

# COMPUTER'S GAZETTE™

\$2.00  
August 1983  
Issue 2 Vol. 1, No. 2  
63380 \$2.50 in Canada

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

## YOUR FIRST HOUR WITH A COMPUTER

The box is opened, the computer unwrapped, and suddenly there are a hundred things you never thought of. Here's how to make that first hour more enjoyable.



## THE VIPER

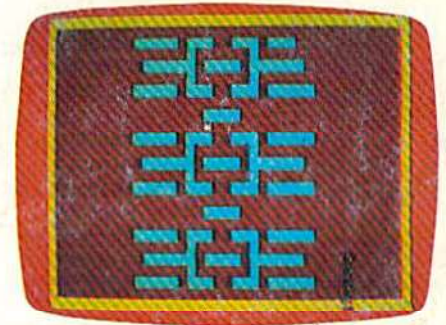
For VIC-20 And 64

No matter how much it devours, The Viper is never satisfied...it just keeps growing. An electrifying game with skill levels to challenge everyone.

## Playing Computer Music



This month's Beginner's Corner shows how to make VIC-20 music easy — and fun.



**Also In This Issue**  
**New Products For Commodore At Comdex/Spring Show**

**Computing For Grownups: Revolution In The Nursery**  
**Mailing List For VIC And 64**

**Commodore 64 Hi-Res Graphics Made Simple**

Puzzled over hard-to-use high-resolution graphics? Puzzle no more. This groundbreaking program adds easy graphics commands that BASIC left out.





# THE VIPER

Dave and Casey Gardner

"The Viper" is a fast-action game with 27 difficulty levels for the unexpanded VIC-20. A translated version for the Commodore 64 has 60 difficulty levels. A joystick is required.

The Viper is a fast, furious, *hungry* snake. It races about, devouring its favorite food – asterisks! And the more it eats, the bigger it gets. Since snakes have a hard time growing wider, the Viper simply gets longer. Since the Viper has such sharp,

venomous teeth, it must not in its haste accidentally run into its own lengthening body. To make things especially interesting, the Viper must maneuver through a maze with electric walls. One false move means certain doom.

With a joystick and a VIC-20 or Commodore 64, you can experience the perils of the Viper. The program is easy to set up and play. Just follow the screen instructions. Maneuver the Viper with a joystick plugged into port one. Don't leave the Play button on the cassette recorder pressed after the program is loaded, or the joystick may not respond properly. Sometimes pressing RUN/STOP and RESTORE before playing helps too, especially for disk users.

You can choose from various difficulty levels to control the Viper's speed – nine levels on the VIC and 20 levels on the 64. You also

select one of three courses – no maze, the easy maze, or the hard maze. Your score is the number of those delicious asterisks eaten multiplied by the skill level you selected, so the harder the game, the more possible points. You get twice as many points for the easy maze, and five times as many for the hard maze.

## Available On Tape

If you don't want to type this program, the authors will save it on tape for you (VIC version only). Send a blank cassette, self-addressed stamped mailer, and \$3 to:

Dave Gardner  
2342 Barnes Road  
Walworth, NY 14568

## A Word To Programmers

The VIC version uses a clever joystick routine published in the October 1982 **COMPUTE!** Magazine. Since the routine is written in machine language, it speeds up the already fast response of the VIC-20. The Commodore 64, however, is "burdened" with more memory and graphics power, which ironically tend to make games run somewhat more slowly. To compensate, the entire main loop of the program was translated into machine language for the 64. The resulting speed was so fast that delay loops had to be inserted just to slow it down to a barely playable level. If you're brave enough, try level 20 – you'll never be able to play it. If anyone can score any points on level 20 with the hard maze, it would be truly miraculous.

Also, the 64 version was changed so you don't have to use the key-



# Father-Son Programming

John Blackford, Assistant Features Editor

"The Viper" was created by a father-son programming team, Dave Gardner and his son Casey, age 11. They've had their VIC-20 only a little more than a year, but they are planning to write even more programs together, using any proceeds to pay for new computer equipment. Their earnings for The Viper, for example, will help purchase a disk drive.

Gardner learned BASIC on the VIC and decided to try his hand at game programming to hone his skills. The idea of making a snake that grows longer as it eats occurred to him after watching a similar game on an Apple computer. Gardner did the actual programming, while Casey dreamed up new ideas and helped with the design.

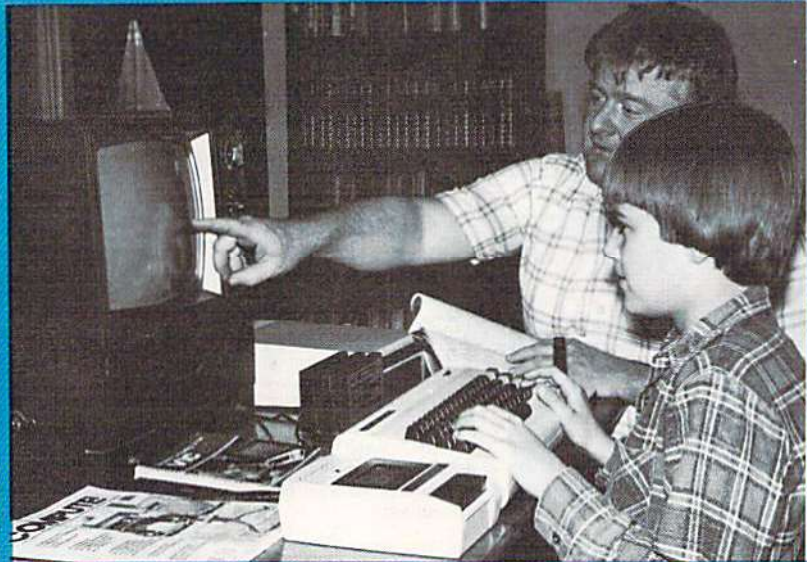
Gardner didn't plan how he was going to implement the game. "I just sat down at the keyboard and started working on it," he says. "First I got the screen set up, then defined the snake." Next came the snake's movement and the interaction with the asterisks it gobbles up. After Gardner had refined the movement of the Viper, he added skill levels by making the snake move faster in each of nine stages. Then he worked out scoring routines that award more points in the higher levels.

As the game progressed, Casey and his father would talk about it and try to improve the concept. Casey came up with the notion of having a maze for the snake to move through and that became his special project.

After his father showed him how to use graph paper to represent the computer's

screen, Casey began to draw various possible mazes. "He did about eight designs," says Gardner. "We talked over what would be the most playable – and the most achievable with the computer." Then, they chose the version that seemed best.

Once they had the basic game, they decided to increase the number of skill levels by offering two mazes, one easy, the other hard. Including the option of no maze at all,



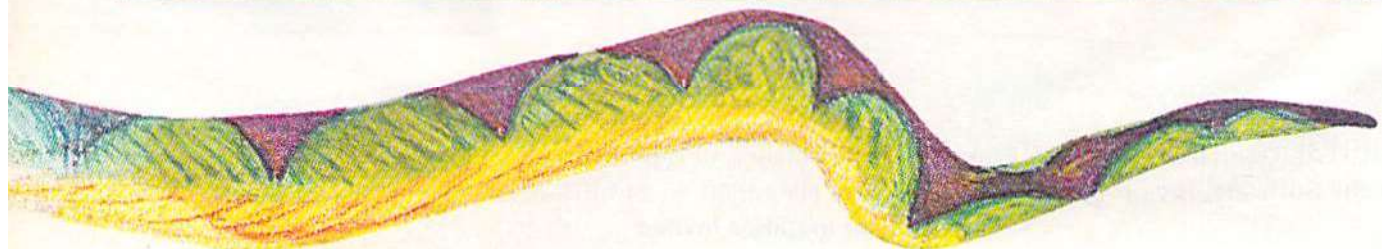
Dave Gardner and his son, Casey.

there are a total of 27 skill levels.

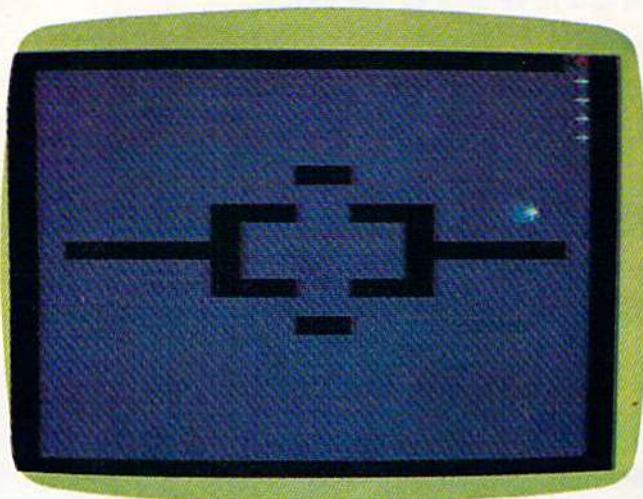
The whole process of creating the game – including writing the program description – took only about two weeks, working an hour or two on weekdays and six or more hours each day on the weekends.

Creating the game was intense, but that has only whetted their appetite. Their current project is an adventure game with quality graphics. Casey has shown particular aptitude for the visual side. "He tends more toward the artistic aspects," says Gardner. Casey, who likes to sit at the computer and work out new character sets, has already designed the title page for the new game.

They haven't decided what they'll do after that, but they enjoy working together and hope to produce many more games as a team.







The Viper has just crashed into a wall in the upper-right corner - VIC-20 version, easy maze.

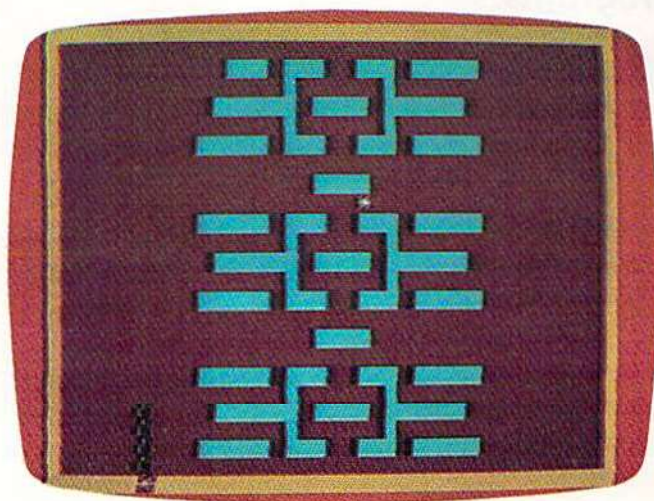
board after typing RUN. This lets you sit back in your chair and make all your selections (such as skill level) with the joystick. This technique may be handy when you write your next program.

Another feature in the 64 version is the word "VIPER" that moves about on the title screen. No, it's not high-resolution graphics, and it's not made of sprites, but rather from simple character graphics found on the keyboard. The movement works with programmable INSerts and DELets. Again, look it over. You may be able to use the technique for animation in your next game.

## Breakdown Of VIC Version

### Line(s)

- 10-60 Initialize variables, DIMension arrays, POKE machine language routine into cassette buffer, set screen and border colors, GOSUB to title page and instructions.
- 70-100 Place border.
- 110-160 Randomly place first asterisk and Viper.
- 170-200 Read joystick.
- 210 Check if Viper has hit anything.



The Viper collides with the lower wall - Commodore 64 version, hard maze.

- 220 Check if asterisk is missing.
- 240 Check if Viper has eaten asterisk.
- 250 Place new asterisk.
- 260-320 Update tail location.
- 330 Viper noise.
- 350 Move Viper.
- 360 Control Viper's speed.
- 370-420 Title page.
- 430-560 Display instructions and set skill level.
- 570 Place hard maze.
- 590 Place easy maze.
- 600-690 End of game, display score, start over.
- 700-720 Find location for new asterisk.
- 730-750 Machine language routine.
- 760-780 Maze DATA.

See program listings on page 116. ☺

## FOXSOFT™ Presents... Spryte Byter™

For the Commodore 64™

The user affectionate sprite development program. Menu-driven, mono/multicolor sprites, joystick/keyboard, tape/disk. 20K w/FAST machine language routines. Over 60 commands: ROTATE (any angle 0 - 360), INVERT/OBVERT, SHIFT, SYMMETRY, AND/OR, REVERSE, REVIEW, MOVIE (animation). Create and edit up to 128 sprites per file. For programming efficiency and FUN! Includes the Game Maker - automatically prepares a base for game development.

Cassette \$29.95 (U.S. Funds)  
Disk \$34.95

### FOXSOFT™

P.O. Box 507  
Deer Park, Texas 77536  
(713) 473-6723



A Division of Foxfire Systems, Inc.

## Commodore 64™ & Apple II™ Assembly Language Debugger



If you are developing software for the Commodore 64 or Apple II or are interested in assembly language programming, you needed this finely-crafted aid "yesterday." All the features you'd expect in an advanced debugger are included - step, breakpoints, windowing, super-complex conditional breaks, symbolic debugging, patch, show prior 128 steps, and compiled BASIC-like language. Supplied complete with a detailed instruction manual. A new data sheet is available on request.

PTD-6510 Debugger for Commodore 64 ..... \$65.10  
PTD-6502 Debugger for Apple II and IIe ..... \$49.95  
Manual (only) for above, each ..... \$10.00  
DisKit 64 Fast single disk copy  
(+utilities) for Commodore 64 ..... \$75.00

\*PTD-6510 (Commodore) requires 1541 disk drive.  
PTD-6502 (Apple) requires DOS 3.3, 48K.



## PTERODACTYL SOFTWARE™

200 Bolinas Road #27, P.O. Box 538  
Fairfax, CA 94930 (415) 485-0714



# Cylon Zap

## For VIC-20 And Commodore 64

Mark Dudley

"Cylon Zap" is an arcade-style game for the 8K expanded VIC-20 and the Commodore 64. A space station in the center of the screen, which you must defend at all costs, is attacked continually by Cylon ships. You must shoot them before they dive (kamikaze-style) into the space station.



To defend against the Cylons, you have two weapons. First, the joystick is moved up, down, right, or left to fire lasers in any of these four directions. Second, the fire button detonates a "smart bomb," which immediately clears the screen of all visible attackers. Smart bombs should be used sparingly, for there are only three available at the beginning of play.

The score and the number of remaining bombs are continually updated at the upper-left corner of the screen. When the score reaches 30, the flank attackers begin to increase speed. When your score reaches 50, the attackers from the top and bottom increase their speed. If your score exceeds 60, you win bonus smart bombs.

If your point total is a high score since the

program was first loaded, you enter your initials with the joystick. Moving the stick right or left steps through the alphabet forward or backward. When the correct letter is found, select it with the fire button. Be sure not to hold the fire button down too long when selecting your initials, or you may inadvertently choose the wrong letters.

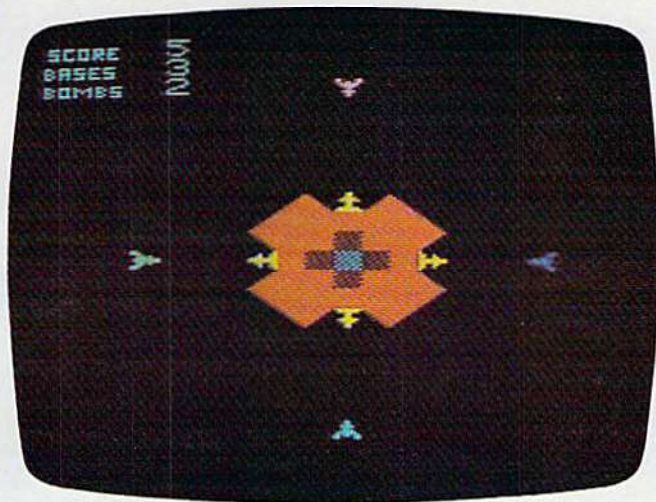
The VIC-20 version of Cylon Zap is in three parts. The first part redefines the start of BASIC to allow room for the redefined character set. It also loads the second part, which puts the special characters into memory and prints instructions. The third part contains the game itself. Each part must be typed in separately and then saved to tape. When saved in sequence, the game will load smoothly. If you use a disk drive, you must load the second and third programs manually when the prompt "Press Play On Tape" appears on the screen.

The Commodore 64 version of Cylon Zap consists of only one section because of the 64's larger memory. Otherwise, the game is essentially the same.

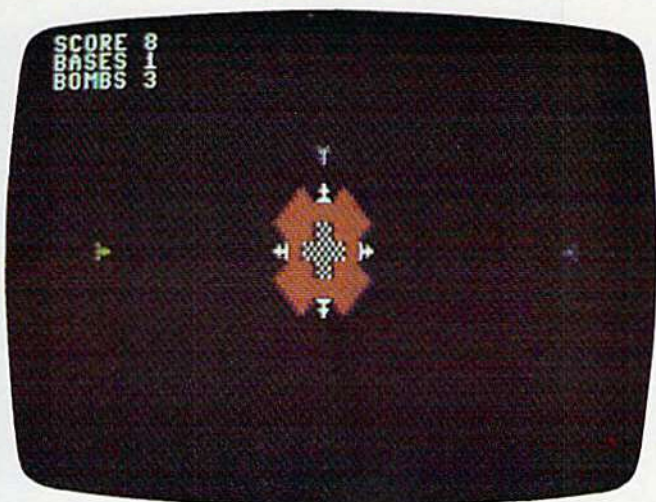
### Breakdown Of The VIC Version

Here are brief explanations of what's going on in the VIC version of Cylon Zap. This information is for programmers who are interested in studying the techniques used.





Attacking Cylon ships surround the player's space station in "Cylon Zap," VIC-20 version.



An enemy Cylon nears its target in "Cylon Zap," Commodore 64 version.

### Program 1: Set-up (VIC-20)

- Sets the start of BASIC pointer.
- Sets both end-of-memory pointers.
- Sets beginning of BASIC to 0.
- Prints "bytes free" message.
- Puts LOAD command into keyboard buffer.

### Program 2: Special Instructions (VIC-20)

- 30 Clear screen, set screen color, lowercase.
- 35 Display title.
- 40-70 Load characters.
- 75 Load machine language routine, ask for instructions.
- 90 POKE LOAD command into keyboard, clear screen.
- 95-125 DATA statements for characters.
- 130-160 Instructions.
- 165 RETURN to continue message.
- 170-180 Instructions, return to LOAD command.
- 190-230 Subroutine for "Hit RETURN To Cont" message.
- 235-290 Opening title.
- 300-310 Routine to move title across screen.
- 400-513 Routine to enter machine language.

### Program 3: Cylon Zap (VIC-20)

- 35-50 Initialize variables.
- 55 Set screen color, clear screen, display high scores.
- 60 Set beginning of play variables.
- 65 Set character pointer.
- 70-80 Clear screen, draw base.
- 90 Print score, bases, bombs in upper-left corner.
- 150-225 Enemy ship appear, move, hit.
- 230 No bases left, display scores.
- 233 Score 50 or more, ships appear faster.
- 235 GOTO beginning of joystick routine.
- 240-315 Draw base.
- 320-340 Fire laser up.
- 345-365 Fire laser down.
- 370-390 Fire laser left.
- 395-415 Fire laser right.
- 420-465 Music for entering initials.
- 470-475 DATA for song.
- 480-525 Explosion for hit on enemy ship, add score.
- 530-580 Bonus base and bomb routine.

- 585-615 Base explodes.
- 620-650 If score is one of high scores, rearrange other scores.
- 655-685 Print high scores and to play again.
- 690-725 Base explode graphics.
- 730-815 Enter initials routine.
- 820-850 Print scores and title.
- 855-895 Smart bomb explosion and scoring.
- 900 Ship alarm sound.
- 905-925 Enter initials print routine.

See program listings on page 119. ●

## SUPER DISK

### Floppy Disk Drive For VIC-20 & Commodore 64

Super Disk<sup>2</sup> is a Commodore compatible disk drive designed to interface to the various Commodore computers such as the PET<sup>1</sup>, VIC-20<sup>1</sup> and the Commodore 64<sup>1</sup>. The disk drive is compatible to the model 4040, 2031, 1540, and the 1541 disk drives and recognizes programs generated on any of these disk drives. The capacities are comparable to those found on the Commodore drives, and Super Disk<sup>2</sup> recognizes the full instruction set of the Commodore drives. Super Disk<sup>2</sup> offers RAM area within the disk unit, a serial and an IEEE bus interface.

### Introductory Offer...\$395.00

#### Also Available:

Gemini-10 w/Interface	\$399.	V3K RAM	25.
CPI Parallel Interface	65.	V8K RAM	45.
Expandoport 3 VIC	25.	V16K RAM	75.
Expandoport 6 VIC	75.	V24K RAM	105.
Expandoport 4 C64	65.	CIE (IEEE for C64)	95.

