculator format
- All keys are user definable
- Keys can be defined as any character
- Keys can be defined as complete strings
- Optional calculator mode

See a complete line of American made Cardco Products at a computer store near you, today.

313 Mathewson • Wichita, Kansas 67214 • (316) 267-6525

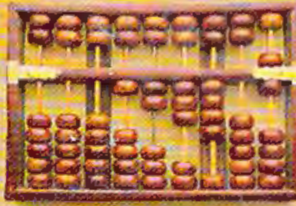
cardco, inc.

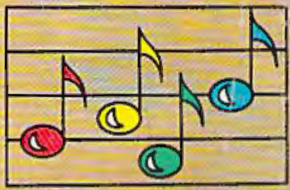
It writes, rates, creates, even telecommunicates. Costs less, does more- the Commodore 64.

When Commodore introduced the 64, the industry suddenly realized that there would be a computer in every home, school and business years before anyone ever dreamed.

That's because Commodore 64 halved the price of high technology: while you can compare the 64's capabilities with those of any sophisticated business PC, you can compare its price with that of an average television.

What can you do with it? Create with its high resolution Sprite Graphics. Add a printer and type with it. Add a disk drive to use spread sheets and other financial programs. Learn and play music through your home sound system on the 64's professional quality music synthesizer.





Add a modem, and hook up with the vast computer networks through your telephone. In short, the Commodore 64 is the ultimate personal computer, at a price you can afford.

COMMODORE 64