Attack Of The Phantom Karate Devils For Commodore 64

Gregg Keizer
Assistant Book Editor

The almost-invisible ninjas approach from the left, materializing from nowhere, it seems. You kick, first with your right leg, then your left, their movement almost a blur. The ninja retreats. Flailing at your enemy with both hands, you advance toward the door which opens to the secret caverns under the Shining Moon Temple. The blows echo in the doorway. The passage is open, but before you enter, you leap into the air, jumping over the dagger whistling toward you. Safe for the moment, you catch your breath, then walk into the gloom, knowing that more ninjas will replace the one you've beaten.

Attack of the Phantom Karate Devils, a game for the Commodore 64 from Phantom Software, is one of the most graphically impressive games I've seen in a long time. Your figure, made up of several sprites which move independently, is lifelike in its animation. The arms, legs, torso, and head react to your commands and to the opponents' blows. The figure's response to your commands is quick and sure, and impressive to watch.

The object of the game is simple: Make your way past the opposing ninjas, their knives and throwing stars, as far as you can toward the Control World.

REVIEWS

As you move deeper into the ninjas' territory, you pass through caverns, across bridges, and enter new doorways. When you finally reach the Control World, you have only 15 seconds to avenge the Master, the climactic battle with the ninjas. Along the way, ninjas leap out at you, and daggers, stars, and arrows race across the screen toward you. All the while, your strength gradually fades away as you ward off the ninjas and their weapons. When your strength reaches zero, the game is over.

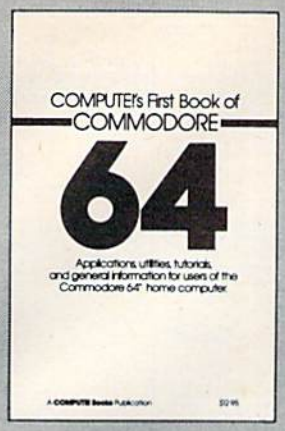
Playing The Game

Attack of the Phantom Karate Devils is a game on disk requiring a Commodore 64, a disk drive, a monitor, and a joystick plugged into port 2. Once you've seen the demonstration, press any key to read the joystick instructions displayed onscreen. The controls are rather complicated and somewhat difficult to master. For example, to jump you must press the fire button at the same time as you move the joystick to the up position. Simply pushing the joystick up, however, makes your figure kick with the left leg. Arm movement is even more elaborate. To punch with the left hand, you push the joystick to the left. Depending on whether you push it to the left and up, to the left, or to the left and down, the figure punches high, middle, or low with the left hand. The joystick controls are hard to get used to at first, but with a bit of practice they become more comfortable. The advantage of such controls is that the figure moves exactly as you want it to. You can make the figure walk, jump, hit a series of blows with either hand, or even kick with either leg. It all looks so lifelike that you soon forget your impatience with the joystick and your first fumbling attempts at controlling the figure.

Once you've read the instructions, you press any key to start the game. Your figure will appear in the Temple garden, on the right side of the screen. An enemy ninja soon leaps toward you from the left. A display on the far right-hand side of the screen shows your figure's present strength, points scored, DAN level (level of karate expertise), and high score and DAN level so far.

The most important are the strength and score displays. As your figure throws blows and is hit by ninjas and their weapons in turn, your strength level drops. Each punch or kick you make deducts points from your strength total. For instance, kicking with the left leg subtracts four points, while punching with the left hand subtracts only one point. Being hit by a dagger or star deducts ten points from your strength, so these should be avoided if possible. Each blow landed by a ninja reduces your strength by one point. Keep track of your strength total: Once it nears zero, you're in danger of losing the game. The only way to replace your strength is to pass through a doorway, or be promoted to a higher DAN level.

The ninjas always approach from the left. Wait until they are close, or advance toward them, before you begin throwing punches or kicks. Otherwise the blows will be wasted, and your strength will fall. The most powerful blow you have is a left kick. Using it when the game first begins, you'll quickly back up the ninja, and the door will open to the caverns. Once your strength falls below 100, you can't use the left kick. Your right



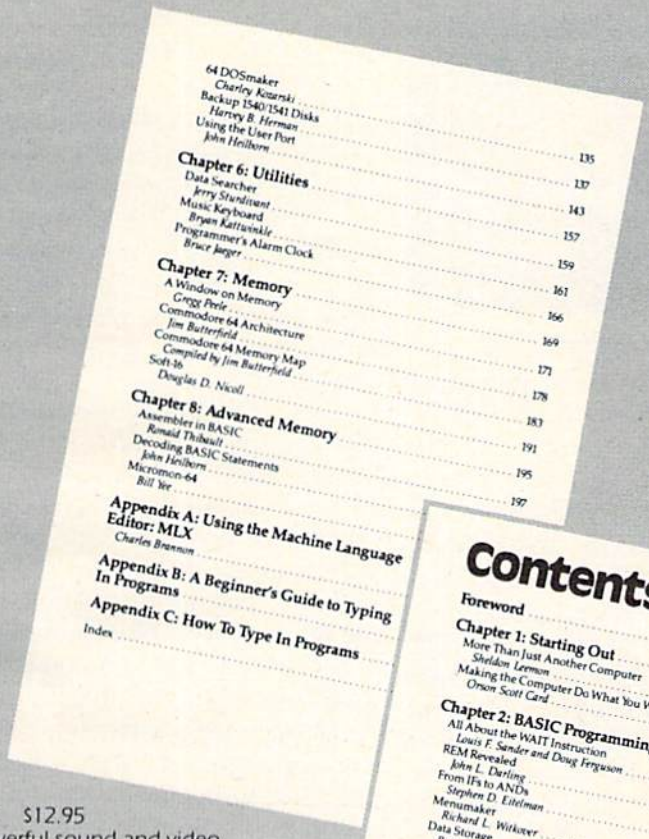
COMPUTE!'s First Book Of Commodore 64

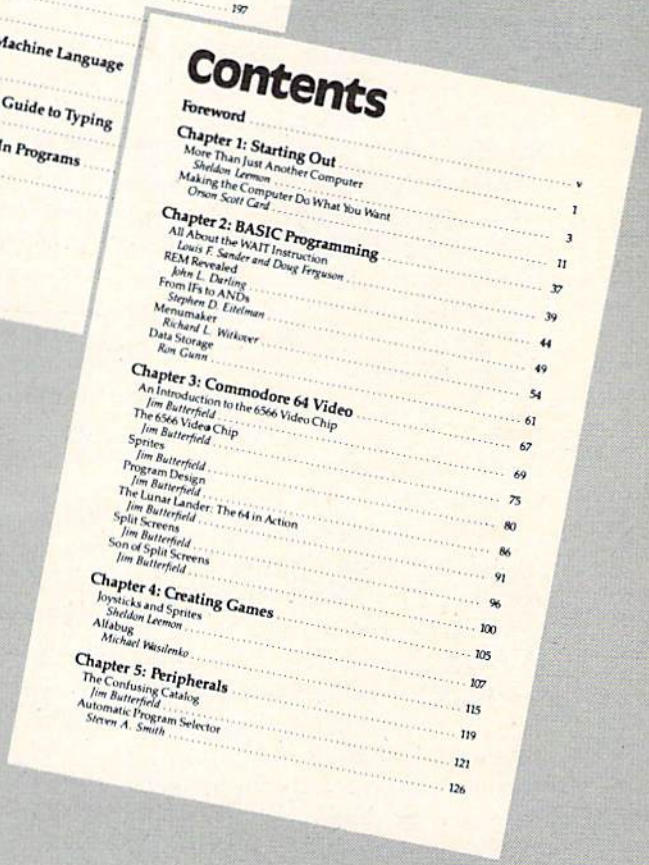
$12.95

The Commodore 64 is a powerful home computer. Its state-of-the-art features include sprite graphics and an advanced musical synthesizer chip. *COMPUTE!'s First Book of Commodore 64* is the definitive resource guide for owners and users of the 64. It includes some of the best articles and programs from *COMPUTE!* Magazine and *COMPUTE!'s GAZETTE*, plus many more that have never before appeared in print.

First Book Of Commmodore 64 contains something for everyone: BASIC programming techniques; data storage programs; examples of sprite graphics and musical synthesis; information about writing games, using peripherals, and working with many utilities (including a machine language assembler and a monitor); a complete memory map; and advanced programming techniques. Also includes many ready-to-type-in programs and games.

If you are a Commodore 64 owner who wants to learn more about using your computer to its full potential, this is the book for you.





Now Available From COMPUTE! Books

- *COMPUTE!'s First Book Of 64 Sound & Graphics* $12.95
Clear explanations to help you use all the 64's powerful sound and video features. Plus great programs for music synthesis, hi-res art, and sprite and character design.

- *COMPUTE!'s First Book Of Commodore 64 Games* $12.95
Packed full of games: Snake Escape, Oil Tycoon, Laser Gunner, Zuider Zee. Arcade-action machine language games for fast hands; strategy games for sharp minds.

- *COMPUTE!'s Reference Guide To 64 Graphics* $12.95
A complete, step-by-step tutorial for programming graphics. You'll like the clear writing, the example programs, and the full-featured sprite, character, and screen editors.

- *Creating Arcade Games On The 64* $12.95
The principles and techniques of fast-action game design, including custom characters, movement, animation, joysticks, sprites, and sound. With complete example game programs.

- *Commodore 64 Games For Kids* $12.95
Dozens of games for kids of all ages, making this an instant library of educational software. Learning, creativity, and excitement.

How To Order

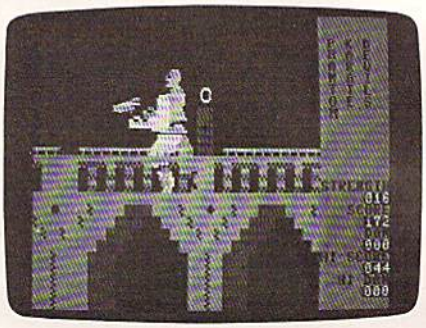
COMPUTE! Books are available at bookstores, newsstands, and computer stores nationwide. If your local store has run out of **COMPUTE! Books** or does not carry them, you can order them directly:

By Mail. Send a check or money order in U.S. funds (no cash please) to: Order Department, COMPUTE! Books, P.O. Box 5406, Greensboro, NC 27403. Please include $2 shipping/handling for each book you order. All orders must be prepaid. (C.O.D. orders are not accepted.) Allow 4–6 weeks for delivery.

By Phone. If you have VISA, MasterCard, or American Express, call our **toll-free number 800-334-0868** (919-275-9809 in NC) between 8:30 AM and 4:30 PM, EST. Your order will be processed promptly; we will bill your credit card (plus $2 shipping/handling). Allow 4–6 weeks for delivery.

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

With a pagoda in the background, you begin the game by fighting your way past a phantom opponent with deadly chopping hands.



Later, you must fight your way over a bridge. The scenes scroll horizontally across the screen.

leg kick will work until your strength falls under 60 points.

You'll see daggers and stars heading toward you from time to time. Like the ninjas, they'll always enter from the left. Ward them off with your hands, feet, or by jumping above them, out of their way. Remember that a dagger or star reduces your strength ten times more than a ninja's punch.

As you kick and punch, landing blows on the ninja in front of you, your score will rise. Soon the door to the caverns will open to you, and you can walk or jump through. Jumping through doors adds 50 points to your score, so try to do that if you can. As soon as you move through the door, your strength level is reset, and you're able to use your full abilities again. You'll find yourself in a tunnel-like cavern, which scrolls from left to right. Always move to the left if you can, especially if there is no ninja in front of you. The more you move, the closer you'll be to the next door.

This leads to the bridge, which also scrolls as you make your way to the left. More ninjas will appear, more daggers and stars will come at you, often at low level. They are difficult to see, for they are hidden by the bridge's structure at times. Leaping over them works the best. If you're good, scoring points as you pummel the ninjas, you'll be promoted in DAN to level 1. Your score will be reset to zero, and you continue toward the Control World.

The Control World contains a larger-than-life ninja who tosses bombs at you while daggers, stars, and arrows fly through the air. Defeating the ninja in the Control World is difficult, but not impossible. The screen may look like random garbage to you when you enter the Control World, but there's nothing wrong with your computer — that's the way it's supposed to look, according to the game's programmer, John Orthel. Unfortunately, it's very difficult to discern anything but your own figure when you're in this section of the game display. Considering the graphic excellence of the rest of the game, this was a disappointment. A dazzling display for this climactic scene of the game would have been far more impressive.

If you do beat off this last ninja's attacks, flinging aside the bombs he throws at you, the game ends and your score and DAN level will show on the screen.

Graphically Entertaining

Attack of the Phantom Karate Devils is a game which uses the Commodore 64's graphic capabilities well. Although the scrolling effect and flying objects such as daggers or stars add to the game, it is the animation of the player's figure which makes the game so much fun to watch and play. The movement of the figure is smooth and lifelike, but very responsive to the joystick. At times too responsive, for the figure lands blows so quickly (just like the actors in those wild karate movies) that you can lose strength points too fast if you're not careful. The only disappointment was in the final scene, the Control World. Had this been as appealing to the eye as the rest of the program, I would have been happy to play it again and again. As it stands, the game is excellent, but not perfect.

The use of sound in this game is also quite good. As the blows land or miss, you'll hear appropriate sounds, from sharp knocks to near misses. You can almost hear the sound of fabric moving as your figure kicks and punches.

The joystick controls, although complicated at first, are necessary to create the separate movements of the figure. The ability to punch with only one hand, for example, makes the game more realistic than if the controls were simplified.

This isn't a game you can sit down and play well without some practice. I had to play a

number of games before I reached the Control World, sometimes not getting any further than the opening scene. But the practice is worth it. Finally opening that door to the Control World will give you the satisfaction of having defeated the ninja hordes in true karate style.

Attack of the Phantom Karate Devils
Phantom Software
1116-A 8th Street, Suite 155
Manhattan Beach, CA 90266
$34.95

Multicolor Character Generator For VIC-20

Bill Gates

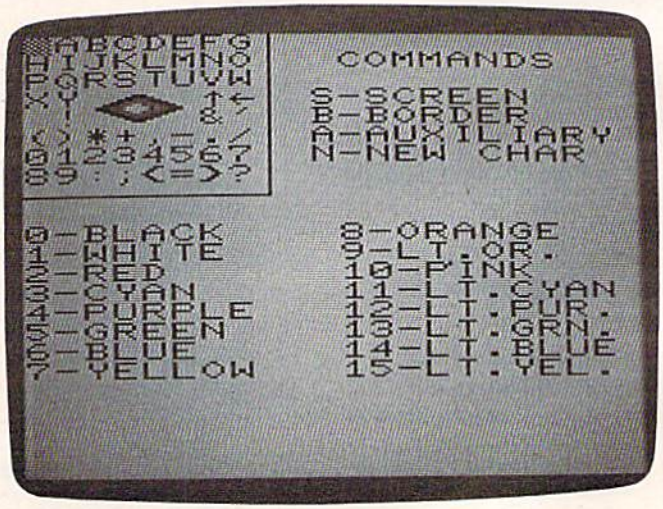
For the unexpanded VIC-20, "Multicolor Character Generator" greatly simplifies the complex task of designing multicolored characters. If you are unfamiliar with the basic techniques of custom characters, see "Introduction To Custom Characters For VIC And 64" and related articles in the November 1983 issue of COMPUTE!'s GAZETTE.

The Commodore VIC-20 has the capability to produce stunning, high-resolution color graphics. You can design a spaceship that has red engines, blue wings, and an orange nose cone—or even a monster with a green head, pink body, and purple legs. Unfortunately, designing multicolor characters has been a tedious time-consuming process which involved laboriously translating characters on paper into numbers that the computer can understand. "Multicolor Character Generator" solves this problem.

Multicolor Character Generator is an interactive *utility* program (a program which helps you with your programming) that makes designing and using your own multicolor characters easy. It allows you to design your custom characters in four different colors, using a pseudo-cursor and an enlarged picture of the character. Then it generates the code necessary to use your multicolored creation in your own programs. This article will show you how to use Multicolor Character Generator and explains how to use the characters in your own programs.

Using The Program

When you run the program, the screen clears and a display is set up showing 64 standard characters, a list of commands, and a list of all 16 color choices. Now, choose the colors you wish to use. Each character has four color choices: screen, border, auxiliary, and character colors. The character color can be different for each character, but the screen, border, and auxiliary colors must be the same for all characters on the screen. Therefore, remember that even though the program allows you to change these three colors at any time, changing



The main menu screen for "Multicolor Character Generator." Notice the diamond-shaped multicolor character which has been designed with ordinary characters.

$u¢h A Deal

*Lowest Prices — Guaranteed!**

COMMODORE 64

Just for You!

PERSONAL FINANCE

Continental Home Accountant (D)	$47
Continental Tax Advantage (D)	$33
Continental FCM First Class Mail	$29
Softsync Personal Accountant (D&C)	$23
CheckEase (C&D)	$24
Timeworks Electronic Checkbook (D&C)	$19
Timeworks Money Manager (D&C)	$19
M.S.I. Inventory (D)	$16
Creative Household Finance (D)	$23
Creative Household Finance (C)	$19
Creative Home Inventory (D)	$13
Creative Home Inventory (C)	$10
Creative Loan Analyzer (D)	$13
Creative Loan Analyzer (C)	$10

WORD PROCESSING

Broderbund Bank St. Writer (D)	$45
Hesware Omniwriter (D)	$49
Data 20 Word Manager (D&C)	$24
Quick Brown Fox (CT)	$42
Rainbow Writers Asst. (D)	$49
Hesware Heswriter (CT)	$29
On-Line HomeWard (D)	$39

ELECTRONIC SPREADSHEETS

Hesware Multiplan (D)	$75
Hesware Omnicalc (D)	$37
MSI Practicalc (D&C)	$35
MSI Programmable Spreadsheet (D)	$55
B. SKY CALC Result Easy (D)	$49
B. SKY CALC Result Advd. (D)	$99
Rainbow Spreadsheet Asst. (D)	$49
Rainbow Spreadsheet w/Graphics (D)	$89

HOME APPLICATIONS

Spinnaker Aerobics (D)	$33
Softsync Computer Mechanic (D)	$19
Softsync Computer Mechanic (C)	$16
Creative Car Costs (D)	$13
Creative Car Costs (C)	$10
Creative Decision Maker (D)	$13
Creative Decision Maker (C)	$10
Hesware Time & Money Manager (D)	$39
Timeworks Data Manager (D&C)	$19
Compuserve Starter Kit (5 hrs)	$29

ART & MUSIC

Epyx Fun with Art (CT)	$27
Epyx Fun with Music (CT)	$27
Hesware Synthesound (CT)	$33
Hesware Paintbrush (CT)	$19
Spinnaker Delta Drawing (CT)	$26
Spinnaker Delta Music (CT)	$26
Koala SpiderEater	$23
Koala Geometric Designs	$23
Koala Crystal Flowers	$23
Koala Logo Designs	$29

EDUCATION

Spinnaker Alphabet Zoo (CT)	$23
Spinnaker Cosmic Life (CT)	$23
Spinnaker Facemaker (CT)	$23
Spinnaker Fraction Fever (CT)	$23
Spinnaker Kids on Keys (CT)	$23
Spinnaker Kindercomp (CT)	$19
Spinnaker Story Machine (CT)	$26
Spinnaker Up For Grabs (CT)	$26
Spinnaker Delta Music (CT)	$26
Spinnaker Delta Drawing (CT)	$26
Creative I Am Your 64 I (D)	$19
Cretive I Am Your 64 II (D)	$19
Lightning Mastertype (D)	$27
Sirius Type Attack (D)	$26
Hesware Turtle Graphics II (CT)	$39
Hesware Turtle Toyland (CT)	$26
Hesware Turtle Toyland Jr. (CT)	$26
Hesware Type 'N' Writer (CT)	$26

COMMODORE 64 cont.