CATALOG OF OTHER HARDWARE & SOFTWARE AVAILABLE ON REQUEST.

We accept: VISA, Mastercharge, and AE

## Southwest Micro Systems, Inc

2554 Southwell • Dallas, Texas 75229

(214) 484-7836

<sup>1</sup>Trademark of Commodore Int. <sup>2</sup>Trademark of MSD



## VIC Music Composer

Gregg Peele,  
Programming Assistant

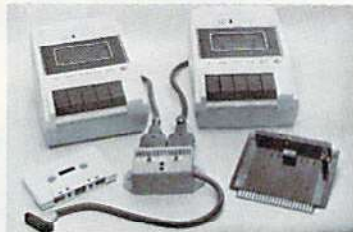
Musicians have coped with the problems of writing and re-writing music passages before Beethoven ever scribbled his scores on parchment. Many an eraser has been demolished while writing music, and many early versions of great masterpieces have ended up in the trash simply because there was no simple way to edit music.

With the advent of word processing, many writers no longer write text over and over again, discarding earlier versions for later improved versions. They simply edit their text to their liking on a screen and then use only the final version. Why not transfer what we have for text editing to the composition of music? The *VIC Music Composer* does just this - providing a dynamic method of composing, editing, and playing music passages. It provides a creative tool which makes simple music composition a much less frustrating task.

### Composing Options

When you first turn on the computer (with the *VIC Music Composer* cartridge in the expansion port), a menu appears on the screen, prompting you to select one of four modes of operation: Compose, Play, Save, or Load. If you select the Compose mode, you can choose among any of the three tone generators for composition. Next, you are asked whether you want to clear

## VIC-20\* OWNERS



The CB-2 is a complete hardware and software package that allows you to easily and efficiently make a back-up copy of your valuable software library. Now you can protect your investment!

### Unique features:

- Allows connection for one or two Datasette recorders (or equivalent). Two recorders required for simple back-up copies
- Exclusive state-of-the-art circuitry lets you actually hear and see tape data being loaded or saved
- Special wave shaping circuitry makes a back-up copy as good or better than the original
- CB-2's Super Block saver software and interface card allow you to make a back-up copy of your cartridge programs.

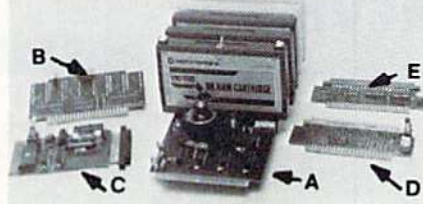
a) CB-2 Assembled	\$89.95
(Suggested Retail Price \$114.95)	
b) Special CB-2 System for Dataspan-50 owners (Please specify which model when ordering)	\$79.95
(Suggested Retail Price \$104.95)	
c) CB-2 Tape Unit For CBM 64*	\$69.95

### B. Champagne Memory on a Beer Budget

Highest quality glass epoxy 16K memory board with gold fingers provides full block switching and write-protection on each 8K block. All block switches are conveniently located at the top edge of the board.

<b>DataRAM</b>	
a) Bare memory board (RAM/RDM)	\$13.95
b) Bare memory board Kit	\$17.95
Includes all components except RAM/RDM chips	
<b>DataRAM 8</b>	
c) Board with 8K RAM - Complete Kit	\$34.95
d) Board with 8K RAM Assembled	\$37.95
(Suggested Retail Price Assembled \$47.95)	
<b>DataRAM 16</b>	
e) Board with 16K RAM - Complete Kit	\$48.95
f) Board with 16K RAM Assembled	\$54.95
(Suggested Retail Price Assembled \$69.95)	

\*Trademark Commodore Bus Machines



### A. Dataspan-50

The Dataspan-50 expansion board is the cornerstone for expanding the VIC-20 to its maximum capabilities. It is the most advanced, yet easiest to use expansion board available. Unlike other expansion boards, the Dataspan-50 has the following exclusive features:

- 5 slot combination rotary and rocker switch selectable expansion board conveniently covers all switching needs now and in the future without using common hazardous slot-by-slot power switching
- Combination switches allow total control between computer cartridges (memory expansion, Programmer's Aid\*, Vic-Mon\* and other utilities) and game cartridges
- Dataspan-50 allows stacking of memory cartridges up to 29K in BASIC and 40K in machine language
- Fully buffered by five hi-tech, low power, integrated circuits that help prevent erratic operation and loss of data common in typical unbuffered expansion boards and isolates the VIC's microprocessor from accidental damage
- Highest quality circuit board with gold contacts throughout
- Fused to protect the VIC-20's power supply
- Master reset button eliminates turning computer off and on
- Independent write-protection on two slots

Dataspan 50 Kit	\$69.95
Dataspan 50 Assembled	\$84.95
(Suggested Retail Price Assembled \$109.95)	

Dataspan-50 3-Slot Board Call

### C., D. & E.: See latest COMPUTE!

### What Makes The VIC Tic?

If you think computer hardware means nuts and bolts, this book is for you! Written especially for the beginner by VIC enthusiasts.

What Makes The VIC Tic?	\$ 7.95
(Shipping included)	

Digital Interface System Co. TERMS:  
P.O. Box 8715  
Portland, OR 97207  
(503) 295-5890  
No C.O.D. Orders  
Shipping and Handling \$3.00  
VISA/MASTERCARD - Add 3%  
Most orders shipped within 48 hours  
(Personal checks-allow 2 weeks)

# DEVELOP-20

## FIVE POWERFUL SOFTWARE DEVELOPMENT TOOLS

Plus The Exciting New Book

### INSIDE THE VIC

#### THE BOOK

A complete clear explanation of machine language, Assembly language, VIC 20 architecture, graphics, joystick and sound effect programming. Detailed step-by-step guide to the use of the development tools. How to combine BASIC and machine language, make auto-start cartridges, interface with the Internal ROM-based programs of BASIC and the Kernal. Sample programs fully explained.

#### THE TOOLS

Assembler/Editor/Loader/Decoder/Monitor Full-featured Assembler allows use of labels, comments and arithmetic expressions to create machine language programs. Create, save, modify Assembly language programs with the Editor. Load and link machine language modules with the Loader. Decode machine language back into assembly language for study or input to the Editor. Single-step program execution with the Monitor. Extended features combine Assembler/Editor for maximum ease of use.

**ALL FOR \$49.95 PLUS \$2.00 POSTAGE AND HANDLING** Standard version runs on any system with Datasette (5K and up) Add \$5.00 for disk version, \$5.00 for extended features (minimum 8K) Send check, M.O., VISA/MC (\$2.00 S.C.) or specify C.O.D. (add \$3.00) to:

*French Silk* smooth ware

P.O. Box 207, Cannon Falls, MN 55009

507-263-4821

VIC-20 is a registered TM of Commodore Business Machines Inc.



## REVIEWS

the memory of all the notes that you have previously composed. You are then asked to specify a key signature and the time signature for your composition.

After these preliminary choices are made, the screen displays the grand staff, including both treble and bass clefs. A large red cursor flashes at the bottom of the screen, and seven notes – each with a different time duration value – are displayed. By placing the cursor over the desired note value and pressing RETURN, you can “lock” the note value directly over an arrow which defines its horizontal position on the screen. The note is then moved up or down the grand staff by using the up/down cursor key. To select the final position of the note, press RETURN once more.

Once the note is fixed in its position, the screen returns to its original display, plus the note that you have just added. In the same way, you can add rests, bar lines, and dotted notes. The delete key removes any note on the screen. Notes can be inserted merely by moving the arrow to the appropriate place and entering your choice. The number keys (1-6) choose the volume (one is softest; six is loudest).

### Instant Playback

Since music on paper (or on the screen) gives the composer only a rough idea of how the music actually sounds, *VIC Music Composer* includes a Play mode so you can hear the composition performed. Since the program can display only one voice at a time (although it can play all three at a time), the Play mode



Composing music on the screen with VIC Music Composer.

asks you which voice should be displayed during playback. Then you select one of nine different tempos, and the music begins to play.

Other options in *VIC Music Composer* allow you to save and retrieve music on tape or disk. The program uses all the normal VIC prompts for loading and saving. Once the cartridge is in place, however, there is no way

to call the disk directory. If by accident you load a BASIC program instead of a music file, the system may act unpredictably.

The *VIC Music Composer* transforms the VIC into an enjoyable instrument for simple music composition. With this tool, music students can create compositions with up to three separate voices and achieve the immediate gratification of hearing their music performed directly after composition.

VIC Music Composer  
Thorn EMI Video  
1370 Avenue of the Americas  
New York, NY 10019  
\$39.95

## VOICE WORLD'S 24K Golden RAM Expansion Chassis

ONLY \$149.00

- Programmer's dream—Game player's delight.
- Boosts VIC memory to 29K!
- 4 expansion slots with switches for in-stant cartridge selection—faster than a disk.
- Accepts any cartridge designed for the VIC 20®.
- System Reset Button.
- Plugs directly into your VIC 20®.
- 8 memory control switches—easy to configure in 8K banks for custom applications.
- Factory tested—one year limited warranty.

- Start address selection at 2000, 4000, 6000, A000 HEX.
- ROM mode switches for memory write protection and PROM, EPROM emulation.
- Memory banks hold programs/data even when de-selected.
- Gold-plated connectors/switch contacts for high reliability.
- Fused to protect your VIC 20®.



**VOICE WORLD**  
13055 Via Esperia  
Del Mar, CA 92014  
(619) 481-7390

### TO ORDER:

Send check or money order. Add 3.00 shipping and handling. California residents add 6% sales tax. COD DEALER INQUIRIES INVITED

VIC 20 is a registered trademark of Commodore Business Machines, INC.



# VIC/64 Mailing List

Joseph J. Shaughnessy

**With a few simple changes you can make yourself, this program will work on either the VIC-20 or Commodore 64, with either disk or tape. Expansion memory is not required on the VIC, but will greatly increase the capacity of the mailing list.**

---

The following program is a modified and expanded version of a utility program from the Toronto PET Users Group, called "Addresses" and originally written in Dutch by Andy Finkel. The program has been translated into English and a printer option added. Using a Commodore printer, it can print the entire list or individual mailing labels.

The accompanying program listing is for the VIC-20 and 1540/1541 disk drive, but it can easily be modified to operate with the cassette recorder by changing the SAVE statement in line 55 to read SAVER\$. This program also works on the Commodore 64 by inserting extra spaces in the PRINT statements in lines 19 and 60 (to center the title display), changing the DATA statement in line 16 to "64 Mailing List", and by either deleting the POKE statement in line 16 or replacing it with the appropriate POKE statements to set the 64's screen color. (Screen color is green with black letters, but this is your choice.)

Each address field is set up to receive eight items of information, as shown in lines 17 and 18. These items can be changed to anything you want (for instance, to set up a filing system instead of a mailing list), but you are limited to eight items because of the size of the keyboard buffer (line 28). Also, since the DATA statements are printed on the screen as part of the procedure for adding them to the program, you must be careful not to make your items of information so wordy that printing eight DATA statements would cause the

first few lines to scroll off the screen and thereby be lost.

At one point, I had a version of this program that used upper- and lowercase letters, but I found this too inconvenient when using the "search" function. I often forgot to use appropriate capital letters either when entering the original information or when inputting the search value.

To aid in searching, names are entered and stored last name first, but they are sent to the printer first name first. Do not use commas when entering your mail list items.

This program will fit into any memory size VIC-20, but memory expansion is necessary to store very many addresses. (For instance, I have 65 names stored, and it takes about 12K of memory.) If you need space for more names (and have the memory) just add more "dummy" DATA statements to the end of the program.


The program prints mailing labels in a single column. Further work could be done to print the labels two or three across the width of the paper, and the formatting could be changed to match the layout of adhesive labels.

I addressed my Christmas cards with this program (tape version) and found it a big time-saver, even though I had to use scissors and tape to put the labels on the envelopes.

A disk drive or printer will certainly enhance the program's usefulness, but they are not essential. The program can probably be modified to run on other computers, depending on the size of the keyboard buffer.

If you don't want to type this program, please send a blank cassette or disk and \$3 with a stamped, self-addressed envelope to:

*Joe Shaughnessy  
4703 Country Club Drive  
Pittsburgh, PA 15236*

*See program listing on page 111. *



# VICreations

DAN CARMICHAEL, ASSISTANT EDITOR

**This month we'll talk about the proper care, handling, and maintenance of disk drives, diskettes, cassette recorders, and tape cassettes.**

Everyone who has worked with computers for some time knows the empty feeling of having lost a valuable program or data file. It might have been stored on a cassette tape that was creased when it became entangled inside the cassette recorder. The program is lost forever, along with the time it took to create it.

While some may take it for granted, data storage and retrieval may be one of the more important aspects of computing. After all, think of how inconvenient it would be for the computerist if there were no storage media. To run your favorite program or game, you would have to retype it every time you sat down at your keyboard — that or leave your computer on 24 hours a day. Or consider how large and expensive a computer would have to be if you couldn't retrieve data from an external device such as a disk drive, and it all had to be stored inside the computer.

Because input and output storage devices are so crucial to computing, the proper handling and care of these devices is especially important.

## Cassette Decks And Tapes

Although quite a bit slower than disk drives, cassette decks are dependable storage devices. The obvious advantage of cassettes is cost — about \$65 for the new Commodore C2N cassette deck compared to about \$350 for the Commodore VIC-1541 disk drive. Also, blank cassette tapes are relatively inexpensive, small, and easy to store in their protective plastic cases.

When using cassette decks, it is important that you follow the manufacturer's instructions. For instance, Commodore recommends that you place the cassette deck at least two feet away from your TV set or monitor. Never place the cassette next to or on top of the TV. This is because most TVs, radios, and monitors emit magnetic fields and radio frequency signals (called RF). These fields and signals interfere with the normal operation of the cassette deck and might cause errors while saving or loading programs with the VIC.

Another important point to watch is temperature. As is true with all computer equipment, do not expose the cassette deck to extreme hot or cold temperatures or moisture. These can be most damaging.

Cleaning the cassette deck heads, capstan pinch roller (the little black rubber roller inside the cassette deck), and tape guide is also important. Commodore recommends that you clean the deck after every 10 to 20 hours of use. You should refer to the instruction manual for complete instructions, but these additional tips may help. Never use alcohol or any solvent that might hurt or damage the plastic or rubber parts in the deck. It is recommended that you use a regular tape cassette head cleaner. These cleaners are widely available at audio stores. When applying the cleaner, do not use the type of cotton swab that is so soft it will leave particles of cotton on the tape heads. Even a tiny speck of cotton on the head can cause tape errors. After cleaning, you should always wait a few minutes before inserting a tape cassette into the deck to be sure that any remaining cleaner has dried. The head cleaning solvent could damage the magnetic tape.

The type of magnetic tape you use is also very important. As is true with audio applications, the better-quality magnetic tapes will give you better performance. For most purposes, audio quality tapes found at your local stereo store will suffice for data recording purposes. Commodore recommends that you *do not* use digital-quality recording tapes. Digital-quality tapes are often advertised as being specifically for use with computers, and are sometimes referred to as certified tapes. Also avoid magnetic tape with other than normal bias, such as high bias chromium dioxide tape.

The latest Commodore manuals suggest using cassettes that are 12 to 30 minutes in length, warning that longer tapes can strain the recorder. Commodore also recommends rewinding the tape back to the leader before placing it in its protective plastic case. Leaving the magnetic tape exposed could subject it to damage. Remember to always store tapes in their plastic cases.

## Disk Drives And Diskettes

Floppy disk drives are the most efficient medium



for data storage and retrieval with home computers. Disk drives such as Commodore's VIC-1540/1541 provide reliability, speed, and ease of operation. But the disk drives and diskettes are very delicate and demand the utmost in care and handling. Here are some hints and tips that should help you achieve many years of error-free performance.

Although there is very little the untrained person should do to the disk drive, there are a few minor things the home computerist can perform.

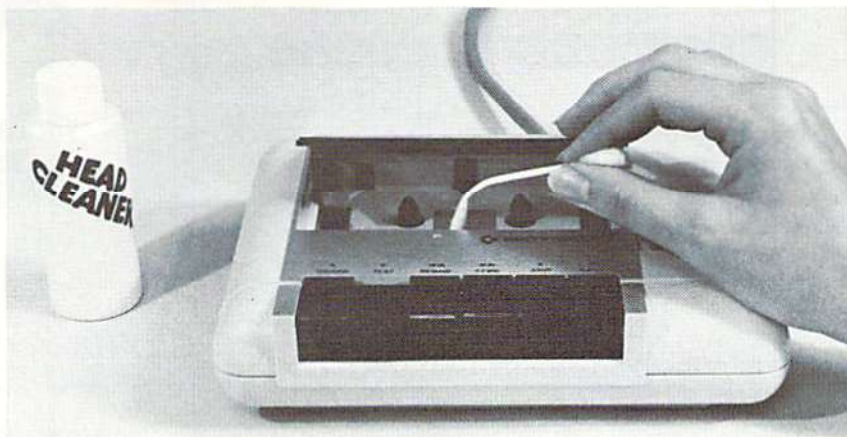
One is keeping the drive clean. Disk drive cleaning kits are available at your local computer shop. These kits contain instructions on the care of disk drives, cleaning fluid, and a special floppy diskette that is composed of a cloth-like material. To clean the disk drive, you place a little cleaning fluid on the cloth diskette, then insert it into the disk drive. As the diskette spins, it acts as a sort of spinning cotton swab, cleaning the magnetic heads in the drive and any other surface that comes into contact with the floppy disks. Although the instructions recommend that you use the cleaning kit once a week, for most home applications once a month should do.

Another bit of preventive maintenance you might perform at home is removing the dust from inside the disk drive. The easiest way to accomplish this is to carefully remove the cover from the disk drive and blow out the dust with compressed air. You can purchase canned compressed air at your local camera store. This air is specifically made for removing dust.

However, there is a word of caution to be observed here. If the canned air is handled wrong, it can spray moisture that could harm your disk drive. Read all directions carefully, and keep the can level at all times. Never use a vacuum cleaner or any other such device to remove dust. This maintenance should be performed about once every six months. Unlike most manuals, the 1541 manual does not warn that opening the disk drive cover will void the warranty. In fact, it recommends opening the drive for other purposes, such as changing the device address number for multiple-drive systems.

## Using And Storing Diskettes

Commodore 1540/1541 disk drives can store up to 144 directory entries and 174,848 bytes on each floppy diskette. Should one of your diskettes become damaged, it would take a lot of work to recreate this much information. This is why the handling and storage of floppy diskettes is so



*Proper care of your equipment is a must for reliable operation.*

important.

When handling floppy diskettes, observe the following tips. Always keep the diskette in its protective sleeve. Even if you remove the diskette from the drive for a brief second, return it to its protective sheath before laying it down. Never touch the shiny (magnetic coating) part of the diskette, because the oils on your fingers could contaminate it. Don't bend the diskette. If it should become even slightly bent, it will not spin properly in the disk drive. Always insert it and remove it from the disk drive carefully. Never force it.

Always store diskettes at least 12 inches away from the TV set or any other magnetic source. Always store diskettes in a vertical position; never stack them atop each other. Store them away from a direct source of heat, or any other temperature extremes. Most manufacturers of floppy diskettes recommend that you store them between 50 and 125 degrees Fahrenheit (10 to 52 degrees Centigrade).

If you observe these rules, your data storage devices should supply you with many years of enjoyable, productive performance. ☐

## Attention Writers

COMPUTE!'s Gazette is looking for well-written, clearly explained articles for beginning and intermediate users of VIC-20 and Commodore 64 personal computers. If you have an idea for a feature article or tutorial, submit a manuscript or send us a query letter. See the Author Guide elsewhere in this issue.

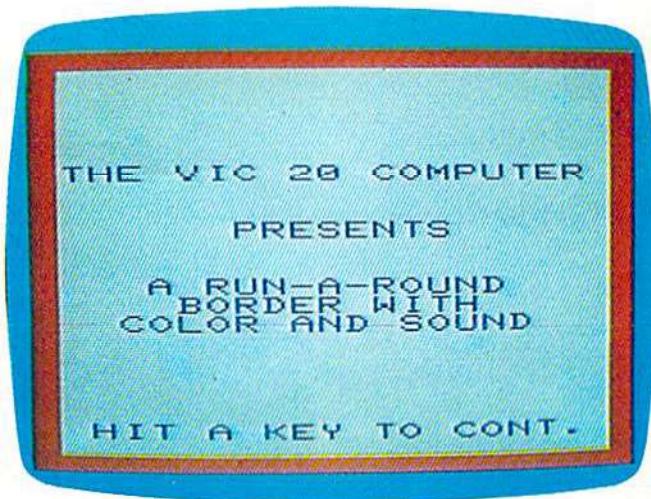


# HINTS & TIPS

## VIC Title Screens

Harry E. Watts

Drawing a run-around border with color and sound can add interest to any program title. As a teacher and novice in computer programming, I developed programs on a VIC-20 for use in my classes. After critiquing my work, it was apparent that the programs needed a nice-looking title screen. The demonstration program shows how a border can enhance your title and give a professional touch to any program. The border sub-



The example program lets you create title screens such as this.

routine can also be used for directions, ending programs, or wherever emphasis is needed.

Line 10 is the start of the program, which branches to the border subroutine at line 1000, which clears the screen and sets the border color. The color is defined by the variable C. Remember that the character color is one number less than the keyboard color keys.

Line 1010 sets the volume for the speaker and starts the loop for the top border. Line 1020 POKES in the screen characters. The demonstration program uses a reverse space (160) to achieve a solid border. A border can be customized to the content of the program by using any screen display code. Many interesting effects can be achieved. Line 1030 POKES in the screen character color,

and line 1040 adds the sound. The program plays low to high notes on a musical scale. Line 1050 creates a short duration period for each musical note and completes the main loop for the top border. Lines 1060 to 1200 generate the right, bottom, and left borders in the same manner. Line 1210 turns off the speaker and sets the volume to 0. Line 1220 RETURNS to the main program.

At this point the title will pop on the screen, giving a dramatic effect. Notice the proper spacing of the title. Remember that the screen display is 20 characters wide by 21 lines with the border. A GET statement in some form should be used to prevent the cursor from crashing the border. The demonstration program uses a simple "Hit a key to cont."

Random colors can also be used in the border subroutine by substituting the following for C in line 1000:

```
DEF FNA(C)=INT(RND(1)*8)
```

Then change C in lines 1030, 1080, 1130, and 1180 to read FNA(C).

```
10 GOSUB 1000
20 PRINT"{5 DOWN}{RIGHT}THE VIC 20 COMPU
  TER"
30 PRINT"{2 DOWN}{6 RIGHT} PRESENTS"
40 PRINT"{2 DOWN}{3 RIGHT} A RUN-A-ROUND
  "
50 PRINT"{5 RIGHT}BORDER WITH"
60 PRINT"{3 RIGHT}COLOR AND SOUND"
70? PRINT"{5 DOWN}{RIGHT} HIT A KEY TO CO
  NT."
80 GET A$:IF A$=""THEN 80
100 END
1000 PRINT"{CLR}":C=2
1010 POKE 36878,15:FOR L=0 TO 21
1020 POKE 7680+L,160
1030 POKE 38400+L,C
1040 POKE 36876,155+L
1050 FOR J=1TO25:NEXTJ,L
1060 FOR L=0TO440 STEP 22
1070 POKE 7723+L,160
1080 POKE 38443+L,C
1090 POKE 36876,176+(L/22)
1100 FOR J=1TO25:NEXTJ,L
1110 FOR L=21 TO 0 STEP-1{2 SPACES}
1120 POKE 8164+L,160{2 SPACES}
1130 POKE 38884+L,C
1140 POKE 36876,219-L
1150 FORJ=1TO25:NEXT J,L
1160 FORL=440TO0 STEP-22
1170 POKE 7702+L,160
1180 POKE 38422+L,C
1190 POKE 36876,241-(L/22)
1200 FORJ=1TO25:NEXTJ,L
1210 POKE 36876,0:POKE 36878,0
1220 RETURN
```



# THE BEGINNER'S CORNER

C. REGENA

## Playing Computer Music

Because of this month's topic – creating sound – The Beginner's Corner departs from its usual custom of covering both the VIC-20 and Commodore 64. Since sound is handled very differently on the 64 than on the VIC, this month's column deals only with the VIC.

Using your computer to play music can be a lot of fun. You may run a program to have your computer play a tune, then you can sing along with the computer. If you play a solo instrument, program the computer to play your accompaniment. If you are learning to play a difficult piece of music, have the computer play the piece so you can tell what it is supposed to sound like. Game programs can be enhanced with musical interludes. Educational programs can use music to reward correct responses. Music teachers and students can use music programs to improve the learning process. Any type of repetitious drill work can be done with the aid of the computer.

To program music on your computer, you need to consider volume (loudness), note (pitch), and duration (length of the note). The VIC-20 allows three *channels* or *voices* for music which may be played simultaneously, plus one voice for noises.

Different computers have different commands in BASIC to play music. Other microcomputers may use words such as SOUND and PLAY. In VIC programming the POKE command is used. POKE can be confusing because the form of the POKE statement is POKE *n1*,*n2*, where *n1* and *n2* are numbers and you need to know what type of numbers to use. *N1* is a *memory location* or *memory address* – a number from 0 to 65535 on the VIC. *N2* is a number that you put in the memory location. On the VIC this must be a number from 0 to 255. Depending on the numbers you use, the computer will act. Some POKE statements refer to graphics; others may change a type of computer operation such as speed or disabling certain keys.

### Volume

To turn on the volume to play music, the location number is 36878, and the volume setting can be from 0 (off) to 15 (loudest). The volume is ordinarily turned off, so if you happen to forget this statement in your program you won't hear any music. POKE 36878,15 sets the volume to the highest level.

You may wish to let the numbers be variables:

```
100 V=36878
110 L=15
120 POKE V,L
.
.
.
300 L=10
310 POKE V,L
.
.
.
500 POKE V,0
```

### Note Or Pitch

Even if the volume is turned on, you will not hear anything until you tell the computer to play a note. There are three locations that may be used for the sound channels: 36874, 36875, and 36876. Each voice has a different range of tones available. 36874 has the highest notes. The numbers you can POKE into these memory locations for sound purposes may range from 128 to 255, where 255 is the highest note in each range. There are charts in the book that comes with your computer and in the *VIC-20 Programmer's Reference Guide* that translate a note by letter name to the number necessary for computer language.

Let's try a few notes. Here is a short program to illustrate the commands necessary to play a note:

```
100 POKE 36878,10
110 POKE 36874,183
120 END
```



RUN the program and you will hear the note A. Notice that even though the program ends, you still hear the note. You need to POKE 36874,0 to end the sound. You may also turn off the volume by the command POKE 36878,0, or by holding down the RUN/STOP key while pressing RESTORE.

## Duration

The computer keeps playing a note until you change the note or tell the computer to stop. Your program may do calculations or draw pictures while the computer is playing notes (that's how sound effects are synchronized with graphics in games). There are several ways to make the computer play a note for a certain length of time and then change to a different note or stop.

One method of delay is to use a FOR-NEXT loop. A loop looks like this:

```
20 FOR D=1 TO 10
25 PRINT D
30 NEXT D
```

The FOR statement tells the computer to perform the loop the first time with D=1, the next time with D=2, the next time with D=3, and so on until D is greater than the limit 10 (the last loop will be with D=10). FOR-NEXT loops are a way of getting the computer to do something many times with only a few lines of instructions. Now, if you delete line 25 above, you will have a FOR-NEXT loop with no commands between, so the computer is just counting from 1 to 10. If you change the 10 to 100, you will notice the loop takes longer to perform. You can get different lengths of time by adjusting the loop limit number. Here is how loops can control the duration of sounds:

```
10 POKE 36878,12
20 POKE 36875,207
30 FOR D=1 TO 300:NEXT D
40 POKE 36875,201
50 FOR D=1 TO 300:NEXT D
60 POKE 36875,195
70 FOR D=1 TO 600:NEXT D
80 POKE 36875,0
90 POKE 36878,0
100 END
```

Keep in mind that with programming there are many ways to accomplish the same thing. Let's suppose we want to continue the tune in the program above and that most of the notes will use the delay loop of 300. We can use a GOSUB to save some typing and memory. GOSUB tells the computer to GO to a *SUBroutine* (a small program within your main program) to perform some lines, then RETURN to the statement following the GOSUB. Anytime you do a procedure a number of times, it may be better to make it a subroutine and use GOSUB. Here is an example:

```
10 POKE 36878,12
20 POKE 36875,207
30 GOSUB 200
40 POKE 36875,201
50 GOSUB 200
60 POKE 36875,207
70 GOSUB 200
80 POKE 36875,215
90 GOSUB 200
.
.
.
180 POKE 36875,0
190 POKE 36878,0
199 END
200 FOR D=1 TO 300
210 NEXT D
220 RETURN
```

To save typing numbers (and to save memory), you could let S=36875. Then every time you need 36875, just type S. In Commodore BASIC you may also leave off the variable in the NEXT statement, and you may combine lines by separating commands with a colon. You may also *tokenize* or abbreviate POKE by typing P, then SHIFT O. You may also leave out spaces to save memory.

The following program illustrates a way to use subroutines and FOR-NEXT loops for various note durations. I let GOSUB 56 represent the length I need for an eighth note. GOSUB 54 is twice as long and would represent a quarter note; GOSUB 52 is a dotted quarter note (equal to three eighth notes in time); and GOSUB 50 is used for a half note.

```
1 REM MELODY
2 POKE 36878,15
3 S=36876
4 T=200
5 POKE S,195:GOSUB 52
7 POKE S,187:GOSUB 56
9 POKE S,183:GOSUB 54
11 POKE S,175:GOSUB 54
13 POKE S,163:GOSUB 54
15 POKE S,175:GOSUB 54
17 POKE S,183:GOSUB 54
19 POKE S,163:GOSUB 54
21 POKE S,175:GOSUB 56
23 POKE S,183:GOSUB 56
25 POKE S,187:GOSUB 56
27 POKE S,175:GOSUB 56
29 POKE S,183:GOSUB 52
31 POKE S,175:GOSUB 56
33 POKE S,163:GOSUB 54
35 POKE S,159:GOSUB 54
37 POKE S,163:GOSUB 50
39 POKE S,0:POKE 36878,0
41 END
50 FOR D=1 TO T:NEXT
```



```

52 FOR D=1 TO T:NEXT
54 FOR D=1 TO T:NEXT
56 FOR D=1 TO T:NEXT:RETURN
60 END

```

Notice that line 4 sets T=200. If you wish to change the tempo of this tune, all you need to do is change this one line. For example, to make the melody play more slowly, change line 4 to T=350 and RUN the program. Now change line 4 to T=100 and RUN. All the note durations stay in the right proportion, but the speed is changed.

Following is a program for the same melody, but using DATA statements to READ in the note values and the duration factors. Line 4 still sets up a variable T so you can change the tempo. (I will explain using DATA and READ statements in more detail in a future column.)

```

1 REM MELODY2
2 POKE 36878,15
3 S=36876
4 T=200
5 READ N,A
7 IF N=-1 THEN 39
9 POKE S,N
11 FOR D=1 TO T*A:NEXT
13 GOTO 5
15 DATA 195,3,187,1,183,2,175,2,16
   3,2,175,2,183,2,163,2,175,1
17 DATA 183,1,187,1,175,1,183,3,17
   5,1,163,2,159,2,163,4,-1,-1
39 POKE S,0:POKE 36878,0
41 END

```

The best way to learn about the music capabilities of your computer is to write your own program. Experiment with the volume to accent certain notes. Try different durations to develop complex rhythms. Try different combinations of numbers with the three different speakers to hear the ranges of tones – and try using more than one voice at a time to harmonize.

## Let's Learn Notes: The Keyboard

This program, "Keyboard," is designed for the beginning piano or organ student who is learning the letter names of the notes on the keyboard. A music teacher may use this program before or after a student's regularly scheduled lesson time.

The program introduces the letter names of the keys on a piano or organ keyboard. After the instruction screens, a quiz of ten keys is presented. One of two keyboards is chosen randomly, either keys starting with a group of two black keys or keys starting with a group of three black keys. A red asterisk appears randomly on one of the keys. The student must press the letter name of the key shown. If the response is correct, the tone will be

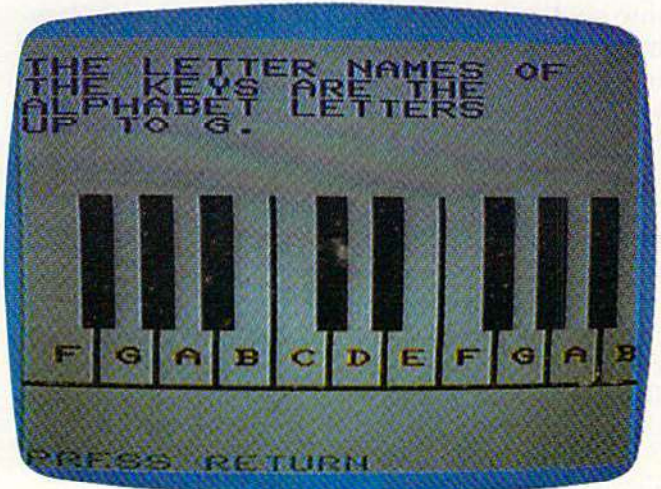
## Program Explanation

Lines	Description
2	Set volume level to loudest; branch past subroutines.
3-16	Subroutines for the notes. AA\$ is the letter name of the note. S is the tone number. CC is the column coordinate of the key on the keyboard. K may be 1 or 2 for the two possible keyboards.
20-23	Subroutines for graphics for drawing keyboards. RVS ON and RVS OFF are used to get the black and white keys.
50-51	Print title. SS is the speaker value 36876.
52-54	Using numbers from DATA, play a scale then branch to instructions.
55	Perform the quiz for ten keys.
56	Randomly choose, then draw one of two keyboards.
58-62	Randomly choose a note N. Depending on the keyboard and the note chosen, performs GOSUB to determine note name, tone, and coordinate.
64-66	Blink a red asterisk on the key chosen.
68	Wait for student to press a key.
70-72	If note is incorrect, sound a noise and return for another answer.
80	If note is correct, print name of note on key, play tone, go to next key.
90-94	Print option to try quiz again and branch appropriately.
104-106	Subroutines to draw graphics for keys.
108-111	Subroutine to print "PRESS RETURN" on an instruction screen and wait for student to respond.
150-152	Print first instruction screen.
154-160	Print second instruction screen.
164-168	Print third instruction screen; branch to quiz.
170	End.

played for that key. If the response is incorrect, there is a short noise and the student must try again.

This program uses graphics to help the instruction. The music capabilities of the computer allow the tones to be played so the student can hear the note as well as see the physical placement.

See program listing on page 125.



One of the sample programs shows how to translate the notes of a piano keyboard to make music with a VIC.



# 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, S 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 N}), 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, {k}, 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}	CTRL 7	
{HOME}	CLR/HOME		{PUR}	CTRL 5		{8}	CTRL 8	
{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}	CTRL 1		{F4}	F4	
{RVS}	CTRL 9		{2}	CTRL 2		{F5}	F5	
{OFF}	CTRL 0		{3}	CTRL 3		{F6}	F6	
{BLK}	CTRL 1		{4}	CTRL 4		{F7}	F7	
{WHT}	CTRL 2		{5}	CTRL 5		{F8}	F8	
{RED}	CTRL 3		{6}	CTRL 6				



# Hi-Res Graphics

## BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs" and "A Beginner's Guide To Typing In Programs" that appear before the Program Listings.

### Program 1: New BASIC

```
0 REM BASIC HI-RES
10 A=0:REM INITIALIZE CHECKSUM
20 REM MOVE BASIC ROM TO RAM
30 FORI=40960TO49151:POKEI,PEEK(I):NEXTI
40 REM CHANGE LET TO HUE
50 FORI=41150TO41152:READN:POKEI,N:A=A+N:NEXTI
60 READL,H:POKE40988,L:POKE40989,H:A=A+L+H
70 DATA 72, 85, 197, 75, 196
80 REM CHANGE WAIT TO PLOT
90 FOR I=41189TO41192:READN:POKEI,N:A=A+N:NEXTI
100 READL,H:POKE41008,L:POKE41009,H:A=A+L+H
110 DATA 80, 76, 79, 212, 130, 196
120 REM CHANGE CONT TO WIPE
130 FORI=41225TO41228:READN:POKEI,N:A=A+N:NEXTI
140 READL,H:POKE41024,L:POKE41025,H:A=A+L+H
150 DATA 87, 73, 80, 197, 53, 196
160 REM CHANGE VERIFY TO SCREEN
170 FORI=41201TO41206:READN:POKEI,N:A=A+N:NEXTI
180 READL,H:POKE41014,L:POKE41015,H:A=A+L+H
190 DATA 83,67,82,69,69,206,11,196
200 REM CHANGE ERROR MESSAGE ROUTINE
210 FORI=42042TO42044:READN:POKEI,N:A=A+N:NEXTI
220 DATA 76, 0, 196
230 REM READ IN NEW ROUTINES
240 FORI=50176TO50480:READN:POKEI,N:A=A+N:NEXTI
250 IFA<>39040THENPRINT"ERROR IN DATA STATEMENTS"
260 END
300 DATA 32, 24,196,138, 10,170, 76, 61, 164, 80, 70, 83, 32,158,183,224, 1
310 DATA144, 5,240, 19, 76, 72,178,169, 27,141, 17,208,169, 21,141, 24,208
320 DATA169,151,141, 0,221, 96,169, 59,1 41, 17,208,169, 8,141, 24,208,169
330 DATA148,208,238,162, 32,169,224,133, 252,160, 0,132,251,152,145,251,200
340 DATA208,251,230,252,202,208,246, 96, 32,123,196,138, 10, 10, 10, 10,133
350 DATA 2, 32,253,174, 32,123,196,138, 5, 2,160,192,132,252,160, 0,132
360 DATA251,162, 2,145,251,200,208,251,2 30,252,202, 16,246,145,251,200,192
370 DATA232,144,249, 96, 32,158,183,224, 16,176, 17, 96, 32,235,183,134, 2
380 DATA169,199, 56,229, 2,133, 2,201,20 0,144, 3, 76, 72,178,165, 21,240
390 DATA 10,201, 1,208,245,165, 20,201,
```

```
64,176,239,169, 0,133,251,169,224
400 DATA133,252,165, 20, 41,248, 24,101, 251,133,251,165, 21,101,252,133,252
410 DATA165, 2, 41, 7, 24,101,251,133,25 1,144, 2,230,252,165, 2, 74, 74
420 DATA 74, 10,170,189,247,196, 24,101, 251,133,251,189,248,196,101,252,133
430 DATA252,165, 20, 41, 7,170,160, 0,12 0,169, 52,133, 1,177,251, 29, 41
440 DATA197,145,251,169, 54,133, 1, 88, 96, 0, 0, 64, 1,128, 2,192, 3
450 DATA 0, 5, 64, 6,128, 7,192, 8, 0, 1 0, 64, 11,128, 12,192, 13, 0
460 DATA 15, 64, 16,128, 17,192, 18, 0, 20, 64, 21,128, 22,192, 23, 0, 25
470 DATA 64, 26,128, 27,192, 28, 0, 30,1 28, 64, 32, 16, 8, 4, 2, 1
```

### Program 2: A Simple Sine Wave

```
10 SCREEN 1: REM TURN ON BITMAP
20 WIPE: REM CLEAR BITMAP
30 HUE 0,1: REM BLACK DOTS, WHITE SCREEN
40 FOR X=0 TO 319 STEP .5
50 Y=INT(90+80*SIN(X/10))
60 PLOT X,Y: REM PLOT POINT
70 NEXT X
80 GET A$: IF A$="" THEN 80: REM WAIT FO R KEYSTROKE
90 SCREEN 0: REM NORMAL SCREEN
```

### Program 3:

#### A Joystick-Driven Doodle Pad

```
10 SCREEN 1: WIPE: HUE 0,1
20 X=159: Y=99: PLOT X,Y
30 GOSUB 100: IF J=15 THEN 30
40 PLOT X,Y: GOTO 30
50 SCREEN 0: END: REM GRACEFUL EXIT
100 REM READ JOYSTICK
110 J=PEEK(56320) AND 15: REM PORT 2
120 IF (J AND 8)=0 THEN X=X+1: REM MOVE RIGHT
130 IF (J AND 4)=0 THEN X=X-1: REM MOVE LEFT
140 IF (J AND 2)=0 THEN Y=Y-1: REM MOVE DOWN
150 IF (J AND 1)=0 THEN Y=Y+1: REM MOVE UP
160 IF Y<0 THEN Y=0: REM STAY IN RANGE
170 IF Y>199 THEN Y=199
180 IF X>319 THEN X=319
190 IF X<0 THEN X=0
200 GET A$:IF A$=CHR$(147) THEN WIPE: RE M CLEAR SCREEN
210 IF A$=CHR$(136) THEN 50: REM F7 KEY TO EXIT
220 RETURN
```

## VIC/64 Mailing List

```
8 REM VIC MAILING LIST PROGRAM-DISK VERS ION
10 REM{2 SPACES}MODIFIED & EXPANDED FROM A TPUG PUBLIC DOMAIN PROGRAM BY ANDY FINKEL
16 POKE36879,93:READR$,R:FORI=1TOR:READO $(I):NEXT:DATA"VIC MAILING LIST
17 DATA8,"NAME(LAST NAME FIRST)","STREET ADDRESS","CITY","STATE","ZIPCODE"
18 DATA"HOME PHONE NO.,""COMPANY NAME",""
```



```

WORK PHONE NO."
19 PRINT"{CLR}{BLK}{A}
*****
[S]":PRINT"-{2 SPACES}"R$" _":PRINT
"[Z]
*****
[X]
20 PRINT"{DOWN} 1. ADD NAME":PRINT"
{DOWN} 2. REMOVAL{4 SPACES}":PRINT"
{DOWN} 3. SEARCH":PRINT"{DOWN} 4. EXA
MINE
21 PRINT"{DOWN}{SHIFT-SPACE}5. CHANGE":P
RINT"{DOWN} 6. SAVE UPDATE":PRINT"
{DOWN} 7. PRINT OPTION":PRINT"{DOWN}
8. END"
22 RESTORE:PRINT"{2 DOWN}WHICH DO YOU WA
NT
23 GETA$: IFA$=""THEN23
24 IFA$<"1"ORA$>"8"THEN23
25 READB$: IFB$<"[-]"THEN25
26 A=VAL(A$):ONAGOTO29,34,37,47,54,55,60
,56
28 POKE198,10:FORI=0TO9:POKE631+I,13:NEX
T:END
29 READA$: IFA$<"[+]"THEN29
30 READA:PRINT"{CLR}INPUT 0 FOR UNKNOWN$
{DOWN}"
31 PRINT"ITEM : "A"{DOWN}":FORI=1TOR:PRI
NTO$(I):INPUTW$(I):PRINT:IFW$(1)="TH
EN19
32 NEXT:W$(0)="XX"+CHR$(34)+",""+STR$(A):
Z=A*10+500:K=0:PRINT"{CLR}{2 DOWN}"
33 FORI=ZTOZ+R:PRINTI;"DATA"CHR$(34)W$(K
):K=K+1:NEXT:PRINT"RUN{HOME}":GOTO28
34 B$="":PRINT"{CLR}WHICH ITEM TO REMOVE
":INPUTB$:IFVAL(B$)=0THEN19
35 PRINT"{CLR}{2 DOWN}":Z=VAL(B$)*10+500
:PRINTZ"DATA"CHR$(34)"[+]"CHR$(34)"
,"VAL(B$)
36 FORI=Z+1TOZ+R:PRINTI:NEXT:PRINT"RUN
{HOME}":GOTO28
37 INPUT"{CLR}SEARCH FOR ";B$:IFB$=""THE
N19
38 H=0:READA$: IFA$="END"THEN19
39 IFA$="[+]"THENREADA:GOTO38
40 READA:FORI=1TOR:READA$(I):IFLEFT$(A$(
I),LEN(B$))=B$THENH=1
41 NEXT:IFH=0THEN38
42 PRINT"{CLR}ITEM : "A"{2 DOWN}":FORI=1T
OR:PRINT"{2 SPACES}"A$(I):NEXT:IFW=1T
HENRETURN
43 PRINT"{2 DOWN}HIT ANY KEY TO PROCEED"
44 GETA$: IFA$=""THEN44
45 IFQ=1THENRETURN
46 GOTO38
47 A$="":INPUT"{CLR}WHICH ITEM";A$:IFA$=
"."THEN19
48 A=VAL(A$):IFA=0THEN19
49 READA$: IFA$="END"THEN19
50 IFA<>VAL(A$)THEN49
51 READA$(1):IFA$(1)="[+]"THEN19
52 FORI=2TOR:READA$(I):NEXT:Q=1:GOSUB42:
Q=0:IFW=1THENRETURN
53 GOTOL9
54 W=1:GOSUB47:W=0:PRINT"{HOME}{2 DOWN}"
:FORI=1TOR:INPUTW$(I):GOTO32
55 PRINT"{CLR}":SAVE"@0:"+R$,8:END
56 END
60 PRINT"{CLR}{2 DOWN}{RVS}{3 SPACES}PRI
NTER OPTIONS{3 SPACES}"
62 PRINT"{DOWN}{RVS}1{OFF} ENTIRE LIST"
64 PRINT"{DOWN}{RVS}2{OFF} MAILING LABEL
S"
66 PRINT"{DOWN}{RVS}3{OFF} INDIVIDUAL DA
TA"
68 PRINT"{DOWN}{RVS}4{OFF} SINGLE MAILIN
G LABEL";
70 PRINT"{DOWN}{RVS}5{OFF} RETURN TO PRO
GRM"
72 GETZ$:IFZ$=""THEN72
73 Z=VAL(Z$)
74 IFZ<1ORZ>5THEN60
76 OPEN1,4:RESTORE
78 ONZGOTO82,98,106,106
80 GOTOL9
82 READB$:IFB$<"[-]"THEN82
84 READB$:IFB$="[+]"THENCLOSE1:GOTOL9
86 IFB$="XX"THENREADA:PRINT#1,CHR$(10)CH
R$(10)"ITEM";A:GOSUB92:GOTO84
88 PRINT#1,B$
90 GOTOL9
92 READB$:FORI=1TO50
93 IFMID$(B$,I,1)=" "THENX=I:I=50
94 NEXTI
95 N2$=LEFT$(B$,X):N1$=RIGHT$(B$,LEN(B$)
-X):PRINT#1
96 PRINT#1,N1$;" ";N2$:RETURN
98 READB$:IFB$<"[-]"THEN98
99 READB$:IFB$="[+]"THENCLOSE1:GOTOL9
100 IFB$<"XX"THEN99
101 READA:PRINT#1,CHR$(10):GOSUB92:GOSUB
102:GOTO99
102 FORI=1TO4:READA$(I):NEXT
103 PRINT#1,A$(1):PRINT#1,A$(2);","";A$(
3);"{3 SPACES}";A$(4)
104 RETURN
106 INPUT"{CLR}{DOWN}WHICH ITEM";Q:RESTO
RE
107 READB$:IFB$<"[-]"THEN107
108 READB$:IFB$="[+]"THENPRINT"NO SUCH
ITEM ON FILE":FORX=0TO1500:NEXTX:CL
OSE1:GOTOL9
110 IFB$="XX"THENREADA:IFA=QTHEN114
112 GOTOL08
114 PRINT#1,CHR$(10)CHR$(10)"ITEM";A:GOS
UB92
116 IFZ=4THEN120
118 FORX=1TOR-1:READB$:PRINT#1,B$:NEXT:C
LOSE1:GOTO60
120 GOSUB102:CLOSE1:GOTO60
500 DATA"[-]"
510 DATA"XX", 1
511 DATA"SHAUGHNESSY JOE
512 DATA"4703 COUNTRY CLUB DR
513 DATA"PITTSBURGH
514 DATA"PENNSYLVANIA
515 DATA"15236
516 DATA"412-882-4655
517 DATA"DICK COAL CORP.
518 DATA"412-664-8280
520 DATA"[+]", 2
530 DATA"[+]", 3
540 DATA"[+]", 4
550 DATA"[+]", 5
560 DATA"[+]", 6
570 DATA"[+]", 7
580 DATA"[+]", 8
590 DATA"[+]", 9
600 DATA"[+]", 10
610 DATA"END"
112 COMPUTE!'s Gazette August 1983

```



# Global Scan

## BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs" and "A Beginner's Guide To Typing In Programs" that appear before the Program Listings.

```

5 REM GLOBAL SCAN
10 GOTO840
20 PRINT"
  *****
  "
30 PRINT"NAME OF LOCATION A:"
40 INPUTA$
50 PRINT"{CLR}ENTER THE LATITUDE OF"
60 PRINTA$;" : "
70 INPUTD,M,S
80 GOSUB650
90 IFQ$="Y"THENGOTO110
100 GOTO50
110 H3=D+(M/60)+(S/3600)
120 PRINT"{CLR}ENTER THE LONGITUDE:"
130 INPUTD,M,S
140 GOSUB650
150 IFQ$="Y"THENGOTO170
160 GOTO120
170 L1=D+(M/60)+(S/3600)
180 PRINT
190 PRINT"{CLR}NAME OF LOCATION B:"
200 INPUTB$
210 PRINT"{CLR}ENTER THE LATITUDE OF"
220 PRINTB$
230 INPUTD,M,S
240 GOSUB650
250 IFQ$="Y"THENGOTO270
260 GOTO210
270 H4=D+(M/60)+(S/3600)
280 PRINT"{CLR}ENTER THE LONGITUDE:"
290 INPUTD,M,S
300 GOSUB650
310 IFQ$="Y"THEN330
320 GOTO280
330 L2=D+(M/60)+(S/3600)
340 P=3.14159:R=P/180:W=180/P
350 H3=H3*R:L1=L1*R:H4=H4*R:L2=L2*R:GOSU
  B470
360 PRINT"{CLR}THE DISTANCE BETWEEN"
370 PRINTA$;" AND"
380 PRINTB$;" IS:"
390 PRINT
400 PRINTM;"MILES"
410 PRINTK;"KILOMETERS"
420 PRINTX1;"DEGREES/NORTH"
440 PRINT
450 PRINT
460 GOTO20
470 REMMATH
480 H1=.5*(H4-H3):H2=.5*(H4+H3):C=L1-L2
490 IFC=0ANDH1=0THENGOTO640
500 IFH1=0THENH1=1E-6
510 IFC=0THENC=1E-6
520 I1=ATN(1/(TAN(.5*C))*(SIN(H1))/(COS(
  H2)))
530 I2=ATN(1/(TAN(.5*C))*(COS(H1))/(SIN(
  H2)))
540 X=I2-I1:Y=I2+I1:Z=2*ATN((TAN(H1))*(S

```

```

IN(I2))/SIN(I1))
550 Z1=Z*W:K=111.12*Z1:M=69.048*Z1:X1=X*
  W:Y1=Y*W
560 IFL1<L2THENX1=360+X1
570 IFL1<L2THENY1=-1*Y1
580 IFL2<L1THENY1=360-Y1
590 K=INT(100*K+.5)/100
600 M=INT(100*M+.5)/100
610 X1=INT(100*X1+.5)/100
620 Y1=INT(100*Y1+.5)/100
630 RETURN
640 X=0:Y=0:Z=0:GOTO550
650 PRINT
660 PRINT"IS THIS CORRECT:"
670 PRINTD;"DEGREES"
680 PRINTM;"MINUTES"
690 PRINTS;"SECONDS"
700 PRINT
710 IFD=0THENGOTO800
720 IFM=0THENGOTO800
730 IFS=0THENGOTO800
740 PRINT"{RVS}Y{OFF}ES OR {RVS}N{OFF}O"
750 GETQ$
760 IFQ$=""THEN750
770 RETURN
780 PRINT"DO NOT ENTER A 0,"
790 PRINT"PLEASE RE-ENTER-"
800 PRINT
810 PRINT"PUSH {RVS}R{OFF} TO REDO-"
820 GOTO750
830 END
840 REM
850 CLR
860 PRINT"{CLR}{8 DOWN}{5 RIGHT}PROGRAM
  FOR"
870 PRINT"{2 RIGHT}DISTANCE AND ANGLE"
880 PRINT"{2 RIGHT}BETWEEN TWO POINTS"
890 FORT=1TO4000:NEXT
900 GOTO20

```

## Disk Menu For The VIC And 64

```

100 REM VIC/64 DISK MENU
110 REM POKE 53281,1 FOR COMMODORE 64
120 CLR
130 PRINT"{CLR}{4 DOWN}{GRN}READING DISK
  DIRECTORY{BLU}"
140 DIMP$(100)
150 REM DIRECTORY
160 REM{2 SPACES}PEEPER
170 OPEN1,8,0,"$"
180 REM LOAD ARRAY P$
190 GET#1,B$:IFST<>0THEN270
200 IFB$<>CHR$(34)THEN190
210 P$=""
220 GET#1,B$:IFB$<>CHR$(34)THENP$=P$+B$:
  GOTO220
230 GET#1,B$:IFB$=CHR$(32)THEN230
240 P$(X)=P$:PRINTX;"{UP}"
250 GET#1,B$:IFB$<>"{DOWN}"THEN250
260 IFST=0THENX=X+1:GOTO180
270 CLOSE1
280 REM{2 SPACES}DISPLAY MENU
290 N=10
300 PRINT"{CLR}{RED}VIC DISK MENU"
310 PRINT"FOR ";P$(0);"{BLU}":PRINT
320 IFN<10THENN=10

```



```

330 IFN>100THENN=100
340 FORJ=(N-9)TON:PRINTJ;"- ";P$(J):NEXT
  J
350 PRINT:PRINT"PRESS{DOWN}{5 LEFT}{RED}
  N{BLU} FOR NEXT SCREEN"
360 PRINT"{RED}L{BLU} FOR LAST SCREEN"
370 PRINT"{RED}E{BLU} TO EXIT TO BASIC"
380 PRINT"{PUR}S{BLU} TO LOAD ONLY"
390 PRINT"{GRN}R{BLU} TO LOAD AND RUN"
400 Z$="":GETZ$:IFZ$=""THEN400
410 IFZ$="N"THENN=N+10:GOTO300
420 IFZ$="L"THENN=N-10:GOTO300
430 IFZ$="E"THENPRINT"MENU STILL PRESENT
  ":END
440 IFZ$="S"THENGOSUB470:GOTO570
450 IFZ$="R"THENGOSUB470:GOTO540
460 GOTO400
470 REM SELECT AND{12 SPACES}PRINT LOAD
480 PRINT"{DOWN}PROGRAM #";
490 INPUTS:PRINT"{17 DOWN}"
500 IFS<LORS>XTHEN480
510 IFLEN(P$(S))>12THENP$(S)=LEFT$(P$(S)
  ,12)+"*"
520 PRINT"LOAD"+CHR$(34)+P$(S)+CHR$(34)+
  ",8{3 UP}"
530 RETURN
540 REM LOAD AND RUN
550 POKE631,13:POKE632,82:POKE633,85:POK
  E634,78:POKE635,13:POKE198,5
560 END
570 REM LOAD ONLY
580 POKE631,13:POKE198,1

```

## Wordspell

### BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs" and "A Beginner's Guide To Typing In Programs" that appear before the Program Listings.

### Program 1: Wordspell, VIC Version

```

0 PRINT"{CLR}":POKE36869,242:GOSUB61
21 PRINT"{CLR}{BLU}{4 DOWN}{3 SPACES}CRE
  ATE NEW LIST":INPUT"{2 DOWN}{4 RIGHT}
  {2 SPACES}(Y OR N)";R$:IFR$="Y"THEN50
22 IFR$<>"N"THEN21
23 V=36878:S=36876:SC=36879
24 DIMA$(19),W$(19)
25 FORP=0TO19:READA$(P):NEXT
26 FORP=0TO19
27 PRINT"{CLR}"
28 PRINTSPC(220)
29 PRINTTAB(INT(22-LEN(A$(P)))/2)
30 GOSUB35
31 GOSUB40
32 NEXT
33 GOTO71
34 REM PRINT OUT WORDS
35 FORX=1TOLEN(A$(P))
36 PRINTMID$(A$(P),X,1);
37 FORT=1TO300:NEXT
38 NEXT
39 RETURN
40 PRINT"{CLR}":PRINTSPC(220)

```

```

41 PRINTTAB((INT(22-LEN(A$(P)))/2)-2):IN
  PUTA$
42 IFA$=A$(P)THENPRINT"{CLR}":PRINTSPC(2
  26)"CORRECT 1":GOSUB82:GOTO46
43 W$(P)="W":GOSUB81
44 PRINT"{CLR}":PRINTSPC(118):PRINT"WRON
  G{SHIFT-SPACE}I":PRINT:PRINT" CORRECT
  {SHIFT-SPACE}SPELLING{SHIFT-SPACE}IS:
  ":PRINT
45 PRINT:PRINT:PRINT:PRINTTAB(INT(22-LEN
  (A$(P)))/2)A$(P):K=K+1
46 FORT=1TO2000:NEXT
47 POKESC,27
48 RETURN
49 REM CREATE WORD DATA
50 PRINT"{CLR}":DIMB$(19)
51 FORI=0TO19:PRINT"WORD";I+1;:INPUTB$(I
  ):NEXT
52 PRINT"{CLR}{WHT}{2 DOWN}"
53 FORI=0TO17STEP4
54 PRINTI+I;"DA"CHR$(34)B$(I)CHR$(34);",
  "CHR$(34)B$(I+1)CHR$(34);
55 PRINT", "CHR$(34)B$(I+2)CHR$(34);", "CH
  R$(34)B$(I+3):NEXT
56 PRINT"GOTO1":PRINT"{HOME}"
57 POKE198,6
58 FORI=0TO5:POKE631+I,13:NEXT
59 END
60 REM INSTRUCTIONS
61 PRINT" USE THIS PROGRAM FOR SPELLING
  PRACTICE.{4 SPACES}WHEN REQUESTED, EN
  TER THE SPELLING";
62 PRINT" WORDS AND{RVS}PRESS RETURN
  {OFF}.{2 SPACES}WHEN{3 SPACES}ALL (20
  ) OF THE WORDS HAVE BEEN ENTERED";
63 PRINT" THEYWILL BE PLACED INTO
  {3 SPACES}THE PROGRAM AS DATA
  {3 SPACES}STATEMENTS.{2 SPACES}RE-SAV
  E"
64 PRINT"-ING THE PROGRAM AT{3 SPACES}TH
  E END OF THE SESSIONWILL SAVE THE ENT
  ERED WORDS FOR";
65 PRINT" USE AT THE{2 SPACES}NEXT PRACT
  ICE."
66 PRINT"{4 DOWN}{5 RIGHT}{RVS}PRESS RET
  URN{OFF}"
67 GETR$:IFR$=""THEN67
68 IFR$=CHR$(13)THENRETURN
69 GOTO67
70 REM PRINT OUT MISSPELLED WORDS AND SC
  ORE
71 PRINT:PRINT"{CLR}{RVS}MISSPELLED WORD
  S:{OFF}":PRINT
72 FORP=0TO19:IFW$(P)="W"THENPRINTTAB(4)
  A$(P)
73 NEXT
74 PRINT"{HOME}{19 DOWN}{RVS}SCORE ="100
  -K*5
75 PRINT:PRINT"{3 SPACES}AGAIN ? (Y OR N
  )"
76 GETR$:IFR$=""THENGOTO76
77 IFR$="Y"THENRUN1
78 IFR$<>"N"THEN76
79 POKE36869,240:POKEV,0:POKES,0
80 GOTO59
81 POKESC,26:POKEV,15:FORX=180TO145STEP-
  1:POKES,X:NEXT:POKEV,0:RETURN
82 POKEV,15:FORX=220TO255:POKE36876,X:NE
  XT:POKEV,0:RETURN

```



## Program 2: Wordspell, 64 Version

```
0 PRINT "{CLR}":PRINTCHR$(14):POKE53280,7
:POKE53281,1:GOSUB61
21 PRINT "{CLR}{BLK}{4 DOWN}{3 RIGHT}
{3 SPACES}CREATE NEW LIST":INPUT"
{2 DOWN}{4 RIGHT}{2 SPACES}(Y OR N)";
R$:IFR$="Y"THEN50
22 IFR$<>"N"THEN21
24 DIMA$(19),W$(19)
25 FORP=0TO19:READA$(P):NEXT
26 FORP=0TO19
27 PRINT "{CLR}"
28 PRINT "{9 DOWN}"
29 PRINTTAB(INT(40-LEN(A$(P)))/2)
30 GOSUB35
31 GOSUB40
32 NEXT
33 GOTO71
34 REM PRINT OUT WORDS
35 FORX=1TOLEN(A$(P))
36 PRINTMID$(A$(P),X,1);
37 FORT=1TO300:NEXT
38 NEXT
39 RETURN
40 PRINT "{CLR}":PRINT "{9 DOWN}"
41 PRINTTAB((INT(40-LEN(A$(P)))/2)-2):IN
PUTA$
42 IFA$=A$(P)THENPRINT "{CLR}":PRINTSPC(2
15)"{5 DOWN}CORRECT 1":GOSUB90:GOTO46
43 W$(P)="W":GOSUB81
44 PRINT "{CLR}":PRINT "{4 DOWN}"SPC(17);"
WRONG 1":PRINT "{2 DOWN}"SPC(9)" CORRE
CT{SHIFT-SPACE}SPELLING{SHIFT-SPACE}I
S:"
45 PRINT:PRINT:PRINT:PRINTTAB(INT(40-LEN
(A$(P)))/2)A$(P):K=K+1
46 FORT=1TO2000:NEXT
47 POKE 53280,7
48 RETURN
49 REM CREATE WORD DATA
50 PRINT "{CLR}":DIMB$(19)
51 FORI=0TO19:PRINT"WORD";I+1;:INPUTB$(I
):NEXT
52 PRINT "{CLR}{2 DOWN}{WHT}"
53 FORI=0TO19STEP4
54 PRINTI+I;"DA"CHR$(34)B$(I)CHR$(34);",
"CHR$(34)B$(I+1)CHR$(34);
55 PRINT", "CHR$(34)B$(I+2)CHR$(34);", "CH
R$(34)B$(I+3):NEXT
56 PRINT"GOTO1":PRINT "{HOME}"
57 POKE198,10
58 FORI=0TO5:POKE631+I,13:NEXT
59 END
60 REM INSTRUCTIONS
61 PRINT "{BLK}{3 DOWN}{6 SPACES}USE THIS
PROGRAM FOR SPELLING":PRINT"PRACTIC
E.{2 SPACES}WHEN";
62 PRINT" REQUESTED, ENTER THE":PRINT" S
PELLING WORDS AND {RVS}PRESS RETURN
{OFF}."{2 SPACES}WHEN"
63 PRINT" ALL (20) OF THE WORDS HAVE BEE
N":PRINT" ENTERED, THEY WILL BE PLACE
D";
64 PRINT" INTO THE":PRINT" PROGRAM AS DA
TA STATEMENTS.{2 SPACES}RE-SAVE-"
65 PRINT" ING THE PROGRAM AT THE END OF
THE":PRINT" SESSION WILL SAVE THE";
66 PRINT" ENTERED":PRINT" WORDS FOR USE
AT THE NEXT PRACTICE."
67 PRINT "{5 DOWN}{12 RIGHT}{RVS}PRESS RE
TURN{OFF}"
68 GETR$:IFR$=""THEN68
69 IFR$=CHR$(13)THENRETURN
70 GOTO68
71 PRINT:PRINT "{CLR}{RVS}MISPELLED WORD
S:{OFF}":PRINT:REM PRINT OUT MISSPELL
ED WORDS,SCORE
72 FORP=0TO19:IFW$(P)="W"THENPRINTTAB(4)
A$(P)
73 NEXT
74 PRINT "{HOME}{19 DOWN}{RVS}SCORE ="100
-K*5
75 PRINT:PRINT "{3 SPACES}AGAIN ? (Y OR N
)"
76 GETR$:IFR$=""THENGOTO76
77 IFR$="Y"THENRUN1
78 IFR$<>"N"THEN76
79 POKE36869,240:POKEV,0:POKES,0
80 GOTO59
81 PRINT "{CLR}":POKE53280,2:S=54272:FORE
=STOS+28:POKEE,0:NEXT
83 POKE54296,15:POKE54277,18:POKE542
78,242
85 POKE 54276,33:POKE 54273,4:POKE54
272,48
87 FORT=1TO 300:NEXT:POKE54276,32:FORT
=1TO 400:NEXT
89 RETURN:REM{14 SPACES}FORE=STOS+28:POK
EE,0:NEXT:RETURN
90 S=54272:FORE=STOS+28:POKEE,0:NEXT
100 POKE54296,15:POKE54277,42:POKE54
278,250
110 POKE 54276,33:POKE 54273,23:POKE
54272,181
120 FORT=1TO 200:NEXT:POKE54276,32:FOR
T=1TO 500:NEXT
130 FORE=STOS+28:POKEE,0:NEXT
140 RETURN
```

## Computing For Kids

### Barney

```
3 REM:VIC-20/C64 BARNEY
4 REM:MODIFIED FROM
5 REM:TALKING HEAD
7 REM:COMPUTE! 9/82
10 PRINT "{CLR}"
20 FOR P=1 TO 800:NEXT P
30 N=1
40 GOSUB 670
50 GOSUB 100
60 GOSUB 260
70 GOSUB 480
80 GOSUB 260
90 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:GOTO 94
0
100 GOSUB 530
110 GOSUB 730
120 GOSUB 630
130 FOR P=1 TO 800:NEXT P
140 GOSUB 750
```



```

160 FOR P=1 TO 600: NEXT P
170 GOSUB 700
180 FOR P=1 TO 100: NEXT P
190 GOSUB 750
200 FOR P=1 TO 800:NEXT P
210 RETURN
230 IF N=14 THEN RESTORE
250 N=N+1
260 READ SNUM
270 FOR K=1 TO SNUM
280 GOSUB 330
290 FOR P=1 TO 1000: NEXT P
300 GOSUB 780
310 NEXT K
320 RETURN
330 PY=1
340 PX=10
350 READ M$
360 IF M$="-1" THEN RETURN
370 IF M$="*" THEN M$=N$
380 PRINTTAB(10);
390 PRINT M$:GOSUB 670
400 FOR P=1 TO 50: NEXT P
410 GOSUB 630
420 FOR P=1 TO 100: NEXT P
440 PY=PY+2
450 GOTO 350
480 PRINT"{HOME}{3 DOWN}";TAB(10);:INPUT
  N$
490 FOR P=1 TO 75: NEXT P
500 GOSUB 780
510 RETURN
530 PRINT"{CLR}"
540 PRINT"{3 SPACES}@@@"
560 PRINT"{2 SPACES}@@@@ "
570 PRINT" @=@=@=@ "
580 PRINT"JK{5 SPACES}JK"
590 PRINT:PRINT"<:{2 SPACES}Q{2 SPACES}:
  >"
600 PRINT"{HOME}{7 DOWN}:{6 SPACES}:"
610 PRINT:PRINT{10 SPACES}" M{5 @}N"
630 PRINT"{HOME}{7 DOWN} : {4 SPACES}:"
640 PRINT"{HOME}{7 DOWN} : M{ @}N : "
650 RETURN
670 PRINT"{HOME}{7 DOWN} : {3 @} : "
680 PRINT"{HOME}{7 DOWN} : --- : "
690 RETURN
700 PRINT"{HOME}{5 DOWN}JK 0 - JK"
710 FOR P=1 TO 1500: NEXT P
720 RETURN
730 PRINT"{HOME}{5 DOWN}JK - - JK"
740 RETURN
750 PRINT"{HOME}{5 DOWN}JK * * JK"
760 RETURN
780 PRINT"{HOME}"
790 FOR I=1 TO 5
820 PRINTTAB(10);"{11 SPACES}"
830 NEXT I
840 RETURN
860 DATA 3
870 DATA I'M BARNEY,-1
880 DATA THE{2 SPACES}CLONE,-1
885 PRINT
890 DATA YOUR NAME?,-1
900 DATA 3
910 DATA GREAT NAME,*,-1
920 DATA THANKS FOR,-1
930 DATA PLAYING!!!,*,-1
940 FOR X=1 TO 2500: NEXT X
950 PRINT"{CLR}"

```

```

960 PRINT"{11 DOWN} ***PRESS ANY KEY***"
970 GET A$:IF A$=""THEN GOTO 970
980 RUN

```

## The Viper

### ATTENTION PROGRAMMERS

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs" and "A Beginner's Guide To Typing In Programs" that appear before the Program Listings.

### Program 1: The Viper - VIC Version

```

10 DT=60:DIMMA(DT):DIMQ(100)
20 FORJ=0TO65:READJM:POKE828+J,JM:NEXT:F
  ORJ=1TODT:READMA(J):NEXT
30 PRINT"{WHT}{CLR}":POKE36864,4:POKE368
  79,111:POKE36878,15:S3=36877:C=30720:
  SC=7680
40 MZ=0:P=0:DR=0
50 V=36878:S1=36875:S2=36876:A=2:N=2:MM=
  0
60 GOSUB370
70 FORJ=7680TO7700:POKEJ+C,0:POKEJ,160:N
  EXT
80 FORJ=7701TO8185STEP22:POKEJ+C,0:POKEJ
  ,160:NEXT
90 FORJ=8184TO8164STEP-1:POKEJ+C,0:POKEJ
  ,160:NEXT
100 FORJ=8142TO7702STEP-22:POKEJ+C,0:POK
  EJ,160:NEXT
110 M=INT(RND(1)*506)+SC
120 IFPEEK(M)<>32THEN110
130 POKEM,42
140 S=INT(RND(1)*506)+SC
150 IFPEEK(S)<>32THEN140
160 POKES,90
170 SYS828
180 IFPEEK(1)-PEEK(2)=0THEN210
190 DR=PEEK(1)-PEEK(2)
200 IFDR=-21THENDR=1
210 IFPEEK(S+DR)=160ORPEEK(S+DR)=43THENP
  OKES,43:POKES+DR+C,2:POKES+DR,90:GOT
  O600
220 IFMM=1THENGOSUB700
230 SYS828
240 IFPEEK(S+DR)=42THENPOKES1,250:POKES2
  ,250:SYS828:P=P+1:N=N+2:MM=1:POKES1,
  0:POKES2,0
250 IFMM=0THENPOKEM,42
260 Q(A)=S+DR
270 SYS828
280 Z=A-N
290 IFZ<0THENZ=101+(A-N)
300 POKEQ(Z),32
310 A=A+1:SYS828
320 IFA>100THENA=0
330 POKES2,230:FORT=1TO2:NEXT:POKES2,0
340 SYS828
350 POKES,43:POKES+DR,90:S=S+DR:SYS828
360 FORT=1TOSK:NEXT:GOTO170
370 IFTR=1THENPRINT"{CLR}":GOTO450
380 N$="{24 SPACES}Z+++VIC-VIPER++++
  {3 SPACES}"

```



```

390 FORJ=1TO45:POKES2,230:FORT=1TO2:NEXT
:POKES2,0
400 PRINT"{HOME}{4 DOWN}"MID$(N$,J,22)
410 FORT=1TO150:NEXT:NEXT
420 PRINT
430 FORT=1TO2000:NEXT:PRINT"{CLR}
{4 DOWN} GET THE '*'S BUT{2 SPACES}
{DOWN}{4 SPACES}DON'T HIT ANYTHING
{DOWN}{3 SPACES}ELSE."
440 PRINT"{2 DOWN}{3 SPACES}USE JOYSTICK
.
450 PRINT"{3 DOWN}{3 SPACES}ENTER SKILL
LEVEL{2 SPACES}{DOWN}{3 SPACES}(EASY
)1 - 9(HARD)"
460 GETA$:IFA$=""THEN460
470 IFA$<"1"ORA$>"9"THEN460
480 SK=(10-(VAL(A$)))↑2
490 IFTR=1THENPRINT"{CLR}":GOTO520
500 PRINT"{CLR}{DOWN} YOU WILL GET 2 TIM
ES{DOWN}{2 SPACES}AS MANY POINTS WIT
H{DOWN}{3 SPACES}AN EASY MAZE"
510 PRINT"{2 DOWN} YOU WILL GET 5 TIMES
{DOWN}{2 SPACES}AS MANY POINTS WITH
{DOWN}{3 SPACES}A HARD MAZE":TR=1
520 PRINT"{2 DOWN} HIT 'H' FOR HARD MAZE
"
530 PRINT" HIT 'E' FOR EASY MAZE"
540 PRINT" HIT 'N' FOR NO MAZE"
550 GETB$:IFB$=""THEN550
560 IFB$="N"THENPRINT"{CLR}":RETURN
570 IFB$="H"THEN:PRINT"{CLR}":FORJ=1TODT
:POKESC+MA(J)+C,0:POKESC+MA(J),160:N
EXT:MZ=1:RETURN
580 IFB$<"E"THEN550
590 PRINT"{CLR}":MZ=2:FORJ=1TO32:POKESC+
MA(J)+C,0:POKESC+MA(J),160:NEXT:RETU
RN
600 POKES3,230:FORJ=15TO0STEP-.05:POKEV,
J:NEXT:POKES3,0
610 FORT=1TO1500:NEXT
620 IFMZ=1THENP=P*5
630 IFMZ=2THENP=P*2
640 R=P*(VAL(A$))
650 PRINT"{CLR}{2 DOWN}{YEL} YOUR SCORE:
"R
660 IFR>HSTHENHS=R
670 PRINT"{2 DOWN} HIGH SCORE:"HS
680 FORT=1TO3000:NEXT
690 GOTO30
700 M=INT(RND(1)*506)+SC:MM=0:SYS828
710 IFPEEK(M)<>32THENMM=1
720 RETURN
730 DATA169,128,141,19,145,169,0,133,1,1
33,2,169,127,141,34,145,162,119,236,
32,145
740 DATA208,4,169,1,133,1,169,255,141,34
,145,162,118,236,17,145,208,4,169,22
,133,1
750 DATA162,110,236,17,145,208,4,169,1,1
33,2,162,122,236,17,145,208,4,169,22
,133,2,96
760 DATA142,143,183,184,185,188,189,190,
205,212,222,223,224,225,226,227,234,
235,236,237
770 DATA238,239,249,256,271,272,273,276,
277,278,318,319,141,144,177,178,179,
180,181,192
780 DATA193,194,195,196,229,230,231,232,
265,266,267,268,269,280,281,282,283,
284,317,320

```

## Program 2: The Viper - 64 Version

```

100 DT=60:DIM MA(DT),Q(100),I$(15)
110 I$(14)=-40:I$(13)=40:I$(11)=-1:I$(7)
=1
120 I$(10)=-41:I$(6)=-39:I$(9)=39:I$(5)=
41:JOY=56321
130 FORJ=1TODT:READMA(J):NEXT
140 PRINT"{WHT}{CLR}"CHR$(142):C=54272:S
C=1024:POKE53281,2:POKE53280,8
150 MZ=0:P=0:DR=0
160 CURR=251:SPEED=49352:INDEX=SPEED+1:L
NGTH=INDEX+1:RTN=LN+1
170 SID=54272:V=SID+24:S1=SID:S2=SID:S3=
S2:A=2:N=2:MM=0:S4=SID+4
180 FORI=0TO24:POKESID+I,0:NEXT:POKESID+
1,25:POKESID+5,6:POKESID+6,0
190 POKESID+24,15
200 GOSUB410:POKESID+5,6:POKESPEED,19-SK
210 FORJ=1024TO1063:POKEJ+C,7:POKEJ,160:
NEXT
220 FORJ=1064TO2024STEP40:POKEJ+C,7:POKE
J,160:NEXT
230 FORJ=2023TO1984STEP-1:POKEJ+C,7:POKE
J,160:NEXT
240 FORJ=1983TO1063STEP-40:POKEJ+C,7:POK
EJ,160:NEXT
250 M=INT(RND(1)*1000)+SC
260 IFPEEK(M)<>32THEN250
270 POKEM,42:POKEM+C,1
280 S=INT(RND(1)*1000)+SC
290 IFPEEK(S)<>32THEN280
300 POKE S,90:POKES+C,16*RND(1):IF(PEEK(
56321)AND15)=15THEN300
310 S%=S/256:POKECURR,S-S%*256:POKECURR+
1,S%:POKEINDEX,0
320 POKELNGTH,N:SYS49152+5:REM MAIN LOOP
GOTO 170
330 HIT=PEEK(RTN)
340 IFHIT<>160ANDHIT<>214THEN360
350 S=PEEK(CU)+256*PEEK(CU+1):POKES,42:P
OKES+C,7:GOTO770
360 IFHIT<>42THEN320
370 POKESID,0:POKESID+5,9:POKES4,128:POK
ES4,129:P=P+1:N=N+2:FORT=1TO50:NEXT
380 POKES4,128:POKESID,0:POKESID+5,6:POK
ESID+24,0:POKESID+24,15
390 GOSUB880:POKEM,42:POKEM+C,1:POKESID+
24,0:POKESID+24,15
400 GOTO320
410 IFTR=1THENPRINT"{CLR}":GOTO470
420 GOSUB950
430 PRINT"{2 DOWN}{3 SPACES}GET THE '*'S
BUT":PRINT"{3 SPACES}DON'T HIT ANYT
HING ELSE"
440 PRINT"{2 DOWN}{3 SPACES}USE JOYSTICK
IN CONTROL PORT ONE."
450 FORJ=1TO45:POKESID,230:POKES4,33:FOR
T=1TO2:NEXT:POKES4,32:POKESID,0
460 POKESID+5,2
470 PRINT"{3 DOWN}"TAB(11)"ENTER SKILL L
EVEL:"
480 PRINTTAB(10)"[8]{RVS}{9 SPACES}111
11111112":SK=10
490 PRINT" {YEL}SLOW{WHT}{2 SPACES}<-
[8]{RVS}12345678901234567890{OFF}
{WHT} ->{2 SPACES}[6]FAST"
500 PRINTTAB(10)"{RVS}{WHT} -{CYN} -
{PUR} -{GRN} -{YEL} -[1] -[6] -
[7] -{BLU} -[3] -":PRINT

```



```

510 PRINT" [UP] "TAB(10+SK); "{WHT}↑{LEFT} "
;
520 J=15-(PEEK(56321)AND15):SK=SK+((JAND
8)=8)*(SK<19)-((JAND4)=4)*(SK>0)
530 IF(PEEK(56321)AND16)=0 THEN560
540 IF TI<T THEN530
550 T=TI+5:PRINT " :GOTO510
560 IFTR=1THENPRINT "{CLR}":GOTO610
570 PRINT "{CLR}{DOWN}{LOWER} YOU WILL GE
T 2 TIMES":PRINT " AS MANY POINTS WIT
H"
580 PRINT " AN EASY MAZE.
590 PRINT "{2 DOWN} YOU WILL GET 5 TIMES"
:PRINT " AS MANY POINTS WITH"
600 PRINT " A HARD MAZE.
610 PRINT "{2 DOWN}{LOWER}{8} PRESS
{WHT}LEFT{8} FOR HARD MAZE"
620 PRINT "{DOWN} PRESS {WHT}RIGHT{8} F
OR EASY MAZE"
630 PRINT "{DOWN} PRESS {WHT}JOYBUTTON
{8} FOR NO MAZE"
640 IFPEEK(56321)<>255 THEN640
650 MZ=0:J=PEEK(56321):IF(JAND16)=0THENP
RINT "{CLR}"CHR$(142);:RETURN
660 IF(JAND15)=15 THEN650
670 PRINT "{CLR}"CHR$(142):IF(JAND4) THEN
720
680 I=-1:PRINT "{HOME}{RVS}HARD MAZE"
690 FORJ=1TODT:POKESC+80+I*3({P}+MA(J)+C
,3:POKESC+MA(J)+80+I*320,160:NEXTJ
700 I=I+1:IFI<2 THEN690
710 MZ=1:RETURN
720 IF(JAND8)THEN570
730 I=-1:PRINT "{HOME}{RVS}EASY MAZE"
740 FORJ=1TO32:POKESC+MA(J)+C+80+320*I,3
:POKESC+MA(J)+80+320*I,160:NEXT
750 I=I+1:IFI<2THEN740
760 MZ=2:RETURN
770 POKESID,0:POKESID+5,15:POKES4,129:FO
RJ=15TO4STEP-.1:POKESID+24,J:NEXT
780 POKESID+24,15:FORT=1TO500:NEXT:POKES
4,128:FORT=1TO200:NEXT:POKESID+5,6
790 IFMZ=1THENP=P*5
800 IFMZ=2THENP=P*2
810 R=P*(SK+1)
820 PRINT "{CLR}{2 DOWN}{YEL} YOUR SCORE:
"R
830 IFR>HSTHENHS=R
840 PRINT "{2 DOWN} {CYN}HIGH SCORE:"HS
850 PRINT:PRINT "{WHT}PRESS {3}{RVS}JOY
BUTTON{OFF} {WHT}TO PLAY AGAIN."
860 IF(PEEK(56321)AND16)THEN860
870 GOTO140
880 M=INT(RND(1)*1000)+SC:MM=0
890 IFPEEK(M)<>32THEN880
900 RETURN
910 DATA 259,260,336,337,338,341,342,343
,376,383,411,412,413,414,415,416
920 DATA 423,424,425,426,427,428,456,463
,496,497,498,501,502,503,579,580
930 DATA 258,259,330,331,332,333,334,345
,346,347,348,349,418,419,420,421
940 DATA 490,491,492,493,494,505,506,507
,508,509,578,581
950 PRINT "{CLR}{WHT} "CHR$(142);:FORI=2T
O39:PRINT"*";:NEXT:PRINT:PRINT"
{4 DOWN}"
960 PRINT " ";:FORI=2TO39:PRINT"*";:NEXT
970 PRINT "{HOME}{DOWN}{3 @} {3 @}
{3 @} {3 @} {2 @} {3 @}"