LANGUAGES & UTILITIES

Hesware 6502 Pro Devel Sys (D)	$19
Hesware Hesmon 64 (CT)	$26
Hesware 64 Forth (CT)	$45
Acess Spritemaster (D&C)	$23
Timeworks Programmer Kits I, II, III (D&C) each	$19

BUSINESS SOFTWARE

Data 20 Business Manager (D) (A.P., A.R., G.L.)	$79
Total Business 3.6 (D)	$65
Total Time Mgmt. (D)	$25
Total Lable-Mail (D)	$15
Total Research Asst. (D&C)	$25

GAMES

Epyx Dragon Riders of Pern (D&C)	$25
Epyx Silicon Warrier (CT)	$25
Sega Congo Bongo (CT)	$25
Infocom Enchanter (D)	$33
Infocom Infidel (D)	$33
Synapse Blue Max (D&C)	$22
Sublogic Pinball (D&C)	$20
Hesware Maze Master (CT)	$26
Broderbund Choplifter (CT)	$27
Broderbund Lode Runner (D)	$23
Sirius Gruds In Space (D)	$23
Sieria On Line Sammy Lightfoot (CT)	$29
Datamost Aztec (D)	$25
Commdata Supercuda (C&D)	$19
Commdata Pakacuda (C&D)	$19
Epyx Pitstop (CT)	$27

COMMODORE VIC 20

PERSONAL PRODUCTIVITY

Creative Home Office (D)	$22
Creative Home Office (C)	$19
Creative Household Finance (D)	$17
Creative Household Finance (C)	$13
Creative Home Inventory (D)	$13
Creative Home Inventory (C)	$10
Thorn Music Composer (CT)	$25
M.S.I. Practicalc Plus (D)	$35
M.S.I. Practicalc Plus (T)	$33
M.S.I. Practicalc (D)	$32
M.S.I. Practicalc (T)	$29
Hesware Synthesound (CT)	$19
Hesware Vic Fourth (CT)	$39
Hesware Hes Mon (CT)	$26
Hesware Heswriter (CT)	$26
Quick Brown Fox (CT)	$42
Hesware 6502 Pro Dev Sys	$19
Epyx Fun with Art (CT)	$26
Epyx Fun with Music (CT)	$26
Broderbund Mastertype (CT)	$24

COMMODORE VIC 20 cont.

EDUCATIONAL SOFTWARE

Hesware-Spinnaker KinderComp (CT)	$23
Story Machine (CT)	$23
Face Maker (CT)	$23
Kids On Keys (CT)	$23
Alphabet Zoo (CT)	$23
Hesware Turtle Graphics (CT)	$26
Creative Pipes (C)	$19
Creative Spills & Fills (C)	$19
Creative Hangman & Hangmath (C)	$10
Creative Math Hurdle & M.Maze (C)	$10

GAMES

Creative Choplifter (CT)	$19
Epyx Temple of Apshai (C)	$24
Sega Congo Bongo (CT)	$25
Sega Star Trek (CT)	$25
Hesware Pharaohs Curse (CT)	$24

Rock Bottom Prices on Peripherals!

DATA 20

Vic 40-80 Display Manager	$77
C64 Video Pak 80 Includes Word Manager FREE	$133
Parallel Printer Interface	$45

HESWARE

Hescard Vic 5 Slot	$39
HesModem Vic & C64	$49

KOALAPAD

C64, Vic, Atari	$67
Apple	$99

PRINTERS

BMC BX80	$249
GEMINI 10X	$269
AlphaCom 40 Column	$99
AlphaCom 80 Column Vic, C64, Atari Cable with AlphaCom FREE	$169

MODEMS

Anchor Singleman Mark II Atari (300 BAUD)	$75

MONITORS

BMC 12" Green	$85
BMC 12" Amber	$99
BMC 13" Composite Color	$239
Monitor cable w/above	$10

PERIPHERALS! cont.

WICO JOYSTICKS

The Boss	$14
Red Ball	$21

CARDCO

Numeric Keypad C64	$29
Graphic Printer Interface	$69
Economy Printer Interface	$39
Commodore 64 5 Slot	$49

And Now VIDEO MOVIES

Fantastic Savings on All Video Movies VHS-BETA

Walt Disney
Children's Video Library
MGM/UA
Paramount
CBS-Fox
Warner
Thorn Emi
RCA-Columbia
Vestron
MCA
UID America

We've Got It All

Adventure
Comedy
Horror
Sports
Drama
Sci-Fi
Music

ALL AT GREAT PRICES!

Write for free catalog
or call (602) 968-9128
for more info!

$u¢h A Deal

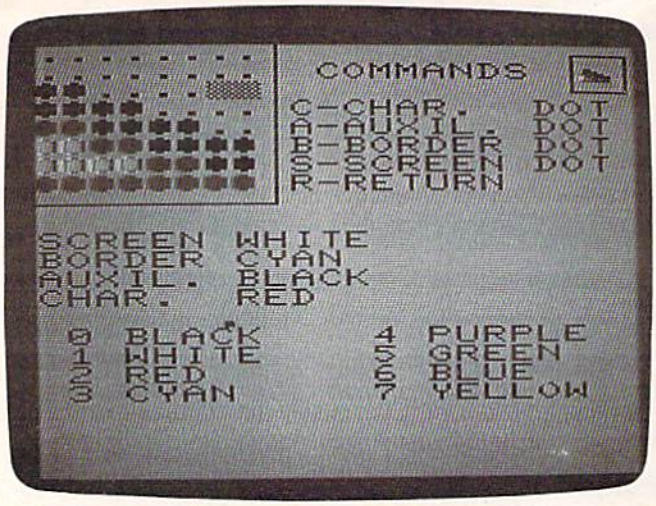
CALL TOLL FREE
1-800-431-8697
Orders Only!

For Information, Customer Service Release Dates, etc.
Call
602-968-9128

*TERMS OF OFFER: If you find a price for any software or peripheral in this issue that is lower than our advertised price, we'll guarantee to beat it! Valid only on products appearing in our ad. Valid only on product in similar in-stock conditions. Valid only on prices appearing in pring in this issue.

ORDERING & TERMS: Send cashier check, money order; personal/company checks allow 3 weeks bank clearance. VISA/MasterCard accepted. Provide phone number with order. SHIPPING: Software add $3.00; Hardware add $10.00. Arizona residents add 6% sales tax. Returns must have authorization number (call 602-968-9128) for authorization number). All returned merchandise subject to restocking fee and **must** come with **all** original packaging. No returns allowed after 30 days from shipping date. Prices are for cash; VISA and MasterCard add 3%. Prices subject to change without notice. All products subject to availability from manufacturers and/or suppliers.

(C) = Tape (D) = Disk (CT) = Cartridge



Multicolor characters are designed with this menu screen, dot by dot.

them will affect all the characters you have designed. So, it would be wise to stick with your three initial color choices unless you plan to redo all the characters you've created.

To make these three choices, press either S, B, or A to select or change the screen, border, or auxiliary colors, respectively. A question appears below the color choice and specifies the range of colors allowed. Type the number that corresponds to the color you want, then press RETURN. Once you have picked a combination of the three colors, use the cursor control keys to move the pseudo-cursor (the graphic symbol in "home" position) to one of the 64 standard characters you wish to replace with the custom character you are about to design. To enter the designing mode, press N for "new character."

When you enter the designing mode, the screen clears and a new display appears. You are then asked to select the character color with the choices listed above the question. Enter the proper number, then press RETURN. You are now ready to make your own multicolor character.

In the upper-left quarter of your screen appears an 8×8 character grid of dots. This is an enlarged representation of the character's pixel pattern in which you will create your multicolor character. In the "home" position is your two-character pseudo-cursor. It is two characters wide because in multicolor mode each dot is two pixels wide. This cursor is also moved with the cursor controls. Move the cursor to where you want a multicolor dot drawn, and press either S, B, A, or C to draw the dot in either screen, border, auxiliary, or character color. Drawing a dot in screen color erases a previously drawn dot.

A very helpful feature of the program is that the actual character in multicolor mode appears in a small box in the top-right corner of the screen, so you may see what you are really creating in the enlarged grid. When you have finished the character, press R and all the numbers for that character will appear. Write these numbers down, because you will need them for your own programs. Pressing another key returns you to the original mode of the program so you can select another character to be replaced.

Incorporating The Characters Into Your Own Programs

Here's an outline of the steps necessary for incorporating into your own programs the multicolor characters you design with the Multicolor Character Generator. The outline also shows how to use the numbers it generated.

1. First, you must reserve memory for the characters in RAM by using the following line of BASIC: POKE 52, 28: POKE 56, 28: CLR. (For the unexpanded VIC.)

2. Next, POKE 36869, 255. This tells the VIC where to find your characters in memory.

3. Here's where you start using the numbers given in Multicolor Character Generator. At the top of the screen on which the numbers are displayed appears something like 7168 TO 7175, followed by eight numbers. You use this by setting the following lines of BASIC:

```
FOR A=7168 TO 7175: READ B: POKE A, B: NEXT
DATA 255, 60, 60, 255, 7, 5, 60, 255
  (the eight numbers)
```

This places the data for your multicolor character in the RAM that was reserved.

4. POKE 36879 with the number given in the program. This sets screen and border colors.

5. POKE 36878 with the number given in the program. This sets auxiliary color. Note that this memory location also controls volume, so if you are using sound in your program, simply add 15 to the number given in the program.

6. These last POKEs depend on where you place your multicolor character on the screen. Use the charts on page 144 of *Personal Computing on the VIC-20*, the manual that comes with every VIC. Then POKE from 38400 to 38905 (use chart) with the number given in the program—it is the one following 38400. And, finally, POKE from 7680 to 8185 (use chart) with the number given in the program—it is the one following 7680. These POKEs are for the character color and screen memory.

Some Insights

In the process of writing this program, I developed some insights which I will pass along. You may notice that when designing characters with the Multicolor Character Generator, if the screen color value is greater than seven, a zero and graphic symbol appear in the bottom-right corner. (They are also there if the screen color value is less than

PC GALLERY

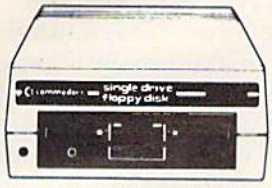
P.O. Box 3354 Cherry Hill, N.J. 08034

Toll FREE (800) 992-3300

For Information Call (609) 424-7106

Call us for reliable service, experience and affordable prices!

CMD 64 Computer
$219



1541 Disk Drive
$245

1520 Color Plotter	$165
1525 Printer	$215
1526 Printer	$319
1530 Datasette	$64
1600 Modem	$59
1650 Auto Modem	$88
1110 8K Memory Exp	$52
1111 16K Memory Exp	$68
1011 RS 232 Terminal Interface	$42
1211 Super Expander	$53
1212 Programmer's Aid Cartridge	$39
1213 Machine Language Monitor	$39

Color Monitor 1702

14" screen, outstanding resolution, special Commodore computer circuit.

Lower Price **$259**

PRINTERS

EPSON

RX-80, RX-80 FT.	$ CALL
FX-80, FX-100	$ CALL

OKIDATA

92	$445

STAR

Gemini 10	$299
Gemini 15	$449

C. ITOH

Gorrilla	$209
Prowriter	$360

SMITH CORONA

TP-2	$468
Cardco Printer Interface	$57
Tymac the Connection	$99

CARDCO

6 Slot Expander Interface	$72
3 Slot Expander Interface	$31

FLOPPY DISK

Elephant SS/DD (10)	$25
Elephant SS/DD (10)	$18
Verbatim SS/DD (10)	$Call

WORD PROCESSING

Quick Brown Fox	$48
Easy Script	$38
Word Pro 64	$64

PROGRAMMING SERIES

Assembler 64	$15
Logo	$48
Pilot	$38
Simon Basic	$19
Screen Editor	$19
CPM	$56

EDUCATION

English I-VII	$9
Math I-VIII	$9
Computer Science	$9
Science I-IV	$9
Technology	$9
History	$9
Business	$9
Geography	$9

GAMES

Avenger	$15
Wizard Wor	$19
Jupiter Lander	$15
Pinball Spect	$19
Lemans	$15
Radar Rat Race	$15
Clowns	$19
Gorf	$19
Omega Race	$19
Sea Wolf	$19

INFOCOM

Zork I, II, III	$27 ea.
Suspended	$29
Starcross	$29
Deadline	$29

COMPUTER COVERS

Features heavy duty canvas with vinyl interior - waterproof.

Reg. $15.95 **$6.99**

CMD 64	$6.99
VIC 20	$6.99
Disk Drive	$6.99
Espon MX 80	$7.99
Espon MX 80 FT	$7.99
Okidata 92	$7.99

BOOKS

Kids and the VIC	$18
Programmer's Reference Guide-VIC	$14
Programmer's Reference Guide-64	$18

COMPUTE

1st Book of CMD 64	$12
1st Book of Sound & Graphics	$12
1st Book of 64 Games	$12
Reference Guide to 64 Graphics	$12
Arcade Games on the 64	$12
1st Book of VIC	$12
2nd Book of VIC	$12
VIC Games	$12
Machine Language for Beginners	$14

Ordering & Payment Policy

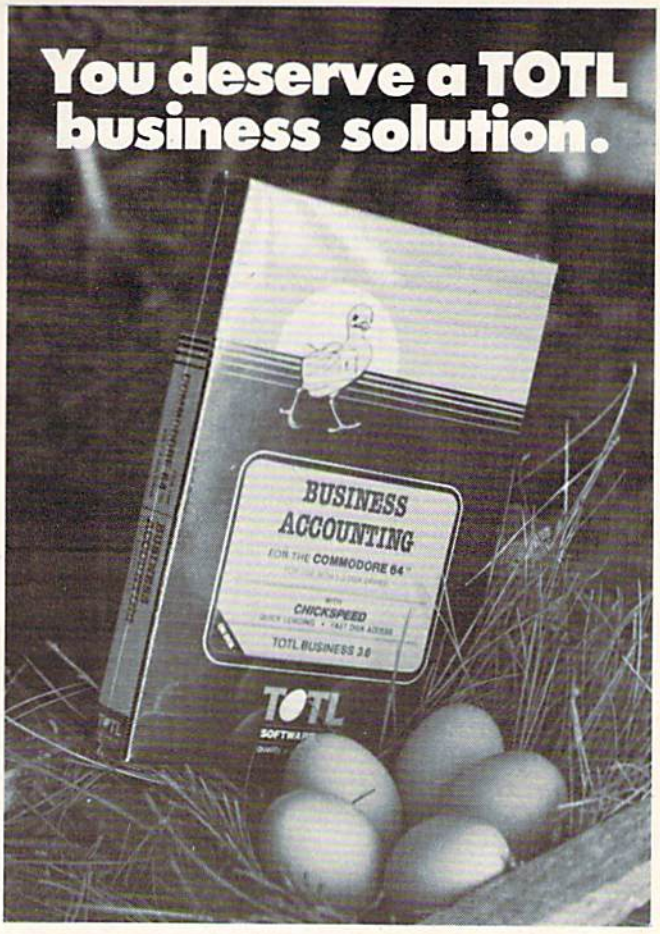
Prices reflect a cash discount. For C.O.D., Visa, and Mastercard add 3% Immediate delivery with certified check or wired funds. N.J. resident add 6%. Prices subject to change.

Shipping

For shipping and handling add 3%. ($3 minimum) Larger shipments require additional charge.

Catalog

We sell a large selection of hardware and software. Send $1 for catalog, refundable with order.



WORD PROCESSING
TOTL.TEXT

MAILING LIST AND LABELS
TOTL.LABEL

TIME MANAGEMENT
TOTL TIME MANAGER

KEYWORD CROSS REFERENCE
RESEARCH ASSISTANT

For Commodore 64™ and VIC 20™

Announcing the newest members of the family...

BUSINESS ACCOUNTING FOR **$95** (SUG. RETAIL)
TOTL.BUSINESS

SPELLING CHECKER FOR **$35** (SUG. RETAIL)
TOTL.SPELLER (64 only)

DATABASE MANAGEMENT FOR **$50** (SUG. RETAIL)
TOTL.INFOBASE

TOTL SOFTWARE, INC.
quality you can afford

Ask your dealer about TOTL Software or send in the coupon for further details and ordering information.

1555 Third Avenue, Walnut Creek, CA 94596
PLEASE SEND ME MORE INFORMATION ON TOTL SOFTWARE

Name: _____

Address: _____

_____ Zip _____

eight, but are invisible because they are drawn in the screen color.)

The reason for this is quite interesting. The program itself required its own custom character; the problem was where to put the data for this character. I knew the character set was 2048 bytes long (256 characters × 8 bytes per character), but I had to figure out where it was located. The first 512 bytes are for the 64 characters in the program stored in locations 7168 to 7679. The last 1024 bytes wrap around to Read Only Memory (ROM) where the normal characters (nonreversed) are stored. This is actually helpful, for if you replace the letters in the first 64 characters and want to PRINT a message, using CTRL-REV will PRINT normal characters. This is also why the cursor does not blink, because what should be the reversed space is actually the standard, nonreversed space.

Now, we're left with 512 bytes unaccounted for. We can find them at the next 512 bytes of memory after 7679, or in other words, screen memory. But the screen takes only 506 bytes (22 rows × 23 columns), so there are six bytes free. Six bytes are not enough to make a character; it takes eight. And that is why those two mystery characters appear on the screen—they are part of a custom character used in the program, with the rest hidden in six bytes of unused screen memory.

If you are wondering what character I went through so much trouble to squeeze in, it is the circle character which represents a bit turned "on." I could not use the normal circle character because if the auxiliary color value were greater than seven, it would appear in multicolor mode looking very strange—certainly not user-friendly.

Typing It In

When typing this program, be careful. First of all, it works only on an unexpanded VIC, so if you have any memory expansion, remove or disable it. Next, remember the program takes virtually all memory available to BASIC. Use absolutely no spaces except those inside quotes. Also, a number of program lines exceed the maximum 80 characters. To type these lines, you must use the abbreviations listed on page 133 of your VIC manual, or the computer simply will not accept the long line.

If you want to save the trouble of typing in the program, send a stamped self-addressed envelope, a blank tape, and $3 to the address below, and I will make you a copy.

Bill Gates
1330 Lomay Place
Pasadena, CA 91103

See program listing on page 184.

MACHINE LANGUAGE FOR BEGINNERS

RICHARD MANSFIELD, SENIOR EDITOR

Tapping Into BASIC

It's all inside there, sparkling like a castle of crystal, waiting for you to say the magic word. BASIC is a collection of machine language (ML) programs. Sometimes, jumping into these prewritten routines is the best solution to an ML problem.

For example, there's one routine which prints a character to the screen. If you JSR 65490, whatever character is held in the accumulator will appear on the screen. Try it from BASIC: POKE 5000,169: POKE 5001,65: POKE 5002,96: SYS 5000: SYS 65490. The 169 is LDA (LoaD the Accumulator), 65 is the letter A, 96 is RTS (just like BASIC's RETURN). So, we SYS to the little ML program we've POKEd in at 5000 and then SYS to the PRINT ML routine within BASIC's ROM collection. If you wanted to print a B, you could just POKE 5001,66.

From ML, you can directly LDA with the character of your choice and then JSR 65490. This is one of the more important BASIC routines to tap into when programming in ML. Why bother? Why not just LDA and then STA (STore the Accumulator) to a known screen RAM address? You could, but you'd then have to keep track of where the cursor is, where each letter is going, and you couldn't send useful formatting and control characters like a carriage return. 65490 is a perfect way to send messages to screen from within ML. And it's much faster than BASIC's PRINT command. Nothing needs to be looked up or interpreted—you're sending control right to the ML within BASIC itself.

The easiest way to print messages on screen in ML is to set aside an area of memory as a message zone. For instance, we could decide to store all our messages between addresses 1000 and 1500. Then, knowing the starting address of each message, we could print any of them easily. Assume that you need to print the words FINAL SCORE at the top left of the screen:

1. When writing your ML program, store 19 (cursor to home position), 70 (the letter F), 73 (I), 78 (N)...until the message is complete. (These letter codes are found as appendices to many books, and a complete table was published in COMPUTE!, November 1983, p. 251.)

2. If you choose to store this message at address 1040, then you can write a short ML routine to print it out, incrementing the Y register from 0 until it equals the total number of characters in the message (12, in this case):

```
5000 LDY #0
5002 LDA 1040,Y (This is the indexed addressing we
                 talked about last month.)
5005 JSR 65490
5008 INY
5009 CPY #12
5011 BNE 5002
```

Something Accidental

What about RND? That's going to be needed in ML games quite often, and it's certainly not going to be easy to duplicate the twisted, lengthy method required to get something accidental out of these relentlessly logical machines. How would you get random numbers in ML? At first you might think you could just PEEK (LDA) the jiffy clock location in zero page (the first 256 memory cells), which is location 162 and which is very rapidly flipping numbers around. From BASIC, try:

```
10 ? PEEK (162);: GOTO 10
```

This is the internal clock, updating itself. Unfortunately, it's not like a clock hanging on the wall. If it were, you could glance at the second hand from time to time and get a fairly random series of numbers that way. The computer's clock governs the timing of events (including BASIC and ML) within the computer—if you regularly call upon the clock for random numbers, you'll be disappointed. Such calls will be controlled by the clock itself.

This is a good chance to use the built-in RND

function. To see how, let's follow the logic of Program 1, the 64 version of this month's addition to our all-ML game. So far, we've filled the color RAM and drawn a frame around the screen. Now we'll put 20 game characters on the top of the screen, in a random pattern. This way, the game will be different each time it's played.

First, we LDA with the number (#) 20 and store this count into address 204 (it's OK to use address 204; it's used by BASIC for the flashing cursor and so will remain unused during ML execution). 204 will hold the number of times we go through the loop, placing characters randomly on screen.

Then in line 29228 we JSR (Jump to Sub-Routine) at 57502 which is the entrance to RND within BASIC's ROM memory. The ML instructions at this address go through a complicated process designed to come up with an unpredictable number. That number is then stored in a zero page location called the Floating Point Accumulator #1, found in addresses 97–102. Non-integer numbers (ones with decimal points) are stored here in a crushed format: exponent, mantissa, and sign. We needn't worry about that, though. We can simply rely on the fact that after this JSR into RND, a random number between 128 and 255 will appear in address 98, ready for us to pick it up and use it however we wish.

Next, we load the Y register with the random number and, using Y as an offset, we check to see if our intended location is already used—that is, if it is part of our screen border. We LDA 936,Y and CMP (compare) it to the border character (224) to see if this location does, in fact, make up part of the border. We don't want to POKE (STA) into the border, so if the CMP is true then the BEQ (Branch if EQual) will take effect and send us back to try for another random number (BEQ 49228, our JSR into RND).

If there is no border there, however, we can LDA with the new character's code (90) and go ahead and store it on screen (STA 936,Y) at the random location. Then we DEC (DECrement, lower by one) the number we stored in address 204 which is acting as a counter for our loop. If it's not yet zero (BNE means Branch if Not Equal to zero), we loop back and JSR into RND once again. This will happen until we've been through the cycle 20 times and address 204 has been DECed down to zero.

Where did we get the 936 in LDA and STA 936,Y? Recall that the RND function is only going to give us numbers between 128 and 255. So, to avoid 128 blank spaces (where no character will ever be printed), we can't use the start of screen RAM (1024) as our initial target address. What's more, we don't want to put anything on the very top line of the screen. That would cover up our border. To decide where the first random character should potentially appear, we must subtract 128 and add the length of the top line, 40. 1024-128+40=936.

Mangled Registers

There are hundreds of frozen ML routines at your disposal with the BASIC language. In practice, you'll probably want to become familiar with a dozen or so—things like RND and disk and tape communications are far easier to accomplish if you don't have to write the ML from scratch. We'll be introducing the key routines in future columns. With each one, you'll need to make notes about what preconditions these routines expect and what effects they might have on your three registers: X, Y, and A (the accumulator). You'll be using one of these registers in nearly every ML instruction you write. Therefore, if you JSR to a BASIC routine which will affect one of them, you'll want to be aware of it.

The PRINT routine expects something in the accumulator as a precondition. It will print what it finds in A, but it has no effect on X, Y, or A. After PRINT does its work for you, it will RTS back to your ML and these registers will have been left intact.

RND, on the other hand, has no preconditions, but it does leave your registers mangled. That's why we couldn't use the X register as our counter and simply DEX down to zero. We had to set up that special register of our own at address 204.

It's easy enough to set up tests of these BASIC routines—just JSR and then look at the registers (you could, for example, LDX #1, LDY #1, LDA #1, JSR 65490, STA 828, STY 829, STX 830) and then look at addresses 828–830 to see if anything had happened to the numbers. You'll find maps of the start of ML routines in BASIC in books and in back issues of COMPUTE!. Here are a few of the more useful ones to explore: 65487—INPUT, 65490—PRINT, 65505—Check STOP key, 65508—GET, 50292—VIC's Warm Start of BASIC (control goes back to BASIC; 42100 for the 64).

Program 1: 64 Version

```
49224   LDA    # 20
49226   STA    204
49228   JSR    57502
49231   LDY    98
49233   LDA    936 ,Y
49236   CMP    # 224
49238   BEQ          49228
49240   LDA    # 90
49242   STA    936 ,Y
49245   DEC    204
49247   BNE          49228
49249   RTS
```

See program listings on page 190.

TELSTAR 64

Sophisticated Terminal Communications Cartridge for the 64.

PFO 10D 00D CP D1 D2 BELL 12:30:00 10:14:36
(TELSTAR's Status Line)

Don't settle for less than the best!

- Upload/Download to/from disk or tape.
- Automatic File Translation.
- Communicates in Industry Standard ASCII.
- Real-Time Clock plus Alarm Clock.
- Line editing capability allows correcting and resending long command lines.
- 9 Quick Read functions.
- Menu-driven.
- Similar to our famous STCP Terminal package.
- Works with Commodore Modems and supports auto-dialing.

The best feature is the price — **only $49.95** (Cartridge and Manual)

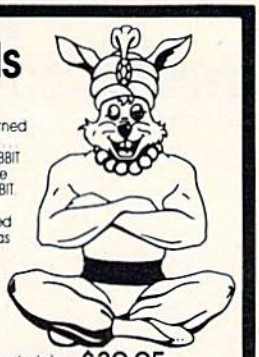
Machine Language Monitor Cartridge
for the CBM 64

More than 20 commands allow you to access the CBM 64's Microprocessors Registers and Memory Contents. Commands include assemble, disassemble, registers, memory, transfer, compare, plus many more.

Someday every CBM 64 owner will need a monitor such as this.

Cartridge and Manual — **$24.95**

8K in 30 Seconds
for your VIC 20 or CBM 64



If you own a VIC 20 or a CBM 64 and have been concerned about the high cost of a disk to store your programs on ... worry yourself no longer. Now there's the RABBIT. The RABBIT comes in a cartridge, and at a much, much lower price than the average disk. And speed ... this is one fast RABBIT. With the RABBIT you can load and store on your CBM datasette an 8K program in almost 30 seconds, compared to the current 3 minutes of a VIC 20 or CBM 64, almost as fast as the 1541 disk drive.

The RABBIT is easy to install, allows one to Append Basic Programs, works with or without Expansion Memory, and provides two data file modes. The RABBIT is not only fast but reliable.

(The Rabbit for the VIC 20 contains an expansion connector so you can simultaneously use your memory board, etc.)

$39.95

MAE NOW THE BEST FOR LESS!

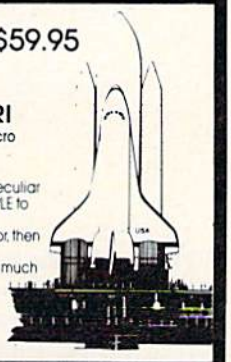
$59.95

For CBM 64, PET, APPLE, and ATARI

Now, you can have the same professionally designed Macro Assembler/Editor as used on Space Shuttle projects.

- Designed to improve Programmer Productivity.
- Similar syntax and commands – No need to relearn peculiar syntaxes and commands when you go from PET to APPLE to ATARI.
- Coresident Assembler/Editor – No need to load the Editor, then the Assembler, then the Editor, etc.
- Also includes Word Processor, Relocating Loader, and much more.
- Powerful Editor, Macros, Conditional and Interactive Assembly, and Auto – zero page addressing.

Still not convinced; send for our free spec sheet!



Eastern House

3239 Linda Dr.
Winston-Salem, N.C. 27106
(919) 924-2889 (919) 748-8446
Send for free catalog!

VISA* MasterCard

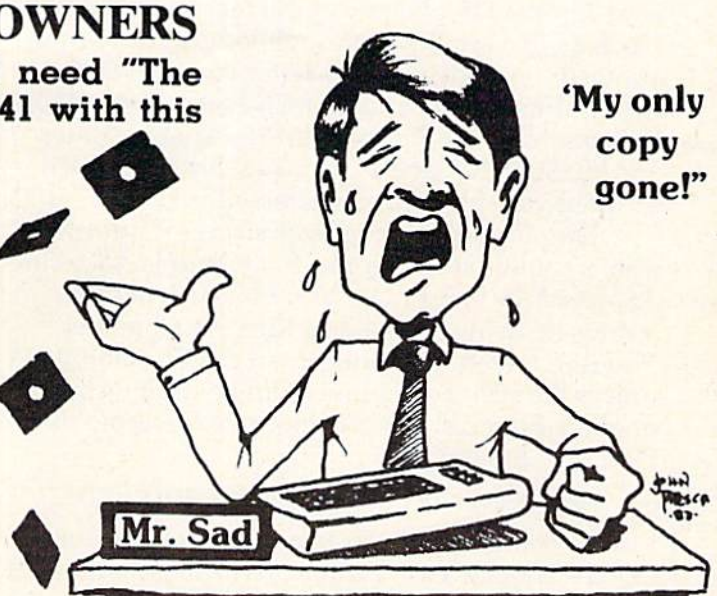
We'll back you up!

MasterCard VISA

ATTENTION COMMODORE 64 OWNERS

If you own a disk drive then you'll need "The Clone Machine". Take control of your 1541 with this package that includes:

1.) Complete and thorough users manual
2.) Copy with one or two drives
3.) Investigate and back-up many "PROTECTED" disks
4.) Copy all file types including relative types
5.) Edit and view track/block in Hex or ASCII
6.) Display full contents of directory and print
7.) Change program names, add, delete files with single keystroke
8.) Easy disk initialization
9.) Supports up to four drives



~~List $49.95~~ **Special intro $39.95**

Dealers & Distributors Inquiries Invited

CALL (201) 838-9027

MICRO WARE

P.O. Box 113
Pompton Plains, N.J.
07444

C. REGENA

String Variables And Functions

There are essentially two kinds of values you work with on the computer: *numbers* and *strings*. Last month's column was about numeric functions. This month we'll look at strings.

A string may be a constant or a variable and may consist of letters, numbers, and symbols. In general, the BASIC language has good string-handling capabilities. You don't have to convert the names of things to numbers, so you don't have to be a mathematician to program your computer to do such things as making lists and rosters or alphabetizing titles.

Commodore BASIC Strings

In Commodore BASIC, string variable names must end with the dollar sign. The variable name may be one or two letters followed by the dollar sign and may not be one of the reserved words (such as GO or OR). Although longer names are allowed, only the first two letters will be recognized, thus BLUE$ and BLACK$ are the same to the computer (BL$). Valid string variable names are A$, SC$, N$, and N3$. String arrays or subscripted variables are also allowed.

The computer recognizes strings if information is contained in quotes. For example, to define A$, use A$="HELLO". You may also define strings by using DATA and READ statements. You don't need to use quotes in DATA statements unless the string contains leading or trailing spaces or embedded commas or colons. An acceptable DATA statement is

```
60 DATA HELLO,"ANN ",THIS IS "JOHN, ED"
```

To combine strings, use the plus sign, such as C$=A$+B$+"." Strings may not be combined with numbers.

Strings may be compared using relational operators just as with numbers, =, <, >, <=, >=, and <>. They are compared by the ASCII character code values of each letter from left to right. If N1$="CINDY" and N2$="CHERY", then N2$ is less than N1$. The first letters are the same, but the second letters are compared and H is "less than" I because the ASCII code for H (72) is less than the ASCII code for I (73). You can use this principle to alphabetize lists.

Commodore BASIC has quite a few built-in string functions. ASC(*s*) returns the ASCII character code value for a string *s*. If the string contains several characters, only the ASCII code for the first character is returned. Valid statements are:

```
10 PRINT ASC("H")
10 E=ASC(N$)
10 IF ASC(A$)=32 THEN 70
```

The following sample program gives the ASCII value of keys that you press on the keyboard.

```
100 PRINT "PRESS A KEY."          :rem 71
110 GET A$                        :rem 215
120 IF A$="" THEN 110             :rem 201
130 A=ASC(A$)                     :rem 159
140 PRINT A$,A                    :rem 244
150 GOTO 110                      :rem 97
160 END                           :rem 110
```

CHR$(*n*) returns the ASCII character corresponding to the number or numeric expression *n*. The number must be from 0 to 255. Some of the characters are actually control characters, such as ones that change the cursor to yellow or shift to lowercase. The following sample program illustrates the use of CHR$. You need to enter a number, then the computer will print the corresponding ASCII character.

```
100 PRINT                         :rem 30
110 PRINT "ENTER A NUMBER"        :rem 235
120 PRINT "FROM 33 TO 127."       :rem 105
130 INPUT N                       :rem 114
140 IF N<33 THEN 100              :rem 212
150 IF N>127 THEN 100             :rem 11
```

80 COLUMNS! 25 LINES

A FULL PROFESSIONAL DISPLAY FOR

Commodore 64

with

Screenmaker™

Screenmaker is a video display generator module that plugs into the expansion connector of the Commodore 64.

- ✓ WORD PROCESSING
- ✓ CALCULATIONS
- ✓ BASIC PROGRAMS

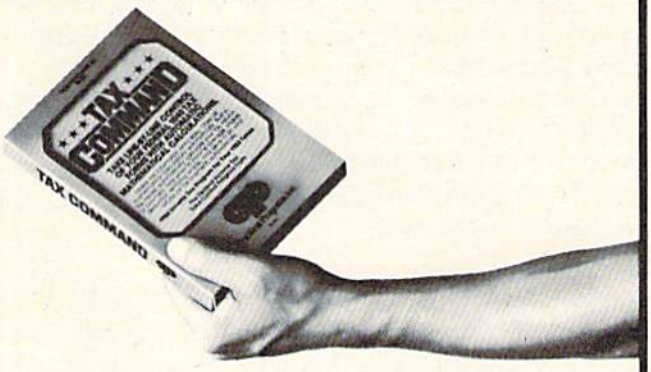
Screenmaker provides a B & W video signal that connects to your video monitor to provide a full 80 characters on each line. With Screenmaker, Screen displays will appear the same as the printer output. Trial printouts can be eliminated. Word processing is easier. Forms and reports can be set up faster. Screenmaker features a bank switched memory, 40/80 video switch, and a full character set including graphics.

SCREENMAKER $159.95
Copy-Writer Word Processor... $ 79.95
SCREENMAKER/
Copy-Writer Package $199.95

MICROTECH
P.O. Box 102
Langhorne, Pa. 19047
215-757-0284

TAX COMMAND

NOW YOUR COMMODORE OR VIC PUTS LINE-BY-LINE CONTROL OF TAX PREPARATION AT YOUR FINGER TIPS.



Calculations are automatic. All you do is enter your tax information. Tax Command does all mathematical calculations for you. **Built-in tax tables eliminate guesswork.** No more finding the right column down and right line across. Tax Command has the 1040 tax tables built right in. So it zeros in on your refund (or tax payment) amount automatically. **Tax Command is fast, easy!** Just fill in the blanks. If you make a mistake, no problem. Just go back and retype your entries. Everything is re-calculated for you. You'll find the 1040. Schedule A. Capitol Gains & Losses. Income Averaging. And more. Anyone who can read can use Tax Command. And the best feature of all . . . just $24.95 plus $2.00 for shipping and handling. Wisconsin residents add 5% tax.

Practical Programs, Inc.

P.O. Box 93104-L • Milwaukee, WI 53203 • (414) 278-0829
Available at fine computer stores everywhere, or by ordering direct.

SOPHISTICATED SOFTWARE OF AMERICA™
PRESENTS

GRAFIX – ARTIST™
(Commodore 64™ version)

THE LATEST IN EDUCATIONAL GRAPHICS SOFTWARE DESIGNED WITH THE CONSUMER IN MIND

CREATE EXTRAORDINARY COLOR – GRAPHICS USING THE:

- Joystick Mode
- Program Mode
- Program to Picture Utility (for your basic or machine-language programs)

GRAFIX–ARTIST™ provides comprehensive

- Reference Card
- Help Screens
- Introduction Tutorial
- Demo's

NO COMPUTER EXPERIENCE IS NECESSARY

Children, parents, artists, educators will enjoy the ease-of-use and options GRAFIX–ARTIST™ provides.

— COMING SOON —

GRAFIX–PRINTER™ and GRAFIX–DESIGNER™

Dealer and Distributor Inquiries Invited

198 Ross Rd.
King of Prussia, PA 19406
(215) 265-2277

*Registered trademarks of Commodore Business Machines, Inc.

HUNDREDS OF PROGRAMS AVAILABLE FOR THE COMMODORE 64 & VIC 20

Commodore 64

All Prices up to 40% OFF RETAIL

VIC 20

HOME/BUSINESS	
Practicalc (D)	41.50
Practicalc (CASS)	37.00
Bank St. Writer (D)	52.50
Smart 64 Terminal	31.95
Quick Brown Fox (CT)	42.95
Paper Clip (D)	93.95
Word Pro/SpellRight (D)	72.95
CalcResult Advanced (D)	112.50
Data Manager (D/CASS)	18.75
Home Acct (Continental) (D)	49.95

EDUCATIONAL	
Facemaker (D/CT)	22.95
Kindercomp (D/CT)	19.95
Delta Drawing (CT)	26.95
Type Attack (D)	28.95
Early Games (D/CASS)	22.50
Electronic Party (VIC 20/CASS)	22.50
Square Pairs (VIC 20/CASS)	22.50
Turtle Trax (CIC 20/CASS)	22.50
Alphabet Zoo (CT)	22.95
Koala Touch Tablet	72.95
Mastertype	28.95

GAMES	
Choplifter (CT)	28.95
Lode Runner (D)	25.95
Enchanter (D)	37.50
Jumpman (D/CASS)	27.95
Beach Head (D)	26.95
Neutral Zone (D/CASS)	26.95
Temple of Apshai (D/CASS)	27.95

HARDWARE AND ACCESSORIES	
Cardco Printer Interface	62.95
Cardprint G	71.95
The Connection Parallel Int.	89.95
Cardco 3 Slot Exp. Board	31.95
Cardco 5 Slot Exp. (C-64)	55.95
Data 20 Video Pak 80 (C-64)	143.95
Data 20 8K Display Mgr. (40/80 col. plus Word Proc.)	119.95
0 K Display Mgr (40/80)	79.95
Zenith 12" Green Monitor	105.00
Brother/Dynax DX-15 (Letter Quality)	485.00
C Itoh Prowriter 8510AP	375.00
Gemini 10X	299.00

Write or call for FREE CATALOG. TO ORDER: CALL 1-714-643-1056
8:00 A.M.-6:00 P.M. PST Mon.-Sat. or send check or credit card number, signature and expiration date. Please include phone number.

CENTURY MICRO PRODUCTS
P.O. Box 2520, Mission Viejo, CA 92690

Visa/Mastercard add 3%. Personal checks allow 2 weeks to clear. CA residents add sales tax. Shipping and handling add $3.00 (hardware extra). Prices subject to change.

```
160 PRINT CHR$(N)                    :rem 196
170 GOTO 130                         :rem 101
180 END                              :rem 112
```

Screen Formatting

When you print on the screen, you can format your printing to make output easier to read. The TAB function works just like the tabulator on a typewriter. The columns on the screen are numbered from 0 to 21 for the VIC-20 and 0 to 39 for the Commodore 64. If you use TAB(*c*) the printing will start in the column *c* you specify. The number *c* may be a constant, variable, or numeric expression. This sample program illustrates printing with the TAB function.

```
100 PRINT"{CLR}"                     :rem 245
110 PRINT"0123456789012345678 9"     :rem 125
120 PRINT TAB(5);"HELLO"             :rem 112
130 PRINT TAB(9);"THIS IS 9"          :rem 14
140 X=5:Y=2                           :rem 97
150 PRINT TAB(X-Y);"TEST"            :rem 232
160 FOR T=0 TO 12                     :rem 69
170 PRINT TAB(T);"GAZETTE"            :rem 52
180 NEXT T                            :rem 44
190 END                              :rem 113
```

Remember that you can use semicolons and commas to separate items in your printing. The semicolon puts two strings right next to each other. The comma starts the printing of the next item in the next print zone.

Another handy function to help you in printing is SPC(*n*), which will print *n* number of blank spaces between items. An example of the format is:

```
PRINT "ABC";SPC(6);"DEF"
```

The semicolons are optional. The following sample program illustrates several ways SPC can be used.

```
100 PRINT "HELLO";SPC(5);"READER":rem 175
110 X=3:Y=4                           :rem 94
120 PRINT "A";SPC(X+Y);"B"           :rem 180
130 FOR I=1 TO 5                      :rem 10
140 PRINT "C";SPC(I);"D"              :rem 39
150 NEXT I                            :rem 30
160 END                              :rem 110
```

You cannot combine strings with numbers or compare strings to numbers. However, there are times you will want to work with strings and numbers combined, such as names and scores. You can convert the number to a string using STR$(*n*). For example:

```
10 S=4                                :rem 37
20 N$="RANDY"                        :rem 211
30 S$=STR$(S)                        :rem 216
40 A$=N$+", "+S$                     :rem 181
50 PRINT A$                           :rem 87
```

Line 40 combines the name N$ with a comma and a space, then the number 4 which has been converted to a string.

Conversely, if you want to get back to numbers from strings for calculations, you can use VAL(*s*). This function will return the numeric value of a string *s*, and the string must contain numbers. For example, you can use a command such as N = VAL(S$).

LEN(*s*) is a string function that returns the LENgth of a string, or the number of characters in a string. For example, LEN("HELLO") is equal to the number of letters, 5. The following sample program lets the computer figure out the length of the various titles to be centered. Be sure to use the appropriate value in line 200 (11 for the VIC, and 20 for the 64).

```
100 PRINT "{CLR}"                    :rem 245
110 T$(1)="SAMPLE TITLE"              :rem 81
120 T$(2)="CENTERING"                :rem 174
130 T$(3)="BY"                       :rem 172
140 T$(4)="AUTHOR"                   :rem 230
150 FOR C=1 TO 4                       :rem 5
160 PRINT "{DOWN}"                   :rem 121
170 L=LEN(T$(C))                      :rem 93
180 REM FOR VIC LET S=11              :rem 29
190 REM FOR C64 LET S=20             :rem 233
200 S=11                             :rem 132
210 PRINT TAB(S-L/2);T$(C)           :rem 188
220 NEXT C                            :rem 22
230 END                              :rem 108
```

Line 100 clears the screen. Lines 110–140 define strings to be printed later. Line 170 calculates the length of the title, then line 210 tabulates an amount depending on the length to center the title.

Dividing Strings

If you have a string, you can look at parts of the string or segments of the whole string by using the functions MID$, LEFT$, and RIGHT$. The MID$ format is MID$(*s*,*f*,*n*) where *s* is the string expressed either in quotes or as a string variable name, *f* is the position you want to start the segment, and *n* is how many characters you want in the segment. For example, PRINT MID$("CHAIRMAN",2,4) looks at the string "CHAIRMAN" and prints the segment starting with the second letter and using four letters. The result printed is HAIR.

The following program uses the MID$ function to print a title and move it across the screen like you would see in an electronic sign or a moving marquee. Put the appropriate value in line 240 if you are using a 64.

```
100 REM MARQUEE                      :rem 133
110 GOTO 500                          :rem 96
190 REM SUBROUTINE                   :rem 142
200 D=0:S$=""                        :rem 117
210 L=LEN(M$)                        :rem 189
220 REM FOR VIC C=21                  :rem 36
230 REM FOR C64 C=39                 :rem 249
240 C=21                             :rem 121
250 K=INT((C-L)/2)                   :rem 201
```

COMPUTER DISCOUNT

TOLL FREE 1-800-621-6131 FOR ORDERS

4251 W. Sahara Ave., Suite E Las Vegas, Nevada 89126

MONDAY THROUGH SATURDAY ● 9 AM TO 6 PM

Comm 64	$229
1541 Disk Drive	249
1525 Printer	229
1702 Color Mont.	259
Hes Mon	29
Paper Clip w/p	115
Calc Result	140
Sysres-Utility	90
Renaissance	30
Vic-20	90
Datasette	64
1600 Modem	85
Word Processor	95
1311 Joystick	8
1312 Paddles	16
1210 3k Expander	34

SPECIAL

Comm 64 ..... $785
1541 Disk Drive
152EE Printer
1600 Phone Modem

HARDWARE

C. Itoh Prowriter	$379
Nec. 8023A	429
Banana Printer	209
Hayes 1200 Modem	489

COMM. 64 DISK

Temple Of Apshai	$33
Upper Reaches Apshai	18
Jump Man	33
Zork I	33
Zork II	33
Zork III	33
Frogger	30
Jawbreaker	24
Ft. Apocalyse	30
Pharoh's Curse	30
Starter Pack D/C	$22
Word Machine	17
Pet Emulator	17
Gen Ledger	69
Mail List Mgr.	43
Hes 6502	23

ATARI

New Educational Programs
Purchases can be made by check, money order, C.O.D.
Carte Blanche and Diners Club. 1-702-367-2215

The Banner Machine™

For the **Commodore 64** (4 extra fonts available). For the **VIC-20** with 24K memory (3 extra fonts available). • Use on any Gemini or Epson MX with Graftrax or the FX and RX printers. Also Commodore 1525E and Banana with the C-64. • Menu-driven program operates like a word processor. • Makes signs up to 13" tall by any length. • Makes borders of widths up to ¾". • 8 sizes of letters from ¼" to 8" high. • Proportional spacing; Automatic centering; Right and left justifying. • $49.95 Tape or Disk (Specify computer equipment)

For the Commodore 64:

Space Raider An amazing arcade simulation. Your mission is to destroy the enemy ships. $19.95

Super Roller Challenging dice game. Sprite graphics and sound. Yahtzee-style rules of play. $14.95

Microbroker Exciting, realistic and educational stock market simulation. $34.95 Tape or Disk

Preschool Educational Programs ABC Fun; 123 Fun; and Ginger the Cat with: Addition and Subtraction, Number Hunt, and Letter Hunt. All programs have bright color, music, and action. Each $14.95

Formulator A scientific calculator for tasks which require repetitive arithmetic computations. Save formulas and numeric expressions. $39.95

Sprite Editor The easy way to create, copy, alter, and save up to 224 sprite shapes. $24.95

Cross Reference Generator for BASIC programs Locates lines with BASIC words or variable names and allows changes, and more. $19.95

VIC-20 Programs Also Available. Ask for Catalog.

Cardinal Software™

Virginia Micro Systems
13646 Jeff Davis Highway
Woodbridge, Virginia 22191
Phone (703) 491-6502

Commodore 64 and VIC-20 are registered trademarks of Commodore Electronics Ltd.

COLOR PROBLEMS?

One of Our Four New Products will Solve Them!

You're not alone. Thousands of Commodore 64 owners have "fuzzy" color on their TV. Most have interference lines crowding out their great graphics. Many have bought expensive monitors or new TVs, and often even that hasn't helped. But, most of us just lived with the problem. Now the engineers at Bytes & Pieces have four simple, inexpensive solutions.

If you have an "old 64" (with the 5 pin Monitor Din Plug), you've probably had color, resolution and interference problems. We can solve them!

1. **The Interference Stopper**... A new kit that installs in minutes with two simple solder connections. Best results when combined with #2, 3, or 4 below. Absolutely stops 90% of the RF interference on your screen.

$15.95

2. **The Color Sharpener**... Use if your "old 64" is hooked up to a TV. Just plug into the monitor plug, and the color and contrast immediately improve. Dramatically. Crisp letters. Great graphics.

$18.95

3. **The NEW Color Sharpener CABLE**... Use if your "old 64" is hooked up to a monitor. A new 2 prong cable, with the Color Sharpener built in. All the benefits of #2, on your monitor.

$24.95

4. **The Monitor "Improver"**... If you have a Commodore 1701 monitor, this cable (3 prong) gives you a picture you won't believe. Better than the cable Commodore built... by a lot. Try it, you won't be disappointed. (Also hooks your "Old 64" to the 1702.)

$24.95

If any of our products do not work to your satisfaction, send it back and we'll refund your purchase price in full.

DUST PROBLEMS?

Solve Them with Matching Dust Covers for Computer, Tape and Disk. $7.95—$9.95

These are the deluxe covers for either the Commodore 64 or the Vic 20 made of brown leather grain Naugahyde, specially lined with a soft non-scratch liner, for a cover you just can't beat.

Don't waste your money on those cheap looking, clear plastic, static filled covers. Get the quality ones, custom fitted to your Commodore computers.

Available singly or as a matched set in beautiful brown simulated leather.

Commodore 64 and Vic 20 are registered trademarks of Commodore Computer Company.

Bytes & Pieces Dealer Inquiries Invited.
550 N. 68th Street, Wauwatosa, WI 53213 414/257-3562

ORDER TODAY!

Please send me the following:

Qty.	Item	Amount
____	Interference Stopper @$15.95	$ ____
____	Color Sharpener @$18.95	$ ____
____	NEW Color Sharpener Cable @ $24.95	$ ____
____	The Monitor Improver @ $24.95	$ ____
____	Computer Dust Cover @ $9.95	$ ____
____	1541 Disk Dust Cover @ $8.95	$ ____
____	Dataset Dust Cover @ $7.95	$ ____
	Shipping & Handling	$ 2.00
	5% State Tax (Wisconsin Residents only)	$ ____
	TOTAL	$ ____

☐ Check or Money Order enclosed
☐ Charge to my VISA or MasterCard
VISA # ____
MasterCard # ____
Inner Bank # ____
Expiration Date ____

Signature ____

SHIP TO:
Name ____
Address ____
City ____
State/Zip ____

```
260 FOR S=1 TO K:S$=S$+"{SHIFT-SPACE}":NE
    XT S                              :rem 110
270 FOR T=C TO K STEP -1              :rem 220
280 D=D+1                             :rem 187
290 PRINT TAB(T);MID$(M$,1,D);S$+"{2 UP}"
                                      :rem 175
300 NEXT T                             :rem 38
310 RETURN                            :rem 116
500 PRINT "{CLR}"                     :rem 249
510 M$="SAMPLE TITLE"                 :rem 204
520 GOSUB 200                         :rem 169
530 PRINT                              :rem 37
540 END                               :rem 112
```

Line 290 prints the segment of the title with the length increasing by one for each loop. The TAB function decreases by one in each loop. The technique is written as a subroutine in lines 200–310 so you can use it in your own programs. Line 520 calls the subroutine. Try different messages in line 510 for M$. The message must be shorter than the number of columns in your screen (22 for the VIC, 40 for the 64).

LEFT$ and RIGHT$ are more specialized functions to work with segments of the string. LEFT$ (*s*,*n*) indicates to take the segment of string *s* starting with the leftmost character and containing *n* number of characters. LEFT$("CHAIRMAN",5) would start at the left of CHAIRMAN and use 5 letters to return the word CHAIR. RIGHT$(*s*,*n*) takes the right segment of the string *s* and containing *n* characters—the last character is the rightmost character of the original string. RIGHT$ ("CHAIRMAN",3) would be MAN.

The following program uses a subroutine containing LEFT$ and RIGHT$ functions to print a money value and line up the decimal places.

```
100 REM MONEY                         :rem 253
110 GOTO 500                           :rem 96
190 REM SUBROUTINE                    :rem 142
200 P$=STR$(P)                          :rem 1
210 IF LEN(P$)=2 THEN P$="0"+RIGHT$(P$,1)
                                      :rem 120
220 PR$=RIGHT$(P$,2)                   :rem 92
230 PL$=LEFT$(P$,LEN(P$)-2)           :rem 213
240 IF LEN(PL$)<2 THEN PL$="{2 SPACES}"
                                      :rem 243
250 P$="$"+PL$+"."+PR$                 :rem 41
260 RETURN                            :rem 120
500 PRINT "{CLR}"                     :rem 249
510 PRINT                              :rem 35
520 PRINT "ENTER A WHOLE NUMBER"      :rem 111
530 PRINT"FROM 0 TO 999"               :rem 27
540 INPUT P                           :rem 121
550 IF P<0 THEN 510                   :rem 170
560 IF P>999 THEN 510                  :rem 40
570 P$=STR$(P)                         :rem 11
580 FOR J=1 TO LEN(P$)                :rem 131
590 IF MID$(P$,J,1)="." THEN 510       :rem 55
600 NEXT J                             :rem 31
610 GOSUB 200                         :rem 169
620 PRINT TAB(10);P$                   :rem 93
630 GOTO 540                          :rem 107
640 END                               :rem 113
```

You can enter an amount of money expressed in the number of cents, a whole number from zero to 999. The value you input is P. Lines 570–600 make sure you have entered a whole number and not a decimal. The subroutine in lines 200–260 convert the price P to a money value to be printed as P$. If you were using the subroutine in your own program, you would be calculating values for P (or reading them in from DATA) and would not have to use lines 570–600 to check for a valid P.

The string-handling capabilities of your computer allow for great versatility in applications. Although a computer can be used as an improved calculator for arithmetic functions, the string functions allow programming and information handling in a variety of other uses.

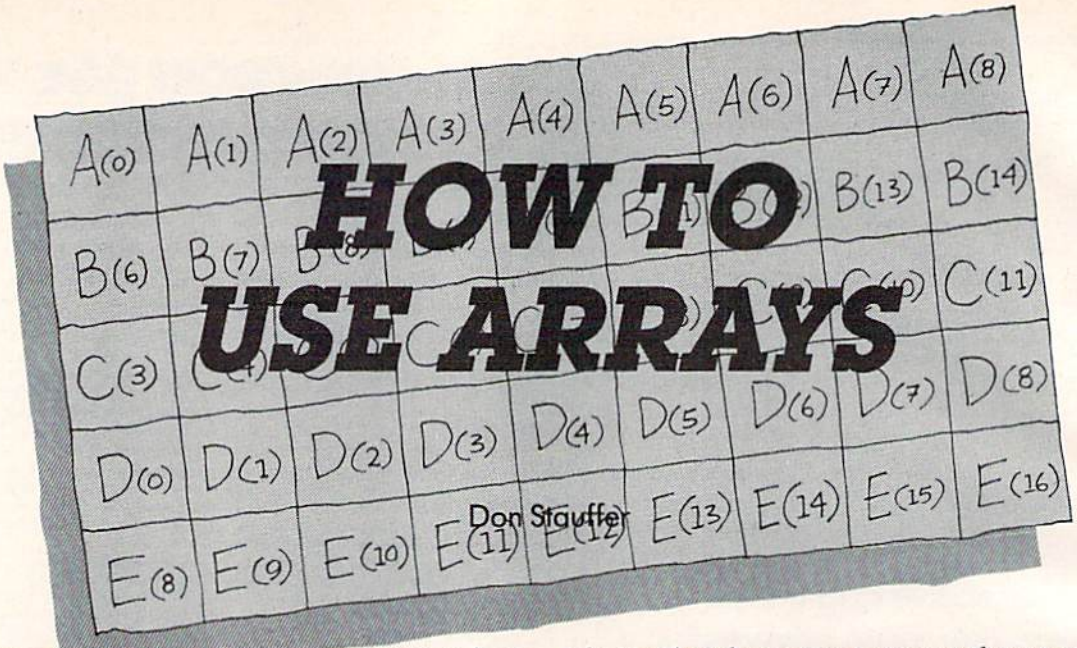
The final program this month offers a drill in identifying verbs. This program uses string arrays A$, B$, and C$. Lines 180–240 read in words to fill the string arrays—subjects, verbs, and adverbs. Lines 290–320 choose random numbers, then lines 340 and 360 print the test sentence. The user types the verb of the sentence and then RETURN. The quiz consists of 10 sentences, after which a score is printed. Lines 270, 400, and 470 illustrate how string comparisons can be used.

Verbs

```
100 REM IDENTIFYING VERBS             :rem 49
110 PRINT "{CLR}"                     :rem 246
120 PRINT TAB(6);"*********"          :rem 119
130 PRINT TAB(6);"* VERBS *"          :rem 212
140 PRINT TAB(6);"*********"          :rem 121
150 PRINT "{3 DOWN}GIVEN A SENTENCE,"
                                      :rem 213
160 PRINT "{DOWN}TYPE THE VERB"       :rem 203
170 PRINT "{DOWN}THEN PRESS RETURN."
                                       :rem 68
180 FOR C=0 TO 9                       :rem 12
190 READ A$(C),B$(C),C$(C)            :rem 252
200 NEXT C                             :rem 20
210 DATA THE CAT,RAN,DOWN,A DOG,JUMPED,AR
    OUND,BOBBY,MOVED,QUICKLY          :rem 147
220 DATA CINDY,JOGGED,AN HOUR,A BOY,CRAWL
    ED,FAST,A GIRL,SKIPPED,HOME        :rem 5
230 DATA JUSTIN,RODE,TO TOWN,KIM,HOPPED,S
    LOWLY,MY PAL,HIKED,THERE          :rem 236
240 DATA THEY,CLIMBED,HAPPILY          :rem 73
250 PRINT "{3 DOWN}PRESS RETURN TO START.
    "                                 :rem 103
260 GET E$:IF E$="" THEN 260           :rem 91
270 IF ASC(E$)<>13 THEN 260            :rem 94
280 FOR T=1 TO 10:PRINT "{CLR}"       :rem 229
290 A=INT(RND(0)*10)                   :rem 69
300 B=INT(RND(0)*10)                   :rem 62
310 C=INT(RND(0)*10)                   :rem 64
320 S=INT(RND(0)*2)                    :rem 34
330 ON S+1 GOTO 340,360               :rem 119
340 PRINT A$(A);" ";B$(B);" ";C$(C);"."
                                       :rem 48
350 GOTO 370                          :rem 107
360 PRINT C$(C);" ";A$(A);" ";B$(B);"."
                                       :rem 50
370 PRINT:V$=""                        :rem 92
380 INPUT "VERB =";V$                 :rem 144
390 PRINT                              :rem 41
```

```
400 IF V$=B$(B) THEN 430          :rem 153
410 PRINT "THE VERB IS ";B$(B)     :rem 70
420 GOTO 450                      :rem 104
430 PRINT "CORRECT!!"             :rem 188
440 SC=SC+1                        :rem 93
450 PRINT "{DOWN}PRESS RETURN."    :rem 22
460 GET E$:IF E$="" THEN 460       :rem 95
470 IF ASC(E$)<13 THEN 460         :rem 36
480 NEXT T                         :rem 47
490 PRINT "{CLR}"                   :rem 1
500 PRINT"VERBS"                  :rem 232
510 PRINT "{2 DOWN}YOUR SCORE IS":rem 240
520 PRINT SC;" RIGHT OUT OF 10.{2 DOWN}"
                                  :rem 245
530 END                           :rem 111
```

To receive additional information from advertisers in this issue, use the handy reader service cards in the back of the magazine.



HOW TO USE ARRAYS

Don Stauffer

Using arrays is a handy BASIC programming technique. This tutorial explains what they are and how to use them when programming on your VIC or 64.

Arrays, sometimes called subscripted variables, are an important feature of Microsoft BASIC, but there is little documentation on what they are and how to use them. This is particularly true of the VIC and 64.

Some time ago, a friend of mine, a new VIC owner, called with a programming problem. He was working on a program in which he needed to generate random numbers for a variable (R). However, he wanted ten different values for R and wanted to save them for later use in the program, in statements where he would use these R values in calculations. I told him that was a perfect spot to use an array. After he looked up arrays in all the reference books he had on the machine, he wasn't much better off than when he first called, so we spent a session going over arrays. It seemed to me that the best way to know how to use arrays was to start with the basics.

What Is An Array?

An array is a type of variable which can have a number of values at any one time. For instance, let's look at a variable, T, which might stand for the maximum temperature for a particular day. T(1) might be the temperature of day 1, T(2) the temperature of day 2, and so on. The number in the parentheses is called the *subscript*. In fact, arrays are sometimes called *subscripted variables*. Although the best way to understand arrays is through examples, which we'll get to shortly, we should first learn a little about how the computer stores and uses arrays.

Since an array is a set of several values, it obviously takes more memory than a normal variable. In fact, unless the computer knows how many values your variable will have, it does not really know how much memory to set aside for that variable. We tell this information to the computer with a DIMension statement:

```
DIM X(15),Y(20)
```

In this example, we told the program we were going to use two arrays, X and Y, and that X would have a maximum of *16* values, and Y would have a maximum of *21*. Notice that the number of values set up is always one greater than the number specified in the DIM statement. Although it's confusing, this is because the computer starts counting with 0, not 1. To avoid confusion, some programmers simply ignore the 0 and treat X(15) as an array of 15 values. This wastes a tiny amount of memory, but it usually doesn't matter.

With the VIC and 64, the DIMension statement is optional unless you are going to use more than 11 values. I recommend, however, that you always DIMension arrays, even if they will have less than 11 values. It is good programming practice, and it will save considerable memory since the computer will not set aside unnecessary memory space. Also, the DIM statement initially sets all array values to zero. Good programming practice dictates that the array should be DIMensioned in one of the first statements of the program, and it obviously must occur before any reference to the array. The DIM statement must not be executed more than once, however, or an error results.

The particular value of an array is called the subscript, which is why the array is sometimes called a subscripted variable. In the following statement:

```
LET X(5)=27.3
```

subscript 5 of the X array is set to 27.3. Whenever

VIC 20™/COMMODORE 64™

CRAZY CONVEYORS™-combines the powerful capabilities of the Commodore 64 with disk drive in an exciting action-packed game with multi-color sprites, custom characters in 11 different colors for building blocks, ladders, fire poles, rotating pulleys, moving conveyors and bonus boxes; **three part harmony music**; high score history, with full names of 10 champions; action pause; start play at screen of your choice; joystick or keyboard; machine language. Also **Screen Creator™** to expand game disk and extra disks to virtually unlimited screens, and **CRAZY CONVEYOR action** to entertain and challenge the most skillful game player. Price: $29.95.

RIDGE RUNNER-for unexpanded VIC 20 on tape or disk. 100% machine language. Includes multicolor U.F.O., blinking mines, spinning asteroids, enemy ships, laser fire, horizontally scrolling playfield, hi-resolution/multicolor graphics, excellent sound, high score, pause button, bonus ships and ever increasing levels of difficulty. Joystick required. Price: $14.95.

DUNGEONS-for VIC 20 with 16K expansion and tape or disk. Explore a 12 level dungeon with 1200 rooms. Purchase weapon and armor, find treasures, battle over fifty types of monsters, cast spells and save game to tape or disk. Excellent sound and three dimensional graphics. Price: $14.95.

PAK ALIEN-for unexpanded VIC 20 with tape or disk. 100% machine language. Includes seven evil aliens, bonus timer, pause feature and 100 levels of increasing difficulty. Joystick or keyboard. Price: $14.95.

INVESTMENT PORTFOLIO MANAGER-for Commodore 64 with disk drive or tape (printer optional), is menu driven and provides one summary page and nine detail pages. Each page can accept nine entries of up to $99,999 each. The program can handle over $8 million. The IPM is quick and makes it easy to track volatile assets such as stocks and stock options. The summary page displays the grand total and the percent of grand total for each of nine investment categories. Price: $14.95.

DISK DIRECTORY MANAGER-for Commodore 64 or VIC 20 (16K min. exp.) with 1540/41 disk drive and 1525 printer, 100% machine language. This handy utility reads directories of diskettes and sorts up to 1556 records on the Commodore 64. In most cases the sort is completed in only a few seconds. Each record contains file name, file size, file type and disk ID. The sorted master directory is sent to the printer. Price: $19.95.

SEND FOR FREE CATALOG!

BYTES and BITS
524 E. Canterbury Ln.
Phoenix, AZ 85022

(602) 942-1475
Please specify tape or disk
Check, money order or C.O.D.
Add $2.00 for postage & handling
Additional $3.00 for C.O.D.

VIC 20 & Commodore 64 are trademarks of Commodore Electronics Ltd
CRAZY CONVEYORS and Screen Creator are trademarks of BYTES and BITS

VIC-20 USERS!

Win your share of over $5000 in VIC-20 Hardware & Software in

THE NATIONAL VIC-20 USERS GROUP

★★★ 1984 VIC-20 SWEEPSTAKES ★★★

cardco, inc.
Amateur Accessories
page 6
MICRODIGITAL
NIBBLES & BITS, INC.
CHEATSHEET PRODUCTS
COMM-DATA COMPUTER HOUSE
Public Domain, Inc.
COM-CAP

- Open to ALL VIC-20 Users
- No Purchase Necessary
- No Entry Fee

WIN:

- Arcade Games
- Educational Programs
- Business Programs
- Stock Market Programs
- Sports Programs
- Children's Programs
- FREE NVUG Memberships
- Typing Tutor Programs
- Special VIC-20 Aids
- Memory Expansion
- Motherboards
- Word Processors

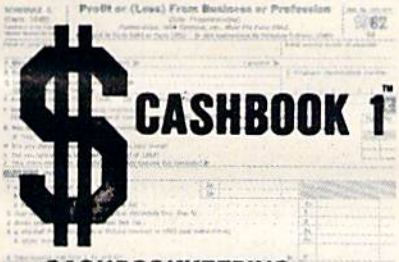
AND MORE!

TOTL
Universal Systems
ACADEMY SOFTWARE
TAYLORMADE SOFTWARE
BASIC BYTE, INC.
APROPOS TECHNOLOGY
Boston Educational Computing inc.
Sirius

The National VIC-20 Users Group and the Nation's Leading VIC-20 Dealers bring you the best thing to happen since you got your VIC!

WRITE TODAY FOR YOUR FREE OFFICIAL ENTRY BLANK, PRIZE LIST, AND COMPLETE RULES!

THE NATIONAL VIC-20 USERS GROUP
BOX 34575 DEPT. G OMAHA, NE 68134
(VIC-20 Reg. TM of Commodore)



$ CASHBOOK 1™

CASHBOOKKEEPING.....

for any small business with a Commodore 64™, & 1540/41 Disk Drive. (printer optional)

Hire Cashbook 1 as your new bookkeeper.

Send $65 or call toll free order desk 1-800-321-6927. 24-Hour Service

- Also -

The Computer Classroom™

Programs for VIC20™ and Commodore 64™ disk or tape. Write for listing. OR, try MOUSE TRAP MATH, for your VIC20™ (10.95 for tape) and we'll include listing.

ROCKY SOFTWARE
ROCKY SOFTWARE
Box 310
Erie, Colorado 80516
(303) 652-2183

NEW FROM HCP!

EXPANSION PRODUCTS FOR YOUR VIC - 20™

For You Who Care Enough To Use The Best

24K RAM	RS-232 CARD	MOTHER BOARD
• Single board for all RAM expansion uses only one slot	• True RS-232 signal levels provide maximum compatibility with peripheral devices	• Adds 4 slots to the memory expansion port
• Low power consumption provides cool, reliable operation and extended product life	• Duel output connectors make installation a snap	• Includes 3K of RAM
• Fully VIC-20 compatible	• Fully compatible with VIC-20 hardware and software	• A socket for a 2764 EPROM Modular power supply to reduce the load on your VIC
• Available in 8K, 16K, 24K configuration		
$199.24	**$49.32**	**$69.64**

ALL PRICES ARE FOB SANTA CLARA

HALLMARK COMPUTER PRODUCTS, INC.
2565 Scott Blvd., Santa Clara, CA 95050
Phone (408) 748-9208

Gold-plated contact fingers for long-lasting solid connection.

ALL PRODUCTS ARE SUPPORTED BY A 2 YEAR WARRANTY!
DEALER DISTRIBUTOR INQUIRIES INVITED

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

the computer comes across a set of parentheses with a number enclosed following a variable name, it knows you are indicating an array. From now on, we will call each separate value in an array an *element*. In our previous DIMension statement, we indicated that X would have 16 elements, and Y would have 21. In the assignment statement, we set element 5 of the X array to 27.3.

As an example of the use of arrays, let's take a look at Program 1, which is part of my friend's program.

Program 1

```
10 PRINT"{CLR}":DIM R(10)                    :rem 221
100 FOR N=1 TO 10                             :rem 56
110 R(N)=INT(RND(1)*10+1)                     :rem 73
120 NEXT N                                    :rem 32
130 REM MAIN PART OF{6 SPACES}PROGRAM FOL
    LOWS                                     :rem 167
140 GOSUB 500                                :rem 170
150 PRINT:PRINT"PRESS A KEY TO COMPUTE";:
    PRINT"{3 SPACES}ANOTHER AVERAGE"
                                             :rem 247
160 GETA$:IF A$=""THEN 160                    :rem 81
170 PRINT"{CLR}":GOTO100                       :rem 0
500 REM SUBROUTINE FOR{4 SPACES}COMPUTING
     AVERAGE{5 SPACES}R                      :rem 115
510 SM=0                                     :rem 163
515 PRINT:PRINT"{RVS}ARRAY{OFF}{2 SPACES}
    {RVS}VALUES{OFF}":PRINT                  :rem 145
520 FOR N=1 TO 10                             :rem 62
530 SM=SM+R(N)                                :rem 49
535 PRINT"R(";N;")=";R(N)                    :rem 130
540 NEXT N                                    :rem 38
550 AV=SM/10                                 :rem 158
560 PRINT:PRINT"AVERAGE =";{5 SPACES}AV
                                              :rem 61
570 RETURN                                   :rem 124
```

Line 10 contains the DIMension statement. Lines 100–120 assign ten random numbers to the ten locations or variables of the R array. The main part of the program is irrelevant to our discussion of arrays, but the subroutine starting at line 500 uses the array further and is a good example. The program is written to find the average value of the ten numbers. The sum is first set to zero in line 510. The FOR-NEXT loop (lines 520–540) recalls the values stored previously in line 110 and computes the sum, which is divided by ten to compute the average in line 550.

Two-Dimensional Arrays

Arrays can have more than one dimension. The arrays we've seen so far are one-dimensional. We can visualize the one-dimensional array as a line of boxes or pigeonholes, as in Figure 1, in which to place values, or a list of values like a list on a piece of paper. The one-dimensional array is probably the most common, but the two-dimensional array is used often, too. The two-dimensional array is often visualized as a table of rows and columns. For instance, an array DIMensioned by

the statement:

```
DIM X(4,3)
```

would be visualized as a table of *five* columns by *four* rows, as shown in Figure 2. Again, notice that DIM X(4,3) actually sets up a 5×4 table because the elements are numbered starting with 0. As with one-dimensional arrays, you may choose to ignore the 0 column and row, spending a few bytes of memory to eliminate a possible source of confusion.

Frequently, a particular problem can be solved by either a one- or a two-dimensional array, and the choice is strictly a matter of style, up to the programmer. Programs 2 and 3 illustrate a similar problem, the first with a one-dimensional array, and the second with a two-dimensional array.

In Program 2, the problem is to record the high temperature for each day, and then find the average high temperature for the week.

Program 2

```
20 DIM TM(7)                              :rem 101
30 REM ENTER DATA                         :rem 223
40 INPUT "{CLR}ENTER DAY NUMBER";N
                                          :rem 121
50 PRINT:PRINT "ENTER HIGH TEMPERATURE FO
   R DAY":INPUT TM(N)                     :rem 184
60 IF N<7 THEN 40                          :rem 73
70 REM                                     :rem 75
80 REM A SUBROUTINE.{5 SPACES}NOT SHOWN H
   ERE,{7 SPACES}WOULD STORE THE
   {7 SPACES}ARRAY TO TAPE                 :rem 224
100 REM                                   :rem 117
120 GOSUB 1000                            :rem 212
130 END                                   :rem 107
1000 REM ROUTINE FOR{7 SPACES}FINDING AVE
     RAGE{7 SPACES}HIGH TEMPERATURE
                                          :rem 26
1010 REM A ROUTINE FOR{5 SPACES}READING T
     HE TAPE,{5 SPACES}NOT SHOWN, WOULD
     {6 SPACES}BE INCLUDED HERE           :rem 79
1030 PRINT                                 :rem 81
1040 SM=0                                 :rem 210
1050 FOR N=1 TO 7                          :rem 67
1060 SM=SM+TM(N)                          :rem 175
1065 PRINT"DAY";N;"TEMP=";TM(N)           :rem 113
1070 NEXT N                                :rem 85
1080 AV=INT(SM/7)                         :rem 223
1090 PRINT:PRINT"AVERAGE HIGH":PRINT"TEMP
     ERATURE FOR WEEK=";AV;" DEGREES
                                          :rem 84
1100 RETURN                               :rem 162
```

The one-dimensional array TM is DIMensioned to 7. An actual application program would have some sort of data file routines, but since tape or disk file handling is another subject altogether, let's leave the storage and retrieval out. Lines 40 and 50 assign the value of the high temperature to the appropriate box in the array. The average high temperature is then found in the subroutine starting at line 1000, in the same manner as in the preceding problem.

Program 3 handles a similar problem using a two-dimensional array.

Program 3

```
20 DIM TM(52,7)                          :rem 248
30 REM ENTER DATA                        :rem 223
40 INPUT"{CLR}ENTER WEEK NUMBER ";WK
                                         :rem 27
50 INPUT"ENTER DAY OF WEEK ";DY          :rem 46
60 PRINT"ENTER HIGH TEMPERATURE":INPUT TM
   (WK,DY)                               :rem 74
65 IF WK<52 THEN 40                      :rem 210
70 REM                                   :rem 75
80 REM SUBROUTINE 500,{3 SPACES}NOT SHOWN
    HERE,{7 SPACES}WOULD STORE            :rem 5
90 REM DATA ON TAPE                      :rem 46
100 REM GOSUB 500 TO TAPE ROUTINE HERE
                                         :rem 161
110 GOSUB 1000                           :rem 211
120 END                                  :rem 106
1000 REM READ TAPE AND{5 SPACES}COMPUTE A
     VERAGE                              :rem 214
1010 REM A TAPE READ{7 SPACES}ROUTINE, NO
     T{10 SPACES}SHOWN, WOULD BE
     {7 SPACES}FOUND HERE                :rem 221
1030 REM                                 :rem 168
1040 S1=0                                :rem 182
1050 FOR W=1 TO 52                       :rem 124
1060 S2=0                                :rem 185
1070 FOR D=1 TO 7                         :rem 59
1080 S1=S1+TM(W,D)                       :rem 242
1090 S2=S2+TM(W,D)                       :rem 245
1100 NEXT D                               :rem 69
1110 WA=S2/7                             :rem 131
1120 PRINT"WEEK ";W;"AVERAGE IS ";WA;"DEG
     REES"                               :rem 186
1130 NEXT W                               :rem 91
1140 YA=S1/365                           :rem 238
1150 PRINT"YEARLY AVERAGE HIGH TEMP"
                                         :rem 191
1160 PRINT"IS ";YA;" DEGREES"            :rem 136
1170 RETURN                              :rem 169
```

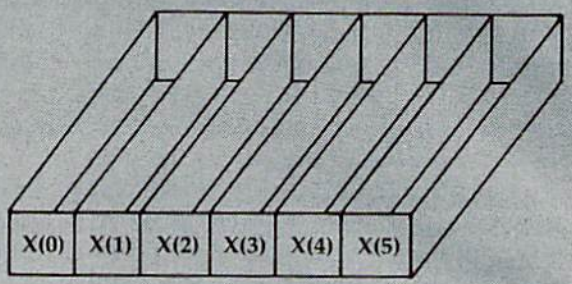
In this version, we store the temperatures week by week and day by day in a table of 52 rows of 7 columns (line 20). We have a column for every day of the week, and a row for every week of the year. The first part of the program stores our data in the array by week number and the number of the day in the week. The subroutine starting at line 1000 again figures the average, but with a new twist (as an advantage of using the two-dimensional array). Now we can find the average temperature for each week as well as for the year.

Another Use Of Arrays

Another handy use of arrays is to relate two sets of values to one another. This can easily be done if each set of values is an array, and these values can then be related by the subscript. A common use of arrays for this purpose is relating a set or sets of values to people's names. The names are held in a *string array*, such as N$(X), while the values are held in *numeric arrays* (having the same dimensions as N$, of course). Program 4 illustrates this use of arrays in a teacher's gradebook program.

Figure 1: One-Dimensional Array

A one-dimensional array can be thought of as a row of boxes or pigeonholes.



DIMX(S)

Figure 2: Two-Dimensional Array

A two-dimensional array is frequently visualized as a table of rows and columns.

	Column 0	Column 1	Column 2	Column 3	Column 4
Row 0	X(0,0)	X(1,0)	X(2,0)	X(3,0)	X(4,0)
Row 1	X(0,1)	X(1,1)	X(2,1)	X(3,1)	X(4,1)
Row 2	X(0,2)	X(1,2)	X(2,2)	X(3,2)	X(4,2)
Row 3	X(0,3)	X(1,3)	X(2,3)	X(3,3)	X(4,3)

Program 4

```
20 DIM N$(15),T1(15),T2(15),HW(15),FS(15)
                                         :rem 52
30 PRINT"{CLR}"                          :rem 199
40 REM DISPLAY MENU                      :rem 147
50 PRINT"{4 SPACES}{RVS}SELECT OPTION
   {OFF}"                                :rem 115
60 PRINT:PRINT"1-ENTER NAMES IN FILE"
                                          :rem 5
70 PRINT:PRINT"2-ENTER SCORES, FIRST
   {3 SPACES}TEST"                       :rem 159
80 PRINT:PRINT"3-ENTER SCORES, SECOND
   {2 SPACES}TEST"                       :rem 213
90 PRINT:PRINT"4-ENTER SCORES,{9 SPACES}H
   OMEWORK"                              :rem 71
100 PRINT:PRINT"5-COMPUTE FINAL SCORE"
                                         :rem 142
110 PRINT:INPUT"ENTER NUMBER";Q     :rem 0
120 ON Q GOSUB 1000,2000,3000,4000,5000
                                         :rem 128
130 END                                  :rem 107
1000 REM INITIALIZE{7 SPACES}STUDENT NAME
      FILE                               :rem 255
1010 FOR N=1 TO 15                       :rem 110
1020 INPUT"ENTER LAST NAME";N$(N):rem 182
1030 NEXT                                 :rem 3
1040 OPEN 1,1,2,"NAMES"                  :rem 199
1050 FOR N= 1 TO 15                      :rem 114
```

Presenting

An income tax program for everyone to use. TAX COMPUTATION has been approved for **COMMODORE'S ENCYCLOPEDIA OF SOFTWARE** as well as many other directories. Your taxes will be completed quickly, easily, and to your benefit.

TAX COMPUTATION Commodore 64 $39.95
VIC 20, TI99/4A $19.90

K. R. Rullman
4550 Murray # 81
Beaverton, OR 97005

The cost of the program is tax deductible along with part of the cost of your computer.

An advertisement in December's issue of Compute!'s Gazette stated that a survey conducted in PC Magazine had recently named Screenplay's Asylum™ one of the top 10 adventure games. We regret that the name "PC Magazine" was used in error. PC World conducted the survey that named Asylum one of the top 10 adventure games for the IBM PC. Screenplay regrets any inconveniences caused by this error.

screenplay

STOP PLAYING GAMES

NEW Disk Commodore 64

- Calculate odds on HORSE RACES with ANY COMPUTER using **BASIC**.
- SCIENTIFICALLY DERIVED SYSTEM really works. TV Station WLKY of Louisville, Kentucky used this sytem to predict the odds of the 1980 Kentucky Derby. See the Wall Street Journal (June 6, 1980) article on Horse-Handicapping. This system was written and used by computer experts and is now being made available to home computer owners. This method is based on storing data from a large number of races on a high speed, large scale computer. 23 factors taken from the "Daily Racing Form" were then analyzed by the computer to see how they influenced race results. From these 23 factors, ten were found to be the most vital in determining winners. NUMERICAL PROBABILITIES of each of these 10 factors were then computed and this forms the basis of this REVOLUTIONARY NEW PROGRAM.
- SIMPLE TO USE: Obtain "Daily Racing Form" the day before the races and answer the 10 questions about each horse. Run the program and your computer will print out the odds for all horses in each race. COMPUTER POWER gives you the advantage!
- YOU GET: 1) Program on cassette or disk.
  2) Listing of BASIC programs for use with any computer.
  3) Instructions on how to get the needed data from the "Daily Racing Form."
  4) Tips on using the odds generated by the program.
  5) Sample form to simplify entering data for each race.

MAIL COUPON OR CALL TODAY

3G COMPANY, INC. DEPT. GA **(503) 357-5607**
RT. 3, BOX 28A, GASTON, OR 97119

Yes, I want to use my computer for FUN and PROFIT. Please send me ____ programs at **$29.95** each. Circle the cassette you need: PET/CBM, VIC-20, Color Computer, TRS-80, Sinclair Timex 1000, Atari, Commodore 64 (disk or cassette), Apple (disk or cassette)

Enclosed is: ☐ check or money order ☐ MasterCard ☐ Visa

Card No. Exp. date

NAME

ADDRESS

CITY STATE ZIP

START USING YOUR COMPUTER FOR FUN and PROFIT!

HOW MUCH LONGER WILL YOU LAST?

How long can you endure? When will it end?... We're not talking about a new shoot'em up game for the Commodore 64, but Commodore's own disk operating system! Commodore made a great computer in the 64 but left its disk operating system out in the cold. If you've been waiting for a true disk operating system, here it is!... If you've been waiting for a great BASIC language enhancement that will let you utilize the Commodore's many special features, here it is! What is it? It's **grafDOS**, the great new utility from Xylex Software that allows the user to actually become friendly with the Commodore 64! **grafDOS** includes commands like DELETE, RENAME, CATALOG, RUN, etc. The BASIC allows you to do high resolution and low resolution graphics, sound, sprite program, plus much, much more for a total of 40 commands! Plus included in every package is MINIMON, a powerful machine language monitor that includes another 20 commands for use in machine language. The disk also comes with sample programs and demos including a great music generator! And all this together is only $49.95! How could you have lasted this long without it?

DON'T WAIT ANY LONGER!

Make your programming easier! **grafDOS** is available now at your local dealer or:

INTERESTING SOFTWARE
21101 S. Harvard Blvd.
Torrance, CA 90501
(213) 328-9422

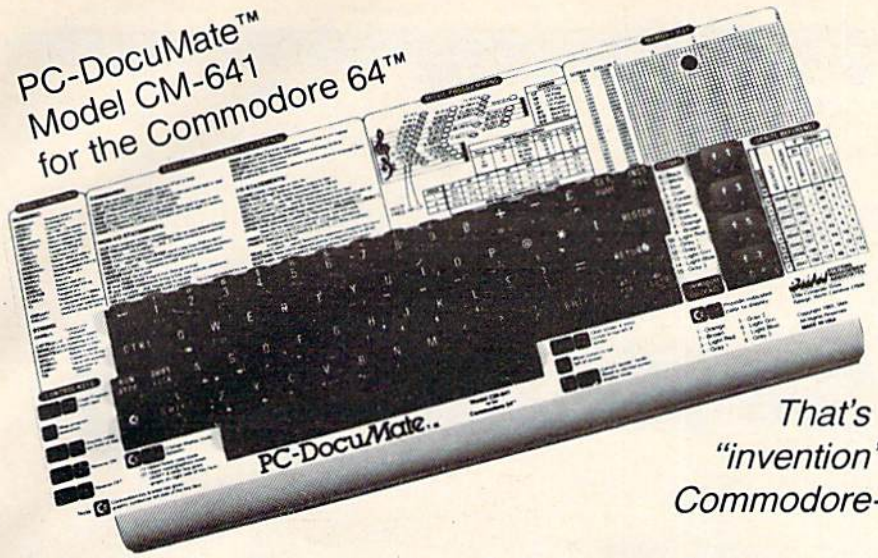
Visa/MC/Check/Money Order
Add $2.00 shipping
CA residents add 6½% sales tax.
Dealer inquiries invited.

```
1060 PRINT#1,N$(N)                    :rem 229
1070 NEXT N                           :rem 85
1080 CLOSE 1                          :rem 112
1090 RETURN                           :rem 170
2000 REM ENTER TEST SCORES            :rem 51
2010 OPEN 1,1,0,"NAMES"               :rem 195
2020 FOR N=1 TO 15                    :rem 112
2030 INPUT#1,N$(N)                    :rem 230
2040 NEXT N                           :rem 83
2045 CLOSE 1                          :rem 114
2050 REM ENTER DATA BY{5 SPACES}NAME
                                      :rem 255
2060 FOR N=1 TO 15                    :rem 116
2070 PRINT"ENTER SCORE FOR ";N$(N)
                                      :rem 199
2080 INPUT T1(N)                      :rem 126
2090 NEXT N                           :rem 88
2100 REM NOW SAVE T1{7 SPACES}ARRAY AS FI
     LE TO{6 SPACES}TAPE              :rem 79
2110 OPEN 2,1,2,"TEST1"               :rem 196
2120 FOR N=1 TO 15                    :rem 113
2130 PRINT#2,T1(N)                    :rem 248
2140 NEXT                              :rem 6
2150 CLOSE 2                          :rem 112
2160 RETURN                           :rem 169
3000 REM NOW WOULD{9 SPACES}FOLLOW TWO MO
     RE                               :rem 38
3010 REM SUBROUTINES{7 SPACES}LIKE THE ON
     E{10 SPACES}ABOVE, EXCEPT        :rem 85
3020 REM REPLACE T1{8 SPACES}WITH T2 IN S
     UB-{7 SPACES}ROUTINE STARTING:rem 44
3030 REM AT LINE 3000,{5 SPACES}AND CALL
     {SPACE}THE FILE{5 SPACES}"TEST2".
                                      :rem 42
3040 REM THEN USE HW{7 SPACES}AND FILENAM
     E{9 SPACES}"HMWRK" FOR THE       :rem 15
3050 REM ROUTINE AT 4000              :rem 43
4000 REM HOMEWORK FILE{5 SPACES}HERE
                                      :rem 88
5000 REM READ TAPE{9 SPACES}FILES AND COM
     PUTE{5 SPACES}SCORE              :rem 206
5010 OPEN 1,1,0,"NAMES"               :rem 198
5020 FOR N=1 TO 15                    :rem 115
5030 INPUT#1,N$(N)                    :rem 233
5040 NEXT                              :rem 8
5050 CLOSE 1                          :rem 113
5060 OPEN 2,1,0,"TEST1"               :rem 201
5070 FOR N=1 TO 15                    :rem 120
5080 INPUT#1,T1(N)                     :rem 1
5090 NEXT                             :rem 13
5100 CLOSE 2                          :rem 110
5105 INPUT"HIT RETURN TO CONTINUE";Q
                                      :rem 248
5110 OPEN 3,1,0,"TEST2"               :rem 199
5120 FOR N=1 TO 15                    :rem 116
5130 INPUT#3,T2(N)                     :rem 0
5140 NEXT                              :rem 9
5150 CLOSE 3                          :rem 116
5155 INPUT "HIT RETURN TO CONTINUE";Q
                                      :rem 253
5160 OPEN 4,1,0,"HMWRK"               :rem 228
5170 FOR N=1 TO 15                    :rem 121
5180 INPUT"4,HW(N)                     :rem 30
5190 NEXT N                           :rem 92
5200 CLOSE 4                          :rem 113
5210 REM NOW COMPUTE{7 SPACES}FINAL SCORE
                                      :rem 163
5220 FOR N= 1 TO 15                   :rem 117
5230 FS(N)=T1(N)+T2(N)+HW(N)           :rem 28
5240 NEXT N                            :rem 88
5250 REM NOW PRINT OUT{5 SPACES}SCORES
                                      :rem 248
5260 OPEN 1,4,7                       :rem 243
5270 PRINT"1,"NAME","SCORE"            :rem 43
5280 FOR N = 1 TO 15                  :rem 123
5290 PRINT#1,N$(N),FS(N)               :rem 82
5300 NEXT N                            :rem 85
5310 RETURN                           :rem 169
```

For demonstration purposes, this program is not a complete program as it stands, and contains no error trapping or user prompts. It could, however, be expanded into a useful gradebook program with some fill-in work. It is instructive of the use of arrays to relate variables. The main program, up to line 130, creates a menu selection which sends the program to the appropriate subroutine. The first routine, starting at line 1000, is used at the beginning of the school term to enter the students' names in a string array, N$(N). The DIMension statement in line 20 of the main program, and all of the FOR-NEXT loops, would have to be adjusted to the actual number of students in the class. Subroutine 2000 would be used to enter the scores of the first test. By reading the N$ array in lines 2010 to 2045, the program prompts the teacher with the student's name for data entry (line 2070). A similar subroutine would be used for each test and maybe a homework score. Subroutine 5000 puts it all together at the end of the term. After reading the grades from all the files, line 5230 figures the grade for every student. In effect, the variable N is a student number which relates each element of each of the four files. This illustrates how N can still be used as a separate variable, even when you've set up a numeric array N(X) or a string array N$(X).

These examples of the use of the array are general but easy to expand on. Arrays can be used in a variety of ways. I'm sure that after using them for a while, you can come up with many more applications on your own.

PC-DocuMate™
Model CM-641
for the Commodore 64™



EUREKA!

That's what we said when our new "invention" solved all our VIC-20™ and Commodore-64™ programming problems

We had a problem. So we invented PC-DocuMate™ to solve it. The problem was how to quickly master the VIC-20 and CBM-64 keyboards and easily start programming in BASIC on our new personal computers. First we went through the manuals.

INCONVENIENT MANUALS

The user's guide was a nuisance and the programmer's reference manual was just plain inconvenient to use. We found the control key combinations confusing and the introduction to BASIC to be too "basic" for our needs. We needed a simple solution to our documentation problems.

So we decided to surround the keyboard of each PC with the information we wanted. We decided to print whatever we needed on sturdy **plastic templates** which would fit the keyboard of either the VIC-20 or Commodore 64.

SIMPLE SOLUTION

This was the simple solution to our problem. Now we could have the essential information right at our fingertips.

On the left side and top of the templates we put **BASIC** functions, commands, and statements. On the lower left we used **key symbols** to remind us of how to use SHIFT, RUN/STOP, CTRL and the "Commodore" key. Over on the bottom right side we put some additional keys to help remember about CLR/HOME and RESTORE. But we were still a little confused.

STILL CONFUSED

We found we were confused about music programming, color graphics, and sprites. On both the VIC-20 and the CBM-64 templates we carefully organized and summarized the essential reference data for **music** programming and put it across the top—showing notes and the scale. All those values you must POKE and where to POKE them are listed.

Then to clarify **color graphics** we laid out screen memory maps showing character and color addresses in a screen matrix. (We got this idea from the manuals.)

For the VIC-20 we added a complete memory address map for documenting where everything is in an expanded or unexpanded VIC.

For the Commodore 64 we came up with a really clever summary table for showing almost everything you ever need to know for **sprite** graphics.

GETTING EASIER

Now we had organized the most essential information for our VIC and 64 in the most logical way. BASIC, music, color graphics, and sprites all seemed a lot easier. Our initial problem was solved by PC-DocuMate™.

But we have a confession to make.

WE CHEATED

We had solved this kind of problem before. In fact, many times before. You see, we at SMA developed the original PC-DocuMate for the IBM PC. We've made templates for IBM BASIC and DOS, for WORDSTAR™, VISICALC™ and other best-selling software packages for the IBM PC.

So we knew we could invent another PC-DocuMate™ to solve our problems with the VIC-20 and Commodore 64. Now our solution can be yours and you can join the thousands of satisfied users of our template products.

Take advantage of our experience and success with PC-DocuMate templates. Get one for your personal computer.

SOME SPECIFICS

Our templates for the VIC and 64 are made from the same high quality **non-glare** plastic as the more expensive IBM PC versions.

The templates are an attractive **gray** color and are imprinted with a special black ink which bonds permanently to the plastic. They are precision **die-cut** to fit your keyboard.

Unlike some other products we've seen in this category, PC-DocuMate templates are professionally and expertly designed. And they are fully guaranteed.

OUR GUARANTEE

We guarantee your satisfaction. **You must be satisfied** with your PC-DocuMate for your VIC-20 or CBM-64. Try it for 10 days and if for any reason you are not satisfied return it to us (undamaged) for a full refund. **No risk.**

SOLVE YOUR PROGRAMMING PROBLEMS WITH PC-DocuMate™

Order your PC-DocuMate today (by phone or mail) and solve your VIC-20 or CBM-64 programming problems. Send only **$12.95** and specify which computer you have. We pay for shipping and handling. Use the coupon below or call **919-787-7703** for faster service.

YES! Please RUSH me _____ VIC-20 templates and/or _____ CBM-64 templates at $12.95 each. I have enclosed $_____ by:

Check ___ Money order ___ MC/VISA ___

Name _____

Address _____

City _____ State _____ Zip _____

Card # _____ Exp. _____

Signature _____

Foreign orders (except Canada) add $5.00 US

Mail to: **Systems Management Associates**
3700 Computer Drive, Dept. J-1
P.O. Box 20025
Raleigh, North Carolina 27619

Dealer inquiries invited.

Auto Line Numbering

Jeff Young

This short program is a handy, time-saving utility for programmers. For the VIC and 64.

"Auto Line Numbering" is a utility for programmers which automatically generates a line number for the current BASIC program statement being entered. As written, the program begins with line 100 and increments by tens (100, 110, 120, etc.). This can be modified as described below.

How To Use The Program

"Auto Line Numbering" is a BASIC program which loads a machine language subroutine into a free block of memory. Program 1 (the 64 version) puts the subroutine at memory location 49152 ($C000). Program 2 (the VIC version) puts the subroutine at 7578 ($1D9A). These areas of memory will not be used by BASIC, so the program should be safe.

Type in the program and SAVE it. After LOADing, type RUN, press RETURN, type NEW, press RETURN, then type SYS 49152 (for the 64) or SYS 7578 (for the VIC). If you wish to leave the program for any reason, just press RETURN immediately after you see a new line number. To return to the program, type SYS 49160 (for the 64) or SYS 7586 (for the VIC). This will continue generating line numbers from where you left off.

Although the program will always begin numbering with 100 and increment by tens, you can modify either of these if you wish. If you want to begin with a number other than 100, determine the number with which you want to start, then subtract ten. Next, POKE this number in low-byte/high-byte format into 251 and 252, then SYS 49160 (for the 64) or SYS 7586 (for the VIC).

For example, if you wish to begin with line 1000, subtract ten. The number you are now working with is 990. To determine low-byte/high-byte, divide 990 by 256. The result, 3, is the number you POKE into location 252—POKE252,3. The remainder of the division is 222. Now, POKE 251,222. The low byte is location 251, and the high byte, 252.

The lines you would type, then, if you wished to begin the line numbering with 1000 are:

```
POKE 251,222:POKE 252,3
SYS 49160   (for the 64)
SYS 7586    (for the VIC)
```

To change the increment from ten, POKE the desired number into location 49179 (for the 64) or 7605 (for the VIC). If you want to increment by fives, for example:

```
POKE 49179,5   (for the 64)
POKE 7605,5    (for the VIC)
```

This utility program can save you a lot of time when programming, and it provides a neat, structured sequence for program line numbers.

Program 1: 64 Version

```
1 X=49152                                   :rem 203
2 READY:IFY=-1THEN4                         :rem 199
3 POKEX,Y:X=X+1:Z=Z+Y:GOTO2                 :rem 22
4 IFZ<>12374THENPRINT"ERROR IN DATA STATE
  MENTS":END                                :rem 236
100 DATA169,90,133,251,169,0,133,252,169,
    19,141,2,3,169,192,141,3,3,96,32,25
                                            :rem 203
110 DATA192,76,134,164,24,169,10,101,251,
    133,251,144,2,230,252,165,251,133;99
                                            :rem 246
120 DATA165,252,133,98,162,144,56,32,73,1
    88,32,221,189,162,0,189,1,1,240,9,32
                                              :rem 4
```

PROSYS

The Professional Systems People And

MICRO WORX

Present Products From

commodore

And

The Software That Makes Them Work!

SOFTWARE

SBSYS
C-64, 8032, 8096 & B-Series
THE SMALL BUSINESS SYSTEM
Available for 1541, 8050 and hard disk drives. GL, AP, AR, INV. and payroll as low as $99.00 each! Call for specific pricing.

PERSYS
VIC 20, C-64, 8032, 8096 & B-Series
THE PERSONAL FINANCIAL SYSTEM
A complete financial package for home and small business, beginning at $69.00 on tape.

VERTICAL PACKAGES INCLUDE:

LEGISYS
8032, 8096 & B-Series.
The total legal office information, accounting and tickler system.

LOADSYS
8032, 8096 & B-Series.
The total truck brokerage accounting system. Call for free intro consulting.

Dealer inquiries invited.

VISA & MasterCard. Add 3% Surcharge.
Shipping paid on prepaid orders.
Prices subject to change without notice.

These are sample unit prices.
We carry support items, cables, games...
WE HAVE IT!

CBM PRODUCTS

8032 Computer	$ 619.00
8050 Disk Drive	979.00
8250 Disk Drive	1279.00
9060 Hard Disk	1979.00
8023 Printer	529.00
6400 Printer	1399.00

C-64 STUFF

C-64 Computer	$219.00
1541 Disk Drive	249.00
1701 Monitor	249.00
1526 Printer	339.00
1600 Modem	69.00

Call Toll-Free by dialing:
Outside Texas:
1-800-221-WORX
Inside Texas:
1-800-692-4265,
wait for beep, then dial 008-3378,
wait for tone and dial 993.
or Lubbock 797-2623,
Dallas/Ft. Worth 498-8080.

MICRO WORX
797-2623
4210 D 50th St. Lubbock, TX 79413

```
130 DATA210,255,157,0,2,232,208,242,32,18
    ,225,201,13,240,3,76,105,165,56,165
                                    :rem 182
140 DATA251,233,20,176,2,198,252,169,131,
    141,2,3,169,164,141,3,3,76,118,165,-1
                                    :rem 36
```

Program 2: VIC Version

```
10 POKE56,29:POKE52,29:CLR:I=7578:rem 175
20 READ A:IF A=256 THEN 40          :rem 54
30 POKE I,A:I=I+1:CK=CK+A:GOTO 20:rem 129
40 IF CK<>12545 THENPRINT"{CLR}ERROR IN D
   ATA STATEMENTS"                  :rem 210
7578 DATA 169,90,133,251,169,0,133
                                    :rem 156
7586 DATA 252,169,173,141,2,3,169:rem 107
7594 DATA 29,141,3,3,96,32,179     :rem 215
7602 DATA 29,76,134,196,24,169,10:rem 104
7610 DATA 101,251,133,251,144,2,230
                                    :rem 169
7618 DATA 252,165,251,133,99,165,252
                                      :rem 3
7626 DATA 133,98,162,144,56,32,73:rem 107
7634 DATA 220,32,221,221,162,0,189
                                    :rem 132
7642 DATA 1,1,240,9,32,210,255     :rem 186
7650 DATA 157,0,2,232,208,242,32   :rem 33
7658 DATA 15,225,201,13,240,3,76   :rem 40
7666 DATA 105,197,56,165,251,233,20
                                    :rem 203
7674 DATA 176,2,198,252,169,131,141
                                    :rem 208
7682 DATA 2,3,169,196,141,3,3      :rem 154
7690 DATA 76,118,197,0,256        :rem 21
```

To receive additional information from advertisers in this issue, use the handy reader service cards in the back of the magazine.

GET THE BEST FOR YOUR COMMODORE 64

BUSINESS

* **FINANCE CALC 64** • Disk	$45.95
* **DATA BASE 64** • Disk	49.95
* **MANAGEMENT SYSTEM 64** • Disk	45.95
* **FAMILY PAC 64 (3 in 1)** • Disk (CHECKBOOK, RECIPE, EDU-GAME)	45.95
* **CHECKBOOK EASE 64** • Disk	29.95
HESWRITER 64 • Cart.	29.95
HESMON 64 • Cart.	25.95
HES MODEM • Cart.	67.95
TURTLE GRAPHICS II • Cart.	39.95
QUICK BROWN FOX (W.P.) • Cart.	45.95
WRITERS ASSISTANT (W.P.) • Disk	59.95
FILING ASSISTANT • Disk	67.95
INVENTORY PACKAGE • Disk	77.95
TOUCH TYPING TUTOR • Disk & Cass.	18.95
CALC RESULT EASY • Cart	67.95
CALC RESULT ADVANCED • Disk	127.45
PAPERCLIP (W.P.) • Disk	99.95
M'FILE • Disk	94.95
WORD PRO/3 (W.P.) • Disk	71.95
SPELL RIGHT PLUS (DICTIONARY) • Cart.	49.95
DELPHI'S ORACLE (DATA BASE) • Disk	125.95
TIME & MONEY MANAGER • Disk	55.95
OMNICALC (SPREADSHEET) • Disk	79.95
CARDCO PRINTER INTERFACE	54.95

EDUCATIONAL

HUNDREDS MORE AVAILABLE

SNOOPER TROOPERS I, II • Disk	$29.95
KINDERCOMP • Disk & Crt.	19.95
IN SEARCH OF MOST AMAZING THING • Disk	26.95
PROGRAMMING KIT I • Disk	19.95
FACEMAKER • Disk	22.95
KIDS ON KEYS • Cart.	29.95
FRACTION FEVER • Cart.	29.95
PIPES • Cart.	29.70
ENGLISH INVADERS • Disk & Cass.	21.95
DUNGEONS ALGEBRA DRAGONS • Disk & Cass.	19.95
UP FOR GRABS • Cart.	29.95
BENJI'S SPACE RESCUE • Disk	29.95

ENTERTAINMENT

* **STUDIO 64** (MUSIC MAKER) • Disk & Cass.	$29.95
* **GAME DESIGNER** • Disk & Cass.	25.95
GRIDRUNNER • Cart.	20.25
TEMPLE OF APSHAI • Disk	25.95
UPPER REACHES OF APSHAI • Disk	13.50
CURSE OF RA • Disk	13.50
ASTROBLITZ • Cart.	29.20
SAVE NEW YORK • Cart.	29.70
PERSONALITY ANALYZER • Disk	28.00
PHANTOM KARATE DEVILS • Disk	29.70
PLANETFALL • Disk	38.20
ENCHANTER • Disk	38.20
SEA FOX • Disk	33.95
CHOPLIFTER • Disk	33.95
PROTECTOR II • Disk & Cass.	24.95
TELENGARD • Cass.	16.95
FROGGER • Disk & Cass.	22.95
FORT APOCALYPSE • Disk & Cass.	22.95
ROBBERS OF THE LOST TOMB • Disk	19.95
JUMPMAN • Disk	25.95
SWORD OF FARGOAL • Disk & Cass.	20.25
PAKACUDA • Disk & Cass.	11.95
SURVIVOR • Disk & Cass.	22.95
PEGASUS ODYSSEY • Disk & Cass.	19.95
NEUTRAL ZONE • Disk & Cass.	27.95
COMPETITION PRO. JOYSTICK	17.95

POLICY

All orders are shipped U.P.S. Shipping charges are $2.00 for prepaid orders and $3.25 for C.O.D. For fast delivery send money order, certified check or credit card.
Please allow approximately three weeks for clearance on personal checks.
All items are subject to availability and price change. Thanks for ordering from House of Software! Call for free catalog!

HOUSE of SOFTWARE

*From EN-TECH Software

To Order Call:
(213) 768-8866
Or
Write To:

VISA MasterCard

HOUSE OF SOFTWARE
9183 Mercedes Ave. • Arleta, CA 91331

A-V Graphics N. Hollywood

SHOW US A BETTER PRICE AND WE'LL BEAT IT!

IF IT'S FOR THE COMMODORE 64 AND IT'S GOOD, IT'S PROBABLY

ENTECH

MANAGEMENT SYSTEM 64
This integrated business program gives you the computer power once reserved for large corporations. Capabilities include invoicing, inventory control, and customer mailing lists. Disk **$69.95**

FINANCE CALC 64
The leader in home and business financial analysis. You can have up to 1440 itemized expenses and print 1085 different financial reports and bar graphs. In addition, it keeps and compares as many as 12 budgets at once. Disk **$59.95**

DATA BASE 64
A perfect record system for any business or home. It can store up to 1200 records and has up to 20 fields for each one. A special label and report designer is included. It can also merge with popular word processors. Disk **$59.95**

GAME DESIGNER 64
Use to animate 16 sprites and design colorful background screens. Several game sub-routines included. Disk **$35.95**

STUDIO 64 SERIES
Anyone can now create music as beautiful as the most advanced programers could one year ago!! Just play and the computer will instantly write the music on the screen. Included are powerful features like block move, single note editing and scrolling. It will save and recall, add music to your own programs and print lead sheets. Disk **$39.95**

FAMILY PAK (3 in 1) **(ALL) $55.95**
Three of the finest home programs available:

- **CHECKBOOK EASE 64**
  Handles over 1300 transactions. Prints statements, and all types of checks, and 40 expense categories. **$39.95**
- **RECIPE KEEPER**
  Searches by ingredient, category or name. Calculates measurements for different serving amounts and prints copies.
- **SPACE MATH 64**
  Learn math, explore the universe, dance to the music and watch the show.

BABIES OF THE DIRT
An earthquake sucks you to the center of the earth. To escape you must battle the BABIES OF THE DIRT. But, don't miss or its doomsday! Watch out for their mother. **$39.95**

NOTE: PROGRAMS ARE COMPATIBLE WITH ALL PRINTERS AND UTILIZE FULL-SCREEN PROCESSING.

A-V Graphics N. Hollywood

P.O. BOX 881, SUN VALLEY, CA 91353 • [213] 768-6646

HORIZONS: 64

CHARLES BRANNON
PROGRAM EDITOR

Singing 64

I recently got to hear The Alien Group's Voice Box speech synthesizer for the 64. If you've never heard your computer talk, it's really quite shocking.

The Voice Box is indeed a "black box" which plugs into the user port. It has a built-in amplifier and speaker, as well as two knobs for volume and pitch. With the supplied software, you can very easily make speech, using English spelling. Like all speech synthesizers, you may need to spell a word phonetically to get it to sound right, since it's almost impossible for software to master the ambiguity of English spelling.

The sample software shows what you can do with the Voice Box, including a spelling game which reads the words aloud. Most interesting is a music composition program. You can enter three-part synthesizer music and—get this—the Voice Box can sing along. The sample music is rich and vibrant, some of the best 64 sound I've heard, but that singing voice takes a bit of getting used to. If a crooning computer isn't enough, there is also a high-resolution face which changes its expression while it mouths the words. You can even edit the face to add a moustache, if you want.

There is a lot of complexity here. I would be happy with the music software alone (you don't have to use the singing option—you can compose music without the Voice Box), but the face and song could enliven any party. (A local user group enthusiastically watched a demonstration of the Voice Box and software.)

After using the Voice Box, it will be interesting to see the product of Commodore's own speech synthesis division, which is ready to market an amazing speech synthesizer for the 64. It will have chips for different "personalities" (male, female, or even children's voices). Commodore plans to market games using the speech synthesizer, such as *Gorf* and *Wizard of Wor*. I just wonder if Commodore can make it sing....

The Alien Group
27 West 23rd Street
New York, NY 10010
$95 plus $25 for music package

VIC Vs. 64: Cousins, Not Siblings

What is the real difference between a VIC-20 and a Commodore 64? This is an often-asked question. Many people would like to upgrade their VIC to a 64. Others are trying to convert programs, or would like to plug VIC cartridges into a 64. It's time to get things straight. The 64 is more than a 40-column VIC.

First, what do these machines have in common?

The keyboard similarity means that you won't have to relearn the layout if you move to a 64 from a VIC.

The common BASIC is far more important. Some VIC programs do not use any special features of the VIC, so these BASIC programs work on the 64, too. The only problem is that the different screen line lengths (22 versus 40) can cause wacky screen formatting. Some other programs that use only VIC color control and normal graphics characters also work OK on the 64.

The difference in screen width is quite significant. At best, you'll use only half of your 40-column screen width when you run VIC programs on your 64. At worst, text will be strewn all over the screen, as cursor controls and screen formatting dependent on a 22-column screen go awry.

Most BASIC game programs POKE characters to the screen for animation. Even if you change the screen address to 1024 for the 64, and color memory to 55296, the POKEs are still based on a line length of 22. Many times the POKEs appear like this:

```
POKE 7815,81
```

This is not too hard to convert. Subtract 7680, the start of screen memory:

```
?7815-7680
135
```

Divide by 22 to get the row:

```
?INT(135/22)
6
```

Now the column is the remainder:

```
?135-22*6
3
```

Now reassemble it to a 64 POKE:

```
?1024+40*6+3
1267
```

So the equivalent to the VIC POKE is:

```
POKE 1267,81
```

That wasn't so hard, eh?

Another effect of the 64's wider screen is that the characters are smaller. Frequently, VIC game characters come out looking cramped and smeared when moved over to the 64.

Internal Similarities

The operating system in Read Only Memory (ROM) is also very similar. In fact, some VIC routines were just reassembled with very minor changes to run on the 64. The Kernal routines, used by machine language programmers for working with files, are identical. Almost all of the VIC zero-page locations are unchanged in the 64. Most other low-memory locations, such as POKE 650,128 for auto-repeat on all keys, also apply to the 64. This compatibility is just as important as the BASIC. It can make a lot of your VIC experience "transportable" to the 64.

The VIC and 64 can share the same peripherals. For example, the VICmodem plugs into and works just fine on the 64. Both machines use the same disk drive, the 1541. The 64 cannot use the 1540 disk drive, though. The extra processing time the VIC-II chip steals from the 64 required that the 1540 be slowed down to let the 64 keep up, hence the 1541. The 1540 can be upgraded to a 1541 by the replacement of a single ROM chip (performed by your service technician).

The 1525 printer can likewise be modified if necessary (1525E), but will otherwise work fine with both the VIC and 64. Future peripherals should also be compatible as long as they use the serial bus or user port (the cartridge slot is still incompatible, so no CP/M for the VIC!).

The video connections are enough alike to let you use the same video monitor, such as the Commodore 1701. And thanks to relocatable loading, you can even LOAD and modify VIC programs from the same cassette drive.

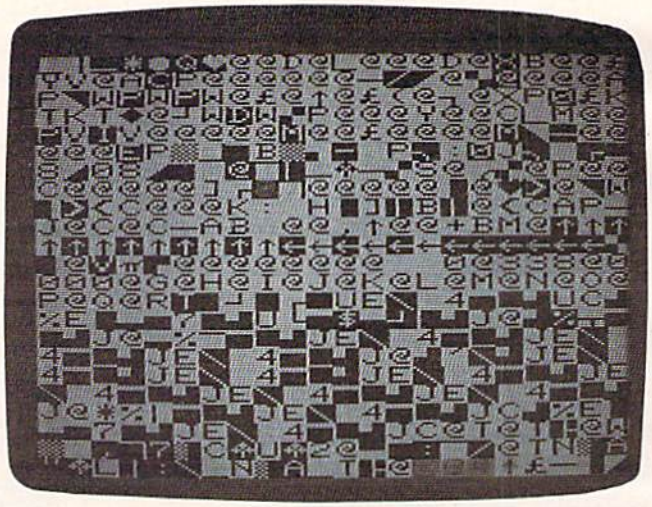
Finally, even some video graphics features are similar, most notably the same basic eight colors: black, white, red, cyan, purple, green, blue, and yellow. As mentioned, the keyboard graphics are also the same, just thickened up a bit to help their appearance on a TV. Other comparable features are: custom characters, multicolor mode, color memory, even extended background color mode.

I said comparable, not compatible. When you get to graphics and sound, the VIC and 64 are as different as Apples and oranges (or Ataris). They share a few concepts, though, such as color memory, not found on other computers. Audio/video is the major difference between the machines, and if you believe your eyes and ears, the machines have nothing in common! Fortunately, we know better.

Dan Carmichael, Submissions Editor

A Window Into The VIC-20

If you've ever wanted to take a look inside the VIC-20 and see what's going on, then this month's column is for you.



Here's a picture of what zero-page memory looks like in the VIC-20. The blurred light-blue block near the center of the screen is a rapidly changing internal clock.

This month we're going to take a guided tour through the VIC's memory. We'll see what's going on in both RAM and ROM, and some of the activities you see may be a surprise.

The Window Program

Before we begin our tour, let's take a look at the program we'll be using, a "window" into the VIC-20. It is basically a machine language program that will look at two *pages* (a page is 256 bytes) of memory and display it on the screen. With this program, you'll have complete control over what you see. You'll be able to scroll through the entire VIC's memory both forward and backward. If you wish to look at a specific area of memory, there's also an option to input that address into the program.

Type in the program at the end of this column, and save it to tape or disk before running. The program is a BASIC loader that will POKE a machine language program into memory. As you may know, a single error in a machine language program can cause the VIC-20 to lock up, so save the program before running it. We'll be using machine language here because of its speed.

Basically, what the program does is read the 512 bytes of memory the program window is currently looking at and store them directly into screen memory. This displays the memory on the screen. Because this column is not primarily aimed at advanced machine language programmers, I won't go into any details on how the program works. However, you machine language programmers may be interested in the technique used here. The program changes, or rewrites itself as it runs. It could have transferred memory to the screen using indirect addressing with zero-page locations, but the self-modifying version was chosen here.

Controlling The Program

To properly use the program, turn off your VIC, unplug all expansion RAM and other cartridges, then turn it back on. After you've typed in and debugged the program, enter RUN, then press RETURN. After a pause of about two seconds (as the machine language program is POKEd into memory), the screen fills with characters. You're now looking at the first two pages of the VIC's memory, or approximately memory addresses 0 through 505.

SJB DISTRIBUTORS. ONE STOP SHOPPING FOR COMMODORE SYSTEMS.

commodore

NEW COMMODORE PRODUCTS

Executive 64	$ Call
CBM B128-80	825
B Series Software	Call

WORD PROCESSING 64

NEW - Mirage 80 col.	$ 95
WordPro 3+/Spellright	79
WordPro 3+ (WP)	59
Spellright (Dictionary)	39
Paper Clip	95
Easy Script (D)	35
Easy Spell (D)	19
SPECIAL - Busiwriter (C,D)	39
Quick Brown Fox (R)	49

SPREADSHEETS 64

Calc Result - Advanced (R,D)	$ 120
Busicalc II - More Power! (R,D)	95
Multiplan 64 (D)	79
Calc Result - Easy (R)	70
Practicalc 64 (D)	45
Busicalc I - SPECIAL! (C,D)	39

DATA BASES 64

Mirage Data Base (D)	$ 95
M'File (merges with WordPro) (D)	89
Micro Spec Data Manager (D)	60
Codewriter (develops programs)(D)	95

PRODUCTS OF THE MONTH

TCS 64/80 - NEW! (D). (WP/Data Base/Spread)	$ 139
Koala Pad - NEW!	89
Delta 10 (160 cps) - NEW!	549

UTILITIES 64

Vic Tree (4.0 Basic) (R)	$ 75
64 Super Expander (R)	25
Simon's Basic (R)	25
Cardco Printer Utility (C)	15
MS-Backup (Back Up Data!) (D)	15

ACCOUNTING 64

Home Accountant (Continental)	$ 75
Tax Advantage (merge w/home accountant) - NEW!	45
General Ledger, A/R, A/P, P/R, Inv. (Info Design's Original) (D)	ea.79
Numeric Keypad (Hardware)	65
Numeric Keypad (Cardco)	35

TELECOMPUTING 64

Vic 1650 (Auto Ans/Dial) Modem	$ 95
Vic 1600 Modem	59
Hes Modem (Downloading Software)	65
Super Term (Download/80-128 Form).	95
Micro Term 64 (Download P/D)	39

EDUCATION 64

Spelling I (Koala) (D)	$ 29
Geometric (Koala) (D)	20
I.Q. Baseball (D)	25
Bible Baseball (D)	25
Happy Tutor (Typing) (D)	15

LANGUAGES 64

Instaspeed Basic Compiler (D)	$ 99
Nevada Cobol (D)	55
Pilot (D)	45
Logo (D)	45
Assembler Development (D)	25
64 Forth (R)	40

PRINTERS - DOT MATRIX

Epson RX80 (80 cps)	$ 299
MX80 w/FT (80 cps)	399
FX80 (160 cps)	549
FX100 (160 cps) 14" width	859
Okidata 82A	429
Okidata 92	549
NEC 8023A	429
Star Delta (160 cps)-NEW!	549
Star Gemini 10X (120 cps)	309
Star Gemini 10/15	Call
Transtar 315 (Hi Res., Color)	575
Micro Edge Printer Paper (540 Sheets)	10

ESSENTIALS

Commodore 64	$ Call
1541 Disk Drive	249
1525 Printer (80 col/DM)	225
1530 Datasette	65
1520 Plotter/Printer (4 Color)	169
1526 Printer	Call
1702 Monitor	249

CBM

8032 (80 column Pet)	$ 625
SuperPet (5 languages!)	1049
8050 Dual Drive (1 mg.)	995
8250 Dual Drive (2 mg.)	1295
9060 Hard Disk (5 mg.)	1995
2031, 170K Single Drive	295
64K Upgrade for 8032	259
SuperPet upgrade for 8032	599
4023 Printer (80 cps, 80 col)	395
8023 Printer (150 cps, graphics)	545
6400 Printer (40 cps, LQ)	1450

LETTER QUALITY PRINTERS

Diablo 620, 25 cps	$ 949
Transtar 130, 16 cps - 132 col	769
Transtar 120, 14 cps - 80 col	500

MONITORS

Panasonic CT 160 (color)	$ 279
Panasonic TR120 (w/spkr,green)	155
Sanyo/Amdek-Green, No Audio, 12"	125
BMC/Sanyo-Green, No Audio, 9"	95
Cable (For Above) A/V	15

VIC ACCESSORIES

8K RAM Expand. Cart	$ 40
16K RAM Expand. Cart	70
24K RAM Expand. Cart	105
27K RAM (Expands Vic to full 32K)	119
3 Slot Expander	30
6 Slot Expander	70
Joystick (Wico-Red)	25
Joystick Blaster (ADR Rapid-Fire)	10

INTERFACES & ACCESSORIES

Data 20 80-Col. Exp.	$ 159
Mr. Computer 80-Col. Exp.	60
5-Slot Exp. (64)	65
Vic Switch (connect 8 64's or Vic's to DD/Printer)	145
Cables 3M, 6M, 12M for above	Call
Verex (Box of 10) 5¼ Diskettes	26
Connection(Pet/64 graphics,2K Buffer)	99
Cardco Print + Graphics	85
Cardco Cardprint	70
MW 302 Parallel	65
PET/IEEE Cable (1m)	33
IEEE/IEEE Cable (1m)	49
Interpod (Intelligent IEEE, RS232, serial)	149
ADA 1800 (IEEE/Parallel)	129
ADA 1450 (IEEE/RS232 (M/F))	129

VISA/MASTERCARD
MONEY ORDERS
BANK CHECK

C.O.D.'s Accepted. (Add $5)
In stock items shipped within 48 hours.
F.O.B. Dallas, Texas (Texas Res., Add 5% Tax).
Products shipped with manufacturer's warranty.
Prices subject to change without notice.
$50 Minimum Order.

*Defective units **must have** return authorization number and include copy of invoice.

SJB DISTRIBUTORS INC.
10520 Plano Road, Suite 206
Dallas, Texas 75238

TO ORDER
CALL TOLL FREE
800-527-4893
800-422-1048
(Within Texas)

CATALOG
Send Postcard with Name & Address to speed processing.

NOTE: SJB HAS A FULL LINE OF COMPUTER MEDIA IN STOCK, CALL OR WRITE FOR MORE INFORMATION.

The program is controlled with the special function keys and the back-arrow key. Here's a quick rundown of what the keys will do.

f1– Pressing the f1 key scrolls forward through the VIC's memory. Each time you press f1, it will scroll about 40 bytes, or two lines of screen memory. Holding down the f1 key will continuously scroll through memory.

f3– Pressing the f3 key scrolls backward through the VIC's memory. Scrolling backward is also done in increments of 40 bytes, or two screen lines. Holding down f3 will scroll backward continuously.

f5– Pressing the f5 key displays the beginning address of the 505 bytes of memory you're currently looking at. When you press f5, you'll see a blinking red number in the upper-left corner of the screen. This is the address of the first byte of memory currently displayed on the screen. This byte is displayed as the character at the home (upper-left corner) position of the screen. Because the VIC's screen has 506 positions, the byte at the lower-right corner is this number plus 505.

Significantly, the characters you see displayed on the screen are actually the *screen (POKE) code* values of the bytes being displayed. For example, a byte that contains a value of zero will be displayed as @, a value of one as "a", and so forth. The screen codes can be found in Appendix H of the manual which came with your VIC.

f7– Pressing the f7 key resets the display to memory address 0. That is, pressing f7 resets the program to the beginning, and bytes 0 through 505 again will be displayed on the screen.

←– Pressing the ← key (back-arrow) key puts the program into the input mode and displays the prompt: *Enter desired address*. You can now enter any address you want to see displayed (between 0 and 65275) and press RETURN. The screen will then display the 505 bytes of memory starting at the address you specified.

The Guided Tour

Now, if you're done testing the various control keys, press f7 to reset the program, climb into the tour bus, and away we'll go.

The screen display you're now looking at (starting at 0) is memory bytes 0 through 505. On the upper half of the screen, slightly to the left, you'll see a byte changing quickly. This and the two bytes to the left are the *jiffy clock*. You're actually watching the VIC's jiffy clock as it is running. Now repeatedly press any key (except one of our program control keys) on the keyboard, and watch what happens. You should see two bytes change as you press the keys. These are the two bytes (197 and 203) that tell the operating system you have pressed a key. The value placed into these bytes will reflect the specific key you pressed. At the bottom of the screen, to the right, you'll see four bytes changing rapidly. These are a few of the memory locations in the *processor stack* area. This is where the operating system temporarily stores information such as return addresses when a program performs a GOSUB. The activity here is caused by the program running.

Now press the f1 key, and scroll up to address 1012 or so. Remember, to see what addresses you're looking at, press f5. If you don't have the 3K expansion RAM plugged into your VIC-20, you'll see a screen full of activity here. This is the area where the 3K expander fits in. Without a 3K expander, this is neither RAM nor ROM, and the activity you see is produced by the program reading spurious values.

Press f1 again and scroll up to 4092. If you pass the desired address, you can scroll backward by pressing f3. If you don't want to waste the time scrolling, press the back-arrow, then enter 4092. On the bottom half of the screen will be many one-, two-, and three-digit numbers. You're now looking at the user BASIC program area, and the numbers you see are part of the BASIC program that POKEd this machine language routine into memory.

As you scroll through the BASIC program area and on into screen memory (7680–8191), you'll get some interesting effects. The screen may turn blank, or it may continually change, showing strange displays. This is caused by a sort of reflective effect. You're looking at screen memory while you're displaying the same memory on the screen. In effect, the screen is echoing itself. This is the same type of effect you may see when you stand in front of a three-way mirror at your local clothing store. The mirrors reflecting each other give the illusion of your reflection going off into infinity.

As you continue scrolling forward from screen memory toward 32767, you'll see the top half of the screen display one character, and the bottom half another. This area (8192–32767) is for expansion RAM, three blocks of 8K each. What you see on the screen are the page numbers of this expansion area. A page is 256 continuous bytes. For example, memory locations 8192 through 8447 would be page 32, memory locations 8448 through 8703 would be page 33, and so forth.

Looking At High Memory

Now let's save some scrolling time. Press the back-arrow, then enter 32768. What you're looking at now is the beginning of character ROM, where the VIC gets its character information. Scroll a couple of pages up to 33802, and notice the change in the screen. The characters appear to be reversed. They are. You're looking at character ROM

(33792–33815) for reversed characters.

Press the back-arrow and enter 36864. You should now see many bytes changing. This is one of the more interesting areas of the VIC to look at—input/output block O. As you scroll forward and backward through the area (36864–38400), you'll observe a lot of different activity. This input/output block is continually updating what you see on the screen. It also handles other I/O such as disk drives, printers, etc. Here, too, you'll find timers, data direction registers, and control registers. This area of the VIC-20 is always active.

The next, and last, area of memory we'll look at is 49152–65535. Press the back-arrow and enter 49152. What you see now could be called the heart of the VIC-20. The first 8000 bytes (49152–57343) is BASIC ROM, where the operating system looks when it needs to interpret BASIC commands such as PEEK, POKE, PRINT, etc. The second 8000 bytes (57344–65535) is the Kernal ROM—the true center of the VIC. Whether you're programming in BASIC or machine language, this area holds the instructions that actually tell the VIC-20 how to do those PRINTs, PEEKs, and POKEs. Without this area of the VIC, or another operating system to take its place, your VIC-20 would be a paperweight—it couldn't add 2 plus 2.

The bus is now unloading, and we hope you enjoyed the tour. It should give you an even better idea of what's going on inside that remarkable little machine of yours.

VIC Window

```
10 POKE52,28:POKE56,28:POKE251,0:CLR
                                    :rem 111
20 B=7168:C=7348:X=7196:Y=256:Z=7197
                                     :rem 3
30 FORA=BTOC:READD:POKEA,D:NEXT     :rem 29
40 SYS7168                          :rem 57
50 PRINT"{HOME}{7 SPACES}";"{HOME}{RED}";
   PEEK(X)+Y*PEEK(Z);"{BLU}"        :rem 28
60 IFPEEK(251)=1THENGOSUB100       :rem 177
70 IFPEEK(197)=64THENGOTO40        :rem 128
80 GOTO50                           :rem 6
100 POKE251,0:PRINT"{CLR}{DOWN} ENTER DES
    IRED ADDRESS":POKE198,0         :rem 78
110 INPUTN                         :rem 112
120 NN=INT(N/256):POKEX,N-(NN*256):POKEZ,
    NN                             :rem 105
130 POKE7205,PEEK(X):POKE7206,PEEK(Z)+1
                                    :rem 27
150 RETURN                         :rem 118
7168 DATA 162,0,169,6,157,0,150,232
                                    :rem 189
7176 DATA 208,250,238,6,28,173,6,28
                                    :rem 204
7184 DATA 201,152,208,236,169,150,141,6
                                    :rem 132
7192 DATA 28,162,0,189,0,0,157,0   :rem 35
7200 DATA 30,232,208,247,189,0,1,157
                                    :rem 232
7208 DATA 0,31,232,208,247,165,197,201
                                    :rem 81
7216 DATA 64,240,230,162,0,232,208,253
                                    :rem 71
7224 DATA 201,39,208,37,24,173,28,28
                                    :rem 245
7232 DATA 105,22,141,28,28,173,29,28
                                    :rem 240
7240 DATA 105,0,141,29,28,24,173,37
                                    :rem 183
7248 DATA 28,105,22,141,37,28,173,38
                                    :rem 247
7256 DATA 28,105,0,141,38,28,76,25
                                    :rem 145
7264 DATA 28,201,47,208,37,56,173,28
                                    :rem 253
7272 DATA 28,233,22,141,28,28,173,29
                                    :rem 246
7280 DATA 28,233,0,141,29,28,56,173
                                    :rem 194
7288 DATA 37,28,233,22,141,37,28,173
                                    :rem 252
7296 DATA 38,28,233,0,141,38,28,76
                                    :rem 155
7304 DATA 25,28,201,55,208,1,96,201
                                    :rem 183
7312 DATA 63,208,19,169,0,141,28,28
                                    :rem 194
7320 DATA 141,29,28,141,37,28,169,1
                                    :rem 192
7328 DATA 141,38,28,76,25,28,201,8
                                    :rem 150
7336 DATA 240,3,76,25,28,169,1,133
                                    :rem 144
7344 DATA 251,96,0,0,0              :rem 51
```

If you've got questions or ideas about subjects you'd like to see covered in this column, write to: VICreations, COMPUTE!'s GAZETTE, *P.O. Box 5406, Greensboro, NC 27403.*

COMPUTE!'s Gazette Back Issues

JULY 1983: Commodore 64 Video Update, Snake Escape, Alfabug, VIC Marquee, Word Hunt, VIC Timepiece, product reviews, Learning To Program In BASIC, Quickfind, 64 Paddle Reader, Machine Language For Beginners, Enlivening Programs With Sound, Using Joysticks On The 64, Simple Answers To Common Questions, VICreations — Speedy Variables, 64 Explorer.

AUGUST 1983: Your First Hour With A Computer, Should You Join A Users Group, Guide To Commodore Users Groups, The Viper, Cylon Zap, product reviews, VIC/64 Mailing List, Word Spell, Global Scan For VIC/64, Machine Language For Beginners, VIC Title Screens, 64 Hi-Res Graphics Made Easy, VIC/64 Four-Speed Brake, Disk Menu, Using A 1540 Disk Drive With The 64, Playing Computer Music, Simple Answers To Common Questions, HOTWARE, VICreations — Caring For Disk Drives/Cassettes, 64 Explorer, News & Products.

SEPTEMBER 1983: Telecomputing Today, Telecomputing Glossary, Commodore's Nationwide Party Line, Commodore Bulletin Boards, Demon Star For VIC/64, Potholes, product reviews, Checkbook Reporter, States & Capitals Tutor For VIC/ 64, MiniTerm-20, TeleTerm 64, POKEing Graphics, Machine Language For Beginners, 64 Searcher, Better Commodore Input, Using The Function Keys, Simple Answers To Common Questions, HOTWARE, VICreations — Understanding Random Numbers.

OCTOBER 1983: The Anatomy of Computers, Telegaming Today And Tomorrow, Commodore's Public Domain Programs, Oil Tycoon, Re-Beep, product reviews, Aardvark Attack, Word Match, A SHIFTy Solution: The WAIT Command, Program Transfers, Machine Language For Beginners, Improved Paddle Reader Routine, How To Use Tape And Disk Files, Understanding 64 Sound — Part 1, Speeding Up The VIC, Simple Answers To Common Questions, HOTWARE, Horizons 64 — Improving 64 Video Quality, VICreations — Using The VIC's Clock, News & Products.

NOVEMBER 1983: Binary Numbers — Part 1, Getting Started With A Disk Drive — Part 1, Chicken Little, Martian Prisoner, product reviews, Munchmath, VIC Super Expander Graphics, 64 Aardvark Attack, 64 Timepiece, Connect The Dots, Custom Characters For VIC/64, Making Custom Characters On The 64, Making Custom Characters On The VIC, VIC/64 Program Lifesaver, Understanding 64 Sound — Part 2, Merging Programs On The 64, Tutorial On DATA, READ, RESTORE Statements, One-Touch Commands For The 64, VIC/64 Disk Defaulter, Machine Language For Beginners, Simple Answers To Common Questions, HOTWARE, VICreations — Animation With Custom Characters, Horizons 64 — Software And Hardware Reviews, News & Products, Automatic Proofreader.

Back issues of July, August, and September 1983 are $2.50 each. Issues from October forward are $3. Bulk rates are 6 issues for $15 or 12 issues for $30. All prices include freight in the U.S. Outside the U.S. add $1 per magazine order for surface postage. $4 per magazine for air mail postage. ALL BACK ISSUES ARE SUBJECT TO AVAILABILITY.

In the Continental U.S. call
TOLL FREE 800-334-0868
(in North Carolina call 919-275-9809)

Or write to:
COMPUTE!'s Gazette for Commodore Back Issues
P.O. Box 5406
Greensboro, North Carolina, 27403, USA

Prepayment required in U.S. funds. MasterCard, VISA, and American Express accepted. North Carolina residents please add 4% sales tax.

NEWS & PRODUCTS

Commodore 64 PILOT

An advanced PILOT language for the Commodore 64 is available from Tamarack Software.

PILOT II includes 25 editing commands, 19 turtle graphics commands, and 23 program commands, with several subcommands and options.

The program includes three graphics options—lo-res turtle, hi-res turtle, and sprites. A full-featured sprite editor also is included.

A STUDENT command allows teachers to design lessons and tests and keep the