```

```

980 PRINT "{2 SPACES}{RVS} {OFF}{RVS}
{OFF}{RVS}{OFF}{RVS}{OFF}
{RVS}{OFF}{RVS}{OFF}{RVS}
{OFF}"
990 PRINT " [T]{RVS} {OFF}{RVS}
{OFF}{RVS}{OFF}{RVS}
{OFF}{T}{RVS}{OFF}{RVS}
[*]"
1000 PRINT " [2 T] [T] [2 T]
{2 SPACES}[2 T] [2 T] [3 T]
{3 SPACES}":IFZZ=1THEN1070
1010 IFPEEK(900)<>232THENGOSUB1130
1020 FOR CO=3 TO 7:POKE894,CO:SYS893
1030 FORI=1TO20:PRINT "{HOME}{DOWN}"CHR$(
148)" {DOWN}{LEFT}"CHR$(148)"
{DOWN}{LEFT}"CHR$(148)" {DOWN}
{LEFT}"CHR$(148)" {DOWN}{LEFT}"
1040 POKESID+1,CO*2+I:POKES4,33:POKES4,3
2:NEXT
1050 FORI=1TO20:PRINT "{HOME}{DOWN} "CHR$(
20)"{DOWN} "CHR$(20)"{DOWN} "CHR$(
20)"{DOWN} "CHR$(20)"{DOWN} "
1060 POKESID+1,CO*2+20-I:POKES4,33:POKES
4,32:NEXT:NEXT
1070 FORI=1TO10:PRINT "{HOME}{DOWN}"CHR$(
148)" {DOWN}{LEFT}"CHR$(148)"
{DOWN}{LEFT}"CHR$(148)" {DOWN}
{LEFT}"CHR$(148)" {DOWN}{LEFT}"
1080 NEXT
1090 POKESID+1,60
1100 FORJ=15TO1STEP-1:POKE894,J:POKESID,
J*10:POKES4,33
1110 SYS893:POKES4,32:POKESID+24,J:NEXT:
POKESID+1,15:POKESID+24,15
1120 ZZ=1:RETURN
1130 FORI=893TO905:READA:POKEI,A:NEXT
1140 PRINT "{HOME}{8 DOWN}{RVS}READY TO P
LAY IN 5 SECONDS..."
1150 DATA 169, 1, 162, 0, 157, 40, 216,
232
1160 DATA 224, 160, 208, 248, 96
1170 FORI=49152TO49350:READA:CK=CK+A:POK
EI,A:NEXT
1180 PRINT "{HOME}{8 DOWN}{30 SPACES}"
1190 IF CK<>29203 THEN PRINT"ERROR IN DA
TA STATEMENTS!":POKE900,0:END
1200 RETURN
1210 DATA169,0,141,199,192,173,1,220
1220 DATA41,15,170,189,183,192,240,3
1230 DATA141,199,192,173,201,192,10,170
1240 DATA165,251,157,205,192,165,252,157
1250 DATA206,192,56,173,201,192,237,202
1260 DATA192,16,3,24,105,128,10,170
1270 DATA189,205,192,133,253,189,206,192
1280 DATA133,254,169,32,145,253,238,201
1290 DATA192,173,201,192,16,5,169,0
1300 DATA141,201,192,169,230,141,0,212
1310 DATA169,32,141,4,212,169,33,141
1320 DATA4,212,169,214,145,251,24,165
1330 DATA251,133,253,165,252,105,212,133
1340 DATA254,169,5,145,253,24,173,199
1350 DATA192,16,13,101,251,133,251,165
1360 DATA252,233,0,133,252,76,138,192
1370 DATA101,251,133,251,165,252,105,0
1380 DATA133,252,24,165,251,133,253,165
1390 DATA252,105,212,133,254,177,251,201
1400 DATA32,208,24,169,81,145,251,169
1410 DATA4,145,253,173,200,192,240,8
1420 DATA162,0,134,162,1977,162,208,252
1430 DATA76,5,192,141,203,192,96,0

```



1440 DATA0,0,0,0,41,217,1,0  
1450 DATA39,215,255,0,40,216,0,0

## Cylon Zap

### BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs" and "A Beginner's Guide To Typing In Programs" that appear before the Program Listings.

### Program 1: Cylon Zap - Setup (VIC)

```
10 POKE52,56:POKE56,56:POKE44,28:POKE716  
8,0:POKE631,131:POKE198,1  
20 PRINT"{CLR}**** CBM BASIC V2 ****":PR  
INT"{LEFT}"FRE(0);"BYTES FREE"
```

### Program 2: Cylon Zap - Special Instructions (VIC)

```
10 REM***CYLON ZAP****  
30 PRINT"{CLR}":POKE36879,8:POKE36869,19  
4  
35 GOSUB235  
40 PRINT"[3 DOWN]{2 SPACES}{RVS}LOADING  
{SHIFT-SPACE}CHARACTERS"  
45 FORA=5120TO7167:POKEA,PEEK(A+27648):N  
EXT  
50 FORA=5384TO5504  
55 READD  
60 IFD<>-1THENPOKEA,D:NEXT  
65 FORA=5584TO5632:READD:IFD<>-1THENPOKE  
A,D:NEXT  
70 FORA=5336TO5359:READD:POKEA,D:NEXT  
75 GOSUB400:PRINT"{UP} INSTRUCTIONS{OFF}  
{RVS}Y{OFF} OR {RVS}N{OFF}"  
80 GETA$:IFA$=""THENPOKE38342,INT(RND(1)  
*7+1):POKE38347,INT(RND(1)*7+1):GOTO8  
0  
85 IFA$="Y"THENPOKE36869,192:GOSUB130  
90 POKE36869,192:POKE198,1:POKE631,131:P  
RINT"[2 DOWN]{CLR}PLEASE WAIT WHILE G  
AMELOADS":END  
95 DATA24,24,60,126,24,24,126,255,1,19,5  
1,255,255,51,19,1,128,200,204,255,255  
,204,200  
100 DATA128,255,126,24,24,126,60,24,24,2  
4,24,60,24,60,126,219,195,3,7,44,254  
,254,44,7,3  
105 DATA192,224,52,127,127,52,224,192,19  
5,219,126,60,24,60,24,24,16,8,16,8,1  
6,8,16,8  
110 DATA145,74,44,113,142,52,82,137,0,0,  
0,170,85,0,0,0,-1  
115 DATA0,0,0,119,68,116,20,119,0,0,0,11  
9,85,87,86,117,0,0,0,112,64,96,64,11  
2  
120 DATA0,0,0,206,170,206,170,202,0,0,0,  
238,136,236,40,238,0,0,0,224,128,224  
,32,224,-1  
125 DATA0,0,0,206,170,202,170,206,0,0,0,  
139,218,171,138,139,0,0,0,56,160,56,  
136,56  
130 PRINT"{CLR}{RED}WELCOME TO CYLON ZAP  
"  
135 PRINT"YOU HAVE A BASE NAMED ALPHA"
```

```
{10 SPACES}:PRINT  
140 PRINT"{CYN}YOUR MISSION IS TO  
{4 SPACES}PROTECT THE":PRINT"NUCLEAR  
REACTOR"  
145 PRINT"{PUR}FROM THE KAMIKAZE STAR":P  
RINT"{UP}FIGHTERS"  
150 PRINT"{DOWN}{GRN}YOU HAVE 4 LASERS  
{5 SPACES}CONTROLLED BY THE  
{5 SPACES}JOYSTICK"  
155 PRINT"{BLU}YOU ALSO HAVE SMART  
{3 SPACES}BOMBS LAUNCHED BY THE FIRE  
BUTTON"  
160 PRINT"{DOWN}{YEL}ALL YOU DO IS POINT  
{3 SPACES}THE GUN AND THE LASER FIRE  
S AUTOMATICALLY"  
165 GOSUB190  
170 PRINT"{CLR}{PUR}{DOWN}THE FIGHTERS W  
ILL FLY FASTER AS MORE OF THEM  
{3 SPACES}ARE DESTROYED"  
175 PRINT"{DOWN}{YEL}BONUS BASE AND BOMB  
AT60 POINTS"  
180 PRINT"{BLU}{DOWN}[7 SPACES]{RVS}GOOD  
LUCK":GOSUB190:RETURN  
185 GOTO2070  
190 A$="{RVS}"  
195 FORL=1TO1000  
200 PRINT"HOME"  
205 PRINTTAB(2)A$;"{CYN}{20 DOWN}HIT RET  
URN TO CONT"  
210 GETR$:IFR$=CHR$(13)THENRETURN  
215 FORI=1TO333:NEXT  
220 IFA$="{RVS}"THENAS="{OFF}":GOTO230  
225 IFA$="{OFF}"THENAS="{RVS}":GOTO230  
230 NEXTL  
235 A$="{RED}*** * * *{3 SPACES}*** *  
{2 SPACES}*":X=LEN(A$):Z$="{DOWN}":G  
OSUB295  
240 A$="**{3 SPACES}* * *{3 SPACES}* * *  
*":X=LEN(A$):Z$="{2 DOWN}":GOSUB295  
245 A$="**{4 SPACES}*{2 SPACES}*  
{3 SPACES}* * * *":X=LEN(A$):Z$="  
{3 DOWN}":GOSUB295  
250 A$="***{2 SPACES}*{2 SPACES}*** * *  
*{2 SPACES}*":X=LEN(A$):Z$="  
{4 DOWN}":GOSUB295  
255 A$="{YEL}{2 SPACES}*** * * * *  
{2 SPACES}* *{2 SPACES}":X=LEN(A$):Z  
$="{7 DOWN}":GOSUB295  
260 A$="{4 SPACES}* * * * *{2 SPACES}* *  
{3 SPACES}":X=LEN(A$):Z$="{8 DOWN}":  
GOSUB295  
265 A$="{3 SPACES}*{2 SPACES}*** * * *  
{2 SPACES}* *{3 SPACES}":X=LEN(A$):Z  
$="{9 DOWN}":GOSUB295  
270 A$="{2 SPACES}*{3 SPACES}* * *  
{10 SPACES}":X=LEN(A$):Z$="{10 DOWN}  
":GOSUB295  
275 A$="{2 SPACES}*** * * *{4 SPACES}* *  
{2 SPACES}":X=LEN(A$):Z$="{11 DOWN}"  
:GOSUB295  
280 PRINT:PRINT  
285 A$="{YEL}{4 SPACES}BY M. DUDLEY  
{5 SPACES}":X=LEN(A$):Z$="{14 DOWN}"  
:GOSUB295  
290 GOTO40  
295 POKE36878,15  
300 FORI=1TOLEN(A$)  
305 PRINT"HOME}{DOWN}"Z$;SPC(X)LEFT$(A$  
,I):POKE36876,255-(I*7)  
310 X=X-1:NEXT:POKE36876,0:RETURN  
400 FORA=4700TO5000  
420 READD  
430 POKEA,D
```



```

440 NEXT
450 RETURN
500 DATA169,8,141,15,144,169,147,32,210,
255,162,8,160,8,32,240,255,169,18,32
,210,255,169
501 DATA169,32,210,255,169,127,32,210,25
5,169,146,32,210,255,169,32,32,210,2
55,169,18,32
502 DATA210,255,169,169,32,210,255,169,1
27,32,210,255,24,162,9,160,7,32,240,
255,169,169
503 DATA32,210,255,169,160,162,5,32,210,
255,202,224,0,208,248,169,127,32,210
,255,24
504 DATA162,10,160,7,32,240,255,169,146,
32,210,255,169,127,32,210,255,169,18
,32,210,255
505 DATA169,160,162,5,32,210,255,202,224
,0,208,248,169,146,32,210,255,169,16
9,32,210,255
506 DATA24,162,11,160,7,32,240,255,169,3
2,32,210,255,169,18,32,210,255,169,1
60,162,5,32
507 DATA210,255,202,224,0,208,248,169,14
6,32,210,255,169,32,32,210,255,24,16
2,11,160,7
508 DATA32,240,255,169,18,32,210,255,24,
162,12,160,7,32,240,255,169,169,32,2
10,255,169
509 DATA160,162,5,32,210,255,202,224,0,2
08,248,169,127,32,210,255,24,162,13,
160,7,32,240
510 DATA255,169,146,32,210,255,169,127,3
2,210,255,169,18,32,210,255,169,160,
162,5,32,210
511 DATA255,202,224,0,208,248,169,146,32
,210,255,169,169,32,210,255,24,169,1
46,32,210
512 DATA255,24,162,14,160,8,32,240,255,1
69,127,32,210,255,169,169,32,210,255
,169,32,32
513 DATA210,255,169,127,32,210,255,169,1
69,32,210,255,24,96

```

### Program 3: Cylon Zap, VIC Version

```

35 DEFFNA(A)=INT(RND(1)*X+A):TT=4348
40 V3=36876:N1=4106:N2=4590:N3=4338:N4=4
359:V1=36876
45 CS=36879:S0=36878:C=33792:V4=36877:W1
=30:W2=20:W3=10:W4=5:W5=1
50 A1$="D..":A2$="U..":A3$="D..":A4$="C.
":A5$="O.."
55 POKECS,8:PRINT "{CLR}":GOTO655
60 BASE=3:S1=1:S2=1:S3=1:S4=1:BOM=3:SC=0
65 POKE36869,205:X=15:Y=1:I=22
70 PRINT "{CLR}{WHT}":POKECS,8
75 DD=37154:P1=37151:P2=37152
80 GOSUB240
90 PRINT "{HOME}{CYN}";<"SC:PRINT "{HOME}
{DOWN}=>?"BA:PRINT "[ ]"BOM
95 POKEDD,127:P=PEEK(P2)AND128
100 J0--(P=0)
105 POKEDD,255:P=PEEK(P1)
110 J1--((PAND8)=0):J2--((PAND16)=0):J3=
--((PAND4)=0):FB--((PAND32)=0):G=42
115 POKETT,102
120 POKETT+C,INT(RND(1)*7+1)
125 IFJ3THEN320
130 IFJ1THEN345
135 IFJ2THEN370
140 IFJ0THEN395
145 IFFBANDBOM>0THEN855
150 A1=FNA(1)

```

```

155 A2=FNA(2)
160 A3=FNA(3)
165 A4=FNA(4)
170 IFA1=1ANDS1<>0THENS1=0:GOSUB900
175 IFA2=2ANDS2<>0THENS2=0:GOSUB900
180 IFA3=3ANDS3<>0THENS3=0:GOSUB900
185 IFA4=4ANDS4<>0THENS4=0:GOSUB900
190 IFS1=0ANDPEEK(N1+22)<>102THENN1=N1+I
:POKEN1+C,4:POKEN1,40:POKEN1-I,32
195 IFPEEK(N1+22)=102THENGOSUB585
200 IFS2=0ANDPEEK(N2-22)<>102THENN2=N2-I
:POKEN2+C,3:POKEN2,37:POKEN2+I,32
205 IFPEEK(N2-22)=102THENGOSUB585
210 IFS3=0ANDPEEK(N3+1)<>102THENN3=N3+Y:
POKEN3+C,5:POKEN3,39:POKEN3-Y,32
215 IFPEEK(N3+1)=102THENGOSUB585
220 IFS4=0ANDPEEK(N4-1)<>102THENN4=N4-Y:
POKEN4+C,6:POKEN4,38:POKEN4+Y,32
225 IFPEEK(N4-1)=102THENGOSUB585
230 IFBASE=0THENGOTO620
233 IFSC>50THENX=4
235 GOTO90
240 PRINT "{RED}":SYS4700
295 POKE4282+C,7:POKE4282,33:POKE4414+C,
7:POKE4414,36:POKE4345+C,7:POKE4345,
34
300 POKE4351+C,7:POKE4351,35
310 POKETT-1,102:POKETT+1,102:POKETT-22,
102:POKETT+22,102
315 RETURN
320 POKEV4,245:POKES0,15
325 FORF=4282TO4106STEP-22
330 IFPEEK(F-22)<>40THENPOKEF+C,7:POKEF,
41:FORT=1TO5:NEXT:POKEF,32:NEXT
335 IFPEEK(F-22)=40THENPOKEN1+C,2:POKEN1
,42:GOSUB480:POKEN1,32:N1=4106:S1=1
340 POKEV4,0:POKE4282,33:GOTO150
345 POKEV4,245:POKES0,15
350 FORF=4414TO4602STEP22
355 IFPEEK(F+22)<>37THENPOKEF+C,7:POKEF,
41:FORT=1TO5:NEXT:POKEF,32:NEXT
360 IFPEEK(F+22)=37THEN:POKEN2+C,2:POKEN
2,42:GOSUB480:POKEN2,32:N2=4590:S2=1
365 POKEV4,0:POKE4414,36:GOTO150
370 POKES0,15:POKEV4,245
375 FORF=4345TO4338STEP-1
380 IFPEEK(F-1)<>39THENPOKEF+C,7:POKEF,4
3:FORT=1TO5:NEXT:POKEF,32:NEXT
385 IFPEEK(F-1)=39THENPOKEN3+C,2:POKEN3,
42:GOSUB480:POKEN3,32:N3=4338:S3=1
390 POKEV4,0:POKE4345,34:GOTO150
395 POKES0,15:POKEV4,245
400 FORF=4351TO4359
405 IFPEEK(F+1)<>38THENPOKEF+C,7:POKEF,4
3:FORT=1TO5:NEXT:POKEF,32:NEXT
410 IFPEEK(F+1)=38THENPOKEN4+C,2:POKEN4,
42:GOSUB480:POKEN4,32:N4=4359:S4=1
415 POKEV4,0:POKE4351,35:GOTO150
420 POKES0,15:RESTORE
425 READP
430 IFP=-1THEN465
435 READD
440 POKEV3,P:POKEV1,P
445 FORN=1TOD:NEXT
450 POKEV3,0:POKEV1,0
455 FORN=1TO20:NEXT
460 GOTO425
465 RETURN
470 DATA217,200,213,200,223,200,227,100,
234,100,230,200
475 DATA227,100,234,100,230,200,223,200,
227,200,217,200,213,300,-1
480 POKES0,15

```



```

485 POKEV4,200
490 FORL=15TO0STEP-1
495 POKES0,L
500 NEXT:POKEV4,0
505 SC=SC+1
510 IFSC=30THENX=INT(X/2):Y=2
515 IFSC=50THENX=4:I=44:BOM=BOM+1
520 IFSC=60ORSC=110ORSC=150THENGOTO530
525 RETURN
530 PRINT"{CLR}{10 DOWN}{9 SPACES}BONUS"
535 PRINT"{6 SPACES}BASE - BOMB"
540 POKEV4,0:POKE36878,15:L=0
545 FORT=1TO10
550 POKE36876,220
555 NEXT
560 FORT=1TO10
565 POKE36876,230
570 NEXT
575 IFL<6THENL=L+1:GOTO545
580 BOM=BOM+1:BASE=BASE+1:SC=SC+5:POKE36876,0:POKE36878,0:PRINT"{CLR}":GOSUB240:GOTO510
585 POKES0,15:Q1=4348:Q2=4349:Q3=4370:Q4=4371:K=0:Q5=Q1-23:Q6=Q3+23:Q7=Q1+21:Q8=4372
590 POKEV4,200:KK=8
595 FORZ=15TO0STEP-2
600 POKES0,Z:GOSUB690:NEXT:POKECS,8:POKEV4,0
605 N1=4106:S1=1:N2=4590:S2=1:N3=4338:S3=1:N4=4359:S4=1:PRINT"{CLR}"
610 BASE=BASE-1:IFBASE<>0THENGOSUB240
615 RETURN
620 POKE36869,192:PRINT"{CLR}"
625 IFSC=>W1THENA5$=A4$:A4$=A3$:A3$=A2$
630 IFSC=>W1THENA2$=A1$:W5=W4:W4=W3:W3=W2:W2=W1:W1=SC:GOTO905
635 IFSC=>W2ANDSC<W1THENA5$=A4$:A4$=A3$:A3$=A2$:W5=W4:W4=W3:W3=W2:W2=SC:GOTO910
640 IFSC=>W3ANDSC<W2THENA5$=A4$:A4$=A3$:W5=W4:W4=W3:W3=SC:GOTO915
645 IFSC=>W4ANDSC<W3THENA5$=A4$:W5=W4:W4=SC:GOTO920
650 IFSC=>W5ANDSC<W4THENW5=SC:GOTO925
655 GOSUB820:PRINT"{HOME}{YEL}{21 DOWN}{4 SPACES}TO PLAY HIT {RVS}{CYN}Y"
660 GETZ$:IFZ$=" "THENFORCC=37897TO37900:POKECC,INT(RND(1)*7+1):NEXT
665 POKE38366,INT(RND(1)*7+1)
670 IFZ$=" "THEN660
675 IFZ$="Y"THEN60
680 IFZ$="N"THENPRINT"{CLR}{BLU}":POKECS,27:END
685 GOTO655
690 K=K+1:M=23:N=22:O=21:R=INT(RND(1)*7+1):IFK>3ANDKK<110THENPOKECS,KK:KK=KK+17
695 POKEQ1,G:POKEQ2,G:POKEQ3,G:POKEQ4,G:POKEQ5,G:POKEQ6,G:POKEQ7,G:POKEQ8,G
700 POKEQ1+C,R:POKEQ2+C,INT(RND(1)*7+1):POKEQ3+C,R:POKEQ4+C,INT(RND(1)*7+1)
705 POKEQ5+C,R:POKEQ6+C,INT(RND(1)*7+1):POKEQ7+C,R:POKEQ8+C,INT(RND(1)*7+1)
710 FORT=1TO10:NEXT
715 IFK>3THENG=46:PRINT"{CLR}"
720 IFK<8THENQ1=Q1-O:Q2=Q2-M:Q3=Q3+O:Q4=Q4+M:Q5=Q5-N:Q6=Q6+N:Q7=Q7-1:Q8=Q8+1:RETURN
725 PRINT"{CLR}":RETURN
730 PRINT"{3 DOWN}":CH=4171:E=1
735 POKEDD,127:P=PEEK(P2)AND128
740 J0=- (P=0)

```

```

745 POKEDD,255:P=PEEK(P1)
750 J2=- ((PAND16)=0)
755 FB=- ((PAND32)=0)
760 IFJ0THENE=E+1
765 IFJ2THENE=E-1
770 IFE=0THENE=26
775 IFE=27THENE=1
780 POKECH,E:POKECH+C,7
785 FORT=1TO100:NEXT
790 POKECH+C,6
795 IFFBANDCH=4171THENN1$=CHR$(E+64):CH=CH+1:E=1:GOTO735
800 IFFBANDCH=4172THENN2$=CHR$(E+64):CH=CH+1:E=1:GOTO735
805 IFFBANDCH=4173THENN3$=CHR$(E+64):CH=CH+1:E=32:GOTO735
810 IFCH=4174THENN5$=N1$+N2$+N3$:RETURN
815 GOTO735
820 PRINT"{CLR}{2 SPACES}{BLU} CYLON ZAP HEROS":PRINT:PRINT"{RED}{4 SPACES}BEST 5 SCORES{OFF}"
825 PRINT"{HOME}{DOWN}{CYN}{4 DOWN}{6 SPACES}"A1$"... "W1
830 PRINT"{PUR}{2 DOWN}{6 SPACES}"A2$".. "W2
835 PRINT"{GRN}{2 DOWN}{6 SPACES}"A3$".. "W3
840 PRINT"{BLU}{2 DOWN}{6 SPACES}"A4$".. "W4
845 PRINT"{RED}{2 DOWN}{6 SPACES}"A5$".. "W5
850 RETURN
855 POKES0,15:POKEV4,220
860 FORCO=127TO8STEP-17
865 POKECS,CO:POKES0,CO/8-1
870 FORT=1TO100:NEXT:NEXTCO
875 IFS1=0THENSC=SC+1:GOSUB510:POKEN1,32:N1=4106:S1=1
880 IFS2=0THENSC=SC+1:GOSUB510:POKEN2,32:N2=4590:S2=1
885 IFS3=0THENSC=SC+1:GOSUB510:POKEN3,32:N3=4338:S3=1
890 IFS4=0THENSC=SC+1:GOSUB510:POKEN4,32:N4=4359:S4=1
895 POKEV4,0:BOM=BOM-1:GOTO150
900 POKES0,15:FORM=180TO235STEP2:POKEV3,M:NEXT:POKEV3,0:POKES0,0:RETURN
905 PRINT"{HOME}#1 ENTER YOUR INITIALS":GOSUB420:GOSUB730:A1$=N5$:GOTO655
910 PRINT"{HOME}#2 ENTER YOUR INITIALS":GOSUB420:GOSUB730:A2$=N5$:GOTO655
915 PRINT"{HOME}#3 ENTER YOUR INITIALS":GOSUB420:GOSUB730:A3$=N5$:GOTO655
920 PRINT"{HOME}#4 ENTER YOUR INITIALS":GOSUB420:GOSUB730:A4$=N5$:GOTO655
925 PRINT"{HOME}#5 ENTER YOUR INITIALS":GOSUB420:GOSUB730:A5$=N5$:GOTO655

```

### Program 4: Cylon Zap, 64 Version

```

100 POKE52,48:POKE56,48:CLR
125 DATA28,149,100,25,30,100,33,135,100,37,162,50,50,60,50
130 DATA42,62,100,37,162,50,50,60,50,42,62,100,33,135,100
140 DATA28,49,100,25,30,100
145 FORX=1TO36:READRT:NEXT
150 PRINT"{CLR}":POKE53281,0:POKE53280,0:PRINTCHR$(14)
160 GOSUB590
170 PRINT"{3 DOWN}{11 SPACES}{RVS}LOADING {SHIFT-SPACE}CHARACTERS"
180 POKE56334,(PEEK(56334)AND254):POKE1,PEEK(1)AND251

```



```

190 FORA=0TO2047:POKE(A+12288),PEEK(A+532
48):NEXT
200 FORA=12552TO12672
210 READD
220 IFD<>-1THENPOKEA,D:NEXT
230 FORA=12288TO14335:READD:IFD<>-1THENPO
KEA,PEEK(A):NEXT
240 FORA=12504TO12527:READD:POKEA,D:NEXT
250 POKE1,55
260 POKE56334,PEEK(56334)OR1
270 GOSUB750:PRINT"{UP}{10 SPACES}INSTRUC
TIONS{OFF} {RVS}Y{OFF} OR {RVS}N
{OFF} "
280 GETA$:IFA$=""THENPOKE56079,INT(RND(1)
*7+1):POKE56084,INT(RND(1)*7+1):GOTO
280
290 IFA$="Y"THENPOKE53272,(PEEK(53272)AND
240)+12:GOSUB380
300 GOTO1000
310 DATA24,24,60,126,24,24,126,255,1,19,5
1,255,255,51,19,1,128,200,204,255,25
5,204,200
320 DATA128,255,126,24,24,126,60,24,24,24
,24,60,24,60,126,219,195,3,7,44,254,
254,44,7,3
330 DATA192,224,52,127,127,52,224,192,195
,219,126,60,24,60,24,24,16,8,16,8,16
,8,16,8
340 DATA145,74,44,113,142,52,82,137,0,0,0
,170,85,0,0,0,-1
350 DATA0,0,0,119,68,116,20,119,0,0,0,119
,85,87,86,117,0,0,0,112,64,96,64,112
360 DATA0,0,0,206,170,206,170,202,0,0,0,2
38,136,236,40,238,0,0,0,224,128,224,
32,224,-1
370 DATA0,0,0,206,170,202,170,206,0,0,0,1
39,218,171,138,139,0,0,0,56,160,56,1
36,56
380 PRINT"{CLR}{RED}WELCOME TO CYLON ZAP"
390 PRINT"YOU HAVE A BASE NAMED ALPHA"
{10 SPACES}:PRINT
400 PRINT"{CYN}YOUR MISSION IS TO
{2 SPACES}PROTECT THE":PRINT"NUCLEAR
REACTOR"
410 PRINT"{PUR}FROM THE KAMIKAZE STAR ":P
RINT" FIGHTERS"
420 PRINT"{DOWN}{GRN}YOU HAVE 4 LASERS
{2 SPACES}CONTROLLED BY THE
{4 SPACES}JOYSTICK"
430 PRINT"{BLU}YOU ALSO HAVE SMART BOMBS
LAUNCHED BY THE FIRE BUTTON"
440 PRINT"{DOWN}{YEL}ALL YOU DO IS POINT
THE GUN AND THE{6 SPACES}LASER FIRES
AUTOMATICALLY"
450 GOSUB500
460 PRINT"{CLR}{PUR}{DOWN}THE FIGHTERS WI
LL FLY FASTER THE MORE{3 SPACES}OF T
HEM YOU DESTROY "
470 PRINT"{DOWN}{YEL}BONUS BASE AND BOMB
AT 60 POINTS"
480 PRINT"{BLU}{DOWN}{9 SPACES}{RVS}GOOD
LUCK":GOSUB500:RETURN
490 GOTO65535
500 A$="{RVS}"
510 FORL=1TO1000
520 PRINT"{HOME}"
530 PRINTTAB(2)A$;"{CYN}{20 DOWN}HIT RETU
RN TO CONT"
540 GETR$:IFR$=CHR$(13)THENRETURN
550 FORI=1TO333:NEXT
560 IFA$="{RVS}"THENA$="{OFF}":GOTO580
570 IFA$="{OFF}"THENA$="{RVS}":GOTO580
580 NEXTL
590 A$="{RED}*** {3 SPACES}* {4 SPACES}
**{2 SPACES}*{2 SPACES}*":X=LEN(A$)
:Z$="{DOWN}":GOSUB710
600 A$="{4 SPACES}* {2 SPACES}*
{4 SPACES}* {2 SPACES}* *":X=LEN(A
$):Z$="{2 DOWN}":GOSUB710
610 A$="{5 SPACES}*{3 SPACES}*{4 SPACES}
* {2 SPACES}* *":X=LEN(A$):Z$="
{3 DOWN}":GOSUB710
611 A$="{5 SPACES}*{3 SPACES}*{4 SPACES}
* {2 SPACES}* *":X=LEN(A$):Z$="
{4 DOWN}":GOSUB710
620 A$="***{3 SPACES}*{3 SPACES}***
{2 SPACES}***{2 SPACES}*{2 SPACES}*
":X=LEN(A$):Z$="{5 DOWN}":GOSUB710
630 A$="{YEL}{2 SPACES}***{2 SPACES}***
{2 SPACES}***{2 SPACES}* {2 SPACES}
":X=LEN(A$):Z$="{8 DOWN}":GOSUB710
640 A$="{4 SPACES}*{2 SPACES}* *
{2 SPACES}* {2 SPACES}* {3 SPACES}
":X=LEN(A$):Z$="{9 DOWN}":GOSUB710
650 A$="{3 SPACES}*{3 SPACES}***
{2 SPACES}***{2 SPACES}* {3 SPACES}
":X=LEN(A$):Z$="{10 DOWN}":GOSUB710
660 A$="{2 SPACES}*{4 SPACES}* *
{2 SPACES}*{10 SPACES}":X=LEN(A$):Z$
="{11 DOWN}":GOSUB710
670 A$="{2 SPACES}***{2 SPACES}* *
{2 SPACES}*{4 SPACES}* {2 SPACES}":
X=LEN(A$):Z$="{12 DOWN}":GOSUB710
680 PRINT:PRINT
700 GOTO170
710 S=54272
711 POKE54296,15 :POKE54277,18:POKE54278,
240
712 POKE 54276,33
720 FORI=1TOLEN(A$):POKE54273,I+40
721 PRINT"{HOME}{DOWN}{8 RIGHT}"Z$;SPC(X)
LEFT$(A$,I):POKE54272,(I*2)+180
730 X=X-1:NEXT:FORG=15TO0STEP-1:POKE54296
,G:NEXT:POKE54,16
735 FORE=STOS+28:POKEE,0:NEXT:RETURN
750 FORA=49152TO49453
760 READD
770 POKEA,D
780 NEXT
790 RETURN
800 DATA169,12,141,33,208,169,147,32,210,
255,162,8,160,16,32,240,255,169,18,3
2
810 DATA210,255,169
820 DATA169,32,210,255,169,127,32,210,255
,169,146,32,210,255,169,32,32,210
825 DATA 255,169,18,32
830 DATA210,255,169,169,32,210,255,169,12
7,32,210,255,24,162,9,160,15,32
835 DATA 240,255,169,169
840 DATA32,210,255,169,160,162,5,32,210,2
55,202,224,0,208,248,169,127
845 DATA 32,210,255,24
850 DATA162,10,160,15,32,240,255,169,146,
32,210,255,169,127,32,210,255
855 DATA 169,18,32,210,255
860 DATA169,160,162,5,32,210,255,202,224,
0,208,248,169,146,32,210,255
865 DATA 169,169,32,210,255
870 DATA24,162,11,160,15,32,240,255,169,3
2,32,210,255,169,18,32,210

```



```

875 DATA255,169,160,162,5,32
880 DATA210,255,202,224,0,208,248,169,146
,32,210,255,169,32,32,210,255,24
885 DATA 24,162,11,160,7
890 DATA32,240,255,169,18,32,210,255,24,1
62,12,160,15,32,240,255,169,169
895 DATA 32,210,255,169
900 DATA160,162,5,32,210,255,202,224,0,20
8,248,169,127,32,210,255,24
905 DATA 162,13,160,15,32,240
910 DATA255,169,146,32,210,255,169,127,32
,210,255,169,18,32,210,255
915 DATA 169,160,162,5,32,210
920 DATA255,202,224,0,208,248,169,146,32,
210,255,169,169,32,210,255,24
925 DATA 169,146,32,210
930 DATA255,24,162,14,160,16,32,240,255,1
69,127,32,210,255,169,169,32
935 DATA 210,255,169,32,32
940 DATA210,255,169,127,32,210,255,169,16
9,32,210,255,24,96
1000 RESTORE:CLR
1060 DEFFNA(A)=INT(RND(1)*X+A):TT=1482
1070 POKE53272,(PEEK(53272)AND240)+12
1080 N1=1042:N2=1922:N3=1464:N4=1502:V1=3
6876
1090 CS=53281:C=54272:W1=30:W2=20:W3=10:W
4=5:W5=1
1100 A1$="D..":A2$="U..":A3$="D..":A4$="C
..":A5$="O.."
1110 POKECS,1:PRINT"{CLR}":GOTO2190
1120 BASE=3:S1=1:S2=1:S3=1:S4=1:BOM=3:SC=
0
1130 POKECS,12:X=15:Y=1:I=40
1140 PRINT"{CLR}{WHT}":POKECS,8
1150 GOSUB1450
1160 PRINT"{HOME}{WHT}SCORE"SC:PRINT"
{HOME}{DOWN}BASES"BA:PRINT"{WHT}BOM
BS"BO
1170 J0=15-(PEEK(56321)AND15)
1180 G=42:FB=(PEEK(56321)AND16)
1190 POKETT,102
1200 POKETT+C,INT(RND(1)*7+1)
1210 IFJ0=1 THEN1510
1220 IFJ0=2 THEN1570
1230 IFJ0=4 THEN1630
1240 IFJ0=8 THEN1690
1250 IFFB=0ANDBOM>0THEN2590
1260 A1=FNA(1)
1270 A2=FNA(2)
1280 A3=FNA(3)
1290 A4=FNA(4)
1300 IFA1=1ANDS1<>0THENS1=0:GOSUB2680
1310 IFA2=2ANDS2<>0THENS2=0:GOSUB2680
1320 IFA3=3ANDS3<>0THENS3=0:GOSUB2680
1330 IFA4=4ANDS4<>0THENS4=0:GOSUB2680
1340 IFS1=0ANDPEEK(N1+40)<>102THENN1=N1+I
:POKEN1+C,4:POKEN1,40:POKEN1-I,32
1350 IFPEEK(N1+40)=102THENGOSUB2050
1360 IFS2=0ANDPEEK(N2-40)<>102THENN2=N2-I
:POKEN2+C,3:POKEN2,37:POKEN2+I,32
1370 IFPEEK(N2-40)=102THENGOSUB2050
1380 IFS3=0ANDPEEK(N3+1)<>102THENN3=N3+Y:
POKEN3+C,5:POKEN3,39:POKEN3-Y,32
1390 IFPEEK(N3+1)=102THENGOSUB2050
1400 IFS4=0ANDPEEK(N4-1)<>102THENN4=N4-Y:
POKEN4+C,6:POKEN4,38:POKEN4+Y,32
1410 IFPEEK(N4-1)=102THENGOSUB2050
1420 IFBASE=0THENGOTO2130
1430 IFSC>50THENX=4
1440 GOTO1160
1450 PRINT"{RED}":SYS49152:POKECS,11
1460 POKE1362+C,1:POKE1362,33:POKE1602+C,
1:POKE1602,36:POKE1479+C,1:POKE1479
,34
1470 POKE1485+C,1:POKE1485,35
1480 POKETT-1,102:POKETT+1,102:POKETT-40,
102:POKETT+40,102
1490 POKETT-1+C,1:POKETT+1+C,1:POKETT-40+
C,1:POKETT+40+C,1
1500 RETURN
1510 POKE54296,15:POKE54273,33:POKE54272,
133:POKE54277,50:POKE54278,120
1520 POKE54276,129
1530 FORF=1362TO1042STEP-40
1540 IFPEEK(F-40)<>40THENPOKEF+C,1:POKEF,
41:FORT=1TO5:NEXT:POKEF,32:NEXT
1550 IFPEEK(F-40)=40THENPOKEN1+C,2:POKEN1
,42:GOSUB1830:POKEN1,32:N1=1042:S1=
1
1560 POKE54296,0:POKE1362,33:GOTO1260
1570 POKE54296,15:POKE54273,33:POKE54272,
133:POKE54277,50:POKE54278,120
1580 POKE54276,129
1590 FORF=1602TO1944STEP40
1600 IFPEEK(F+40)<>37THENPOKEF+C,1:POKEF,
41:FORT=1TO5:NEXT:POKEF,32:NEXT
1610 IFPEEK(F+40)=37THENPOKE2+C,2:POKEN2,
42:GOSUB1830:POKEN2,32:N2=1922:S2=1
-40
1620 POKE54296,0:POKE1602,36:GOTO1260
1630 POKE54296,15:POKE54273,33:POKE54272,
133:POKE54277,50:POKE54278,120
1640 POKE54276,129
1650 FORF=1479TO1464STEP-1
1660 IFPEEK(F-1)<>39THENPOKEF+C,1:POKEF,4
3:FORT=1TO5:NEXT:POKEF,32:NEXT
1670 IFPEEK(F-1)=39THENPOKEN3+C,2:POKEN3,
42:GOSUB1830:POKEN3,32:N3=1464:S3=1
1680 POKE54296,0:POKE1479,34:GOTO1260
1690 POKE54296,15:POKE54273,33:POKE54272,
133:POKE54277,50:POKE54278,120
1700 POKE54276,129
1710 FORF=1485TO1502
1720 IFPEEK(F+1)<>38THENPOKEF+C,1:POKEF,4
3:FORT=1TO5:NEXT:POKEF,32:NEXT
1730 IFPEEK(F+1)=38THENPOKEN4+C,2:POKEN4,
42:GOSUB1830:POKEN4,32:N4=1502:S4=1
1740 POKE54296,0:POKE1485,35:GOTO1260
1745 FORS0=54272TO54272+28:POKES0,0:NEXT
1750 POKE54296,15:POKE54277,53:POKE54278,
69:POKE54276,33
1770 RESTORE:FORGB=1TO12:READHA,LA,DU:POK
E54273,HA:POKE54272,LA
1780 FORT=1TODU:NEXTT
1790 NEXTGB:FORSE0=54272TO54272+28:POKES0,
0:NEXT
1800 RETURN
1810 DATA217,200,213,200,223,200,227,100,
234,100,230,200
1820 DATA227,100,234,100,230,200,223,200,
227,200,217,200,213,300,-1
1830 POKE54296,15:POKE54277,53:POKE54278,
67:POKE54276,129
1840 POKE54272,200:POKE54273,33
1850 FORL=15TO0STEP-1
1860 POKE54296,L
1870 NEXT:POKE54276,0
1880 SC=SC+1

```



```

1890 IFSC=30THENX=INT(X/2):Y=2
1900 IFSC=50THENX=4:I=80:BOM=BOM+1
1910 IFSC=60ORSC=110ORSC=150THENGOTO1930
1920 RETURN
1930 PRINT"{CLR}{10 DOWN}{10 SPACES}BONUS
";
1940 PRINT" BASE - BOMB":L=0
1950 POKE54296,15:POKE54277,50:POKE54278,
167:POKE54276,17
1960 FORT=1TO10
1970 POKE54272,230:POKE54273,33
1980 NEXT
1990 FORT=1TO10
2000 POKE54272,180:POKE54273,28
2010 NEXT
2020 IFL<6THENL=L+1:GOTO1950
2030 FORD=54272TO54272+28:POKED,0:NEXT
2040 BOM=BOM+1:BA=BA+1:SC=SC+5:PRINT"
{CLR}":GOSUB1450:GOTO1890
2050 POKE54296,14:Q1=1482:Q2=1484:Q3=1522
:Q4=1524:K=0:Q5=Q1-41:Q6=Q3+41:Q7=Q
1+39
2060 QB=1526:POKE54277,44:POKE54278,56:PO
KE54276,129
2070 POKE54272,200:POKE54273,34:KK=8
2080 FORZ=15TO0STEP-2
2090 POKE54296,Z:GOSUB2260:NEXT:POKECS,8-
POKE54276,0
2100 N1=1042:S1=1:N2=1922:S2=1:N3=1464:S3
=1:N4=1502:S4=1:PRINT"{CLR}"
2110 BASE=BASE-1:IFBASE<>0THENGOSUB1450
2120 RETURN
2130 PRINT"{CLR}"
2140 IFSC=>W1THENA5$=A4$:A4$=A3$:A3$=A2$
2150 IFSC=>W1THENA2$=A1$:W5=W4:W4=W3:W3=W
2:W2=W1:W1=SC:GOTO2730
2160 IFSC=>W3ANDSC<W2THENA5$=A4$:A4$=A3$:
W5=W4:W4=W3:W3=SC:GOTO2750
2170 IFSC=>W4ANDSC<W3THENA5$=A4$:W5=W4:W
4=SC:GOTO2760
2180 IFSC=>W5ANDSC<W4THENW5=SC:GOTO2770
2190 GOSUB2510:PRINT"{HOME}{BLK}{21 DOWN}
{12 SPACES}TO PLAY HIT {RVS}{BLK}Y"
2200 GETZ$:IFZ$=""THENFORCC=55312TO55315:
POKECC,INT(RND(1)*7+1):NEXT
2210 POKE56165,INT(RND(1)*7+1)
2220 IFZ$=""THEN2200
2230 IFZ$="Y"THEN1120
2240 IFZ$="N"THENPRINT"{CLR}{BLU}":POKECS
,27:END
2250 GOTO2190
2260 K=K+1:M=41:N=40:O=39:R=INT(RND(1)*7+
1)
2270 IFK>3ANDK<110THENPOKECS,KK:KK=KK+31
2280 POKEQ1,G:POKEQ2,G:POKEQ3,G:POKEQ4,G:
POKEQ5,G:POKEQ6,G:POKEQ7,G:POKEQ8,G
2290 POKEQ1+C,R:POKEQ2+C,INT(RND(1)*7+1):
POKEQ3+C,R:POKEQ4+C,INT(RND(1)*7+1)
2300 POKEQ5+C,R:POKEQ6+C,INT(RND(1)*7+1):
POKEQ7+C,R:POKEQ8+C,INT(RND(1)*7+1)
2310 FORT=1TO10:NEXT
2320 IFK>3THENG=46:PRINT"{CLR}"
2330 IFK<8THENQ1=Q1-O:Q2=Q2-M:Q3=Q3+O:Q4=
Q4+M:Q5=Q5-N:Q6=Q6+N:Q7=Q7-1:Q8=Q8+
1
2340 RETURN
2350 PRINT"{CLR}":RETURN
2360 PRINT"{3 DOWN}":CH=1160:E=1
2370 J0=15-(PEEK(56321)AND15)
2380 FB=PEEK(56321)AND16
2390 IFJ0=8THENE=E+1
2400 IFJ0=4THENE=E-1
2410 IFE=0THENE=26
2420 IFE=27THENE=1
2430 POKECH,E:POKECH+C,7
2440 FORT=1TO100:NEXT
2450 POKECH+C,1
2460 IFFB=0 ANDCH=1160THENN1$=CHR$(E+64):
CH=CH+1:E=1:GOTO2370
2470 IFFB=0ANDCH=1161THENN2$=CHR$(E+64):C
H=CH+1:E=1:GOTO2370
2480 IFFB=0 ANDCH=1162THENN3$=CHR$(E+64):
CH=CH+1:E=32:GOTO2370
2490 IFCH=1163THENN5$=N1$+N2$+N3$:RETURN
2500 GOTO2370
2510 POKE53281,1
2520 PRINT"{CLR}{2 SPACES}{BLK}{9 SPACES}
CYLON ZAP HEROS":PRINT:PRINT"{RED}
{12 SPACES}BEST 5 SCORES{OFF}"
2530 PRINT"{HOME}{DOWN}{BLK}{4 DOWN}
{14 SPACES}"A1$"...W1
2540 PRINT"{BLU}{2 DOWN}{14 SPACES}"A2$".
.."W2
2550 PRINT"{GRN}{2 DOWN}{14 SPACES}"A3$".
.."W3
2560 PRINT"{PUR}{2 DOWN}{14 SPACES}"A4$".
.."W4
2570 PRINT"{RED}{2 DOWN}{14 SPACES}"A5$".
.."W5
2580 RETURN
2590 POKE54296,15:POKE54277,43:POKE54278,
73:POKE54276,129
2600 FORCO=127TO8STEP-17
2610 POKECS,CO
2620 FORT=1TO100:NEXT:NEXTCO:POKECS,11
2630 IFS1=0THENSC=SC+1:GOSUB1890:POKEN1,3
2:N1=1042:S1=1
2640 IFS2=0THENSC=SC+1:GOSUB1890:POKEN2,3
2:N2=1922:S2=1
2650 IFS3=0THENSC=SC+1:GOSUB1890:POKEN3,3
2:N3=1464:S3=1
2660 IFS4=0THENSC=SC+1:GOSUB1890:POKEN4,3
2:N4=1502:S4=1
2670 FORS0=54272TO54272+28:POKES0,0:NEXT:
BOM=BOM-1:GOTO1260
2680 S=54272:FORE=STOS+28:POKEE,0:NEXT
2690 POKE54296,15:POKE54277,51:POKE54
278,84
2700 POKE 54276,17:FORJ=1TO40STEP4:POKE
54273,J:POKE54272,255-J-25:NEXT
2710 FORT=1TO 100:NEXT:POKE54276,32:FOR
T=1TO 50:NEXT
2720 FORE=STOS+28:POKEE,0:NEXT:RETURN
2730 PRINT"{HOME}NUMBER 1 ENTER YOUR INIT
IALS":GOSUB1745:GOSUB2360:A1$=N5$:G
OTO2190
2740 PRINT"{HOME}NUMBER 2 ENTER YOUR INIT
IALS":GOSUB1745:GOSUB2360:A2$=N5$:G
OTO2190
2750 PRINT"{HOME}NUMBER 3 ENTER YOUR INIT
IALS":GOSUB1745:GOSUB2360:A3$=N5$:G
OTO2190
2760 PRINT"{HOME}NUMBER 4 ENTER YOUR INIT
IALS":GOSUB1745:GOSUB2360:A4$=N5$:G
OTO2190
2770 PRINT"{HOME}NUMBER 5 ENTER YOUR INIT
IALS":GOSUB1745:GOSUB2360:A5$=N5$:G
OTO2190

```



# Beginner's Machine Language

# The Four-Speed Brake

## BEFORE TYPING...

Before typing in programs, please refer to "How To Type COMPUTE!'s Gazette Programs" and "A Beginner's Guide To Typing In Programs" that appear before the Program Listings.

### Program 1: RAMtest, VIC Version

```

800 FOR ADRES=864TO989:READ DATTA:POKE A
    DRES,DATTA:NEXT ADRES
864 DATA 84, 69, 83, 84, 32, 79
870 DATA 86, 69, 82, 32, 32, 66
876 DATA 65, 68, 32, 66, 89, 84
882 DATA 69, 32, 169, 16, 133, 58
888 DATA 169, 0, 133, 57, 160, 0
894 DATA 24, 141, 0, 30, 145, 57
900 DATA 209, 57, 240, 21, 152, 72
906 DATA 165, 58, 72, 32, 179, 3
912 DATA 104, 133, 58, 104, 168, 169
918 DATA 0, 230, 57, 208, 7, 230
924 DATA 58, 24, 105, 1, 208, 221
930 DATA 200, 208, 218, 32, 193, 3
936 DATA 230, 58, 165, 58, 201, 30
942 DATA 144, 207, 76, 203, 3, 162
948 DATA 10, 160, 0, 185, 106, 3
954 DATA 32, 210, 255, 200, 202, 208
960 DATA 246, 72, 152, 72, 32, 194
966 DATA 221, 104, 168, 104, 96, 169
972 DATA 13, 32, 210, 255, 160, 0
978 DATA 185, 96, 3, 32, 210, 255
984 DATA 200, 192, 10, 208, 245, 96
    
```

### Program 2: RAMtest, 64 Version

```

800 FOR ADRES=864TO995:READ DATTA:POKE A
    DRES,DATTA:NEXT ADRES
864 DATA 84, 69, 83, 84, 32, 79
870 DATA 86, 69, 82, 32, 32, 66
876 DATA 65, 68, 32, 66, 89, 84
882 DATA 69, 32, 169, 8, 133, 58
888 DATA 169, 0, 133, 57, 160, 0
894 DATA 24, 141, 0, 4, 145, 57
900 DATA 209, 57, 240, 21, 152, 72
906 DATA 165, 58, 72, 32, 179, 3
912 DATA 104, 133, 58, 104, 168, 169
918 DATA 0, 230, 57, 208, 7, 230
924 DATA 58, 24, 105, 1, 208, 221
930 DATA 200, 208, 218, 32, 193, 3
936 DATA 230, 58, 165, 58, 201, 160
942 DATA 144, 207, 76, 208, 3, 162
948 DATA 10, 160, 0, 185, 106, 3
954 DATA 32, 210, 255, 200, 202, 208
960 DATA 246, 72, 152, 72, 169, 32
966 DATA 32, 210, 255, 32, 201, 189
972 DATA 104, 168, 104, 96, 169, 13
978 DATA 32, 210, 255, 160, 0, 185
984 DATA 96, 3, 32, 210, 255, 200
    192, 10, 208, 245, 96, 0
    
```

### Program 1: Speed Brake For The 64

```

1 REM *** 4 SPEED BRAKE FOR THE C64 ***
60000 FORA=828TO894:READB:POKEA,B:NEXT:END
60010 DATA120,169,73,141,20,3,169,3,141,
    21,3,88,96,162,0,160,0,165,197,201
60020 DATA4,208,10,232,208,253,200,192,4
    8,208,248,160,0,201,5,208,6,232,20
    8
60030 DATA253,200,208,250,201,6,208,8,23
    2,234,234,208,251,200,208,248,32
60040 DATA159,255,165,197,201,3,240,247
    76,49,234
    
```

### Program 2: Speed Brake For The VIC

```

1 REM *** 4 SPEED BRAKE FOR THE VIC ***
60000 FORA=828TO894:READB:POKEA,B:NEXT:EN
    ND
60010 DATA120,169,73,141,20,3,169,3,141,
    21,3,88,96,162,0,160,0,165,197,201
    ,39,208,10
60020 DATA232,208,253,200,192,48,208,248
    ,160,0,201,47,208,6,232,208,253,20
    0,208,250,201
60030 DATA55,208,8,232,234,234,208,251,2
    00,208,248,32,159,255,165,197,201,
    63,240,247
60040 DATA76,191,234
    
```

## The Beginner's Corner

### Keyboard

```

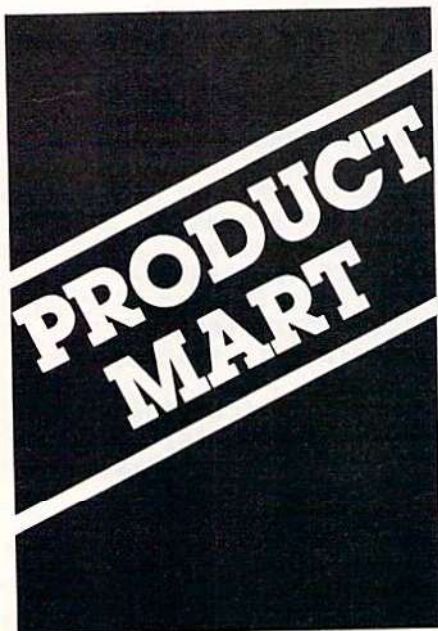
2 POKE36878,15:GOTO50
3 AA$="C":S=131:CC=0:RETURN
4 AA$="D":S=145:CC=3:RETURN
5 AA$="E":S=158:CC=6:RETURN
6 AA$="F":S=161:IFK=1THENCC=8:RETURN
7 CC=0:RETURN
8 AA$="G":S=173:IFK=1THENCC=11:RETURN
9 CC=3:RETURN
10 AA$="A":S=181:IFK=1THENCC=14:RETURN
11 CC=6:RETURN
12 AA$="B":S=189:IFK=1THENCC=17:RETURN
13 CC=9:RETURN
14 AA$="C":S=192:CC=11:RETURN
15 AA$="D":S=200:CC=14:RETURN
16 AA$="E":S=206:CC=17:RETURN
20 FORKI=1TO10:PRINT" {BLK}{RVS}
    {2 SPACES}{OFF} {RVS}{2 SPACES}{OFF}
    B {RVS}{2 SPACES}{OFF} {RVS}
    {2 SPACES}{OFF} {RVS}{2 SPACES}{OFF}
    B":NEXT
21 FORKI=1TO3:PRINT"{2 SPACES}[G]
    {2 SPACES}[G] B{2 SPACES}[G]
    {2 SPACES}[G]{2 SPACES}[G] B":NEX
    T:RETURN
22 FORKI=1TO10:PRINT" {BLK}{RVS}
    {2 SPACES}{OFF} {RVS}{2 SPACES}{OFF}
    {RVS}{2 SPACES}{OFF} B {RVS}
    {2 SPACES}{OFF} {RVS}{2 SPACES}{OFF}
    B":NEXT
    
```



```

23 FORKI=1TO3:PRINT"{2 SPACES}{G}
  {2 SPACES}{G}{2 SPACES}{G} B
  {2 SPACES}{G}{2 SPACES}{G} B":NEX
T:RETURN
50 PRINT"{CLR}{BLU}{3 DOWN}{2 SPACES}LET
'S LEARN NOTES":PRINT"{2 DOWN}
{4 SPACES}THE KEYBOARD{7 DOWN}"
51 PRINTTAB(5);"BY REGENA":SS=36876
52 DATA131,145,158,161,173,181,189,192
54 FORI=1TO8:READS:POKESS,S:FORD=1TO200:
NEXTD,I:POKESS,0:GOTO150
55 FORI=1TO10
56 PRINT"{CLR}{BLK}":K=INT(RND(1)*2)+1:O
NKGOSUB20,22:PRINT"{22 T}"
58 N=INT(RND(1)*7)+1:ONKGOTO60,62
60 ONNGOSUB3,4,5,6,8,10,12:GOTO64
62 ONNGOSUB6,8,10,12,14,15,16
64 POKE7944+CC,42:CS=38664+CC:FORD=1TO200
:POKECS,7:POKECS,2:NEXT
66 PRINT"{BLU}{3 DOWN}NAME THE NOTE"
68 GETA$:IFA$=""THEN68
70 IFA$=AA$THEN80
72 POKE36877,128:FORD=1TO400:NEXT:POKE36
877,0:GOTO68
80 POKE7944+CC,ASC(A$)-64:POKESS,S:FORD=
1TO400:NEXTD:POKESS,0:NEXTI
90 PRINT"{3 DOWN}{GRN}TRY AGAIN (Y/N)"
92 GETA$:IFA$="Y"THEN55
93 IFA$<>"N"THEN92
94 PRINT"{CLR}{BLU}":END
104 FORI=1TO7:PRINT"{2 SPACES}{RVS}
{OFF} {RVS} {OFF} {RVS} {OFF} B
{RVS} {OFF} {RVS} {OFF} B {RVS}
{OFF} {RVS} {OFF} {RVS} "":NEXT:RETUR
N
106 FORI=1TO3:PRINT"{2 SPACES}B B B B B
B B B B B":NEXT:PRINT"{22 Y}":RET
URN
108 PRINT"{2 DOWN}{GRN}PRESS RETURN";
109 GETA$:IFA$=""THEN109
110 IFASC(A$)<>13THEN109
111 PRINT"{CLR}{BLU}":RETURN
150 PRINT"{CLR}{DOWN}{BLU}A PIANO OR ORG
AN":PRINT"KEYBOARD HAS GROUPS
{3 SPACES}OF TWO BLACK ?KEYS"
152 PRINT"AND THREE BLACK KEYS. {2 DOWN}
{BLK}":GOSUB104:GOSUB106:GOSUB108
154 PRINT"{DOWN}LOOK AT A SET OF":PRINT"
TWO {BLK}BLACK{BLU} KEYS":PRINT"
{DOWN}THE NAMES OF THE KEYS ARE C, D
, AND E."
156 PRINT"{BLK}{DOWN}":FORI=1TO7:PRINT"
{3 SPACES}{H}{3 SPACES}{RVS}
{2 SPACES}{OFF}{3 SPACES}{RVS}
{2 SPACES}{OFF}{4 SPACES}{H}":NEXT
158 PRINT"{3 SPACES}{H}{4 SPACES}{H}
{4 SPACES}{H}{4 SPACES}{H}
{6 SPACES}{H}{RED}C{BLK}
{2 SPACES}{H}{RED}D{BLK}
{2 SPACES}{H}{RED}E{BLK}
{2 SPACES}{H}":PRINT"{3 SPACES}
{H}{4 SPACES}{H}{4 SPACES}{H}
{4 SPACES}{H}"
160 PRINT"{22 Y}":GOSUB108
164 PRINT"THE LETTER NAMES OF{3 SPACES}T
HE KEYS ARE THE{6 SPACES}ALPHABET LE
TTERS{6 SPACES}UP TO G.{3 DOWN}{BLK}
"
166 GOSUB104:PRINT"{2 SPACES}B
{SHIFT-SPACE}B{SHIFT-SPACE}B
{SHIFT-SPACE}B{SHIFT-SPACE}B
{SHIFT-SPACE}B{SHIFT-SPACE}B
{SHIFT-SPACE}B{SHIFT-SPACE}B
{SHIFT-SPACE}B{2 SPACES}{RED}F{BLK}B
{RED}G{BLK}B{RED}A{BLK}B{RED}B{BLK}B
{PUR}C{BLK}B{PUR}D{BLK}B{PUR}E{BLK}B
{RED}F{BLK}B{RED}G{BLK}B{RED}A{BLK}B
{RED}B{BLK}";
168 PRINT"{2 SPACES}B{SHIFT-SPACE}B
{SHIFT-SPACE}B{SHIFT-SPACE}B
{SHIFT-SPACE}B{SHIFT-SPACE}B
{SHIFT-SPACE}B{SHIFT-SPACE}B
{SHIFT-SPACE}B{SHIFT-SPACE}B
{SHIFT-SPACE}{22 Y}":GOSUB108:GOTO 55
170 END

```



**NEW!** MULTI-COLOR  
and  
HIGH RESOLUTION  
CHARACTER EDITORS  
For VIC-20 and CBM64

**Multi-Color**

- Character color and mode set for each character.
- Auxiliary, border, and screen color commands.
- 8 Power magnification for editing area.
- Cursor controls, RETURN, HOME, CLR, all work within the editing area.
- 4x5 animation area cycles through as many as eight user-selected character collections.
- Saves set to all output devices.
- Loads set from all input devices.
- VIC-20 VERSION
  - 64 characters edited at one time.
  - Fits in an unexpanded VIC-20.
- CBM64 and EXPANDED VIC-20 VERSION
  - 256 characters edited; 192 available at one time for CBM64, 64 available for VIC-20.
  - 90 degree rotation command.
  - Move command shifts pixels of one character, as a group, up, down, right, or left.
- VIC-20 program fits in any VIC with at least an BK expansion.

Please specify the type of computer, and if it is expanded.

CASSETTE-\$19.95 BIKETTE-\$21.95 (A RES ADD AT TAX)  
Just Another Software Co. Dept. AB  
P.O. Box 893, Palo Alto, CA 94301

**TREASURE  
RAIDERS**

**ARK**  
Innovations, Inc.  
Box 155  
(1813) School St.  
Amador City  
CA 95601

→ For VIC-20, unexpanded  
 → Escape with treasure  
 amidst pits, darts, boulders  
 → Fast response joystick control  
 → Machine language by Bonda Kirby  
 → Satisfaction guaranteed cassette  
 → ARK pays shipping & tax

**\$10.00**

\* VIC-20 is a reg. trademark of Commodore Business Machines, Inc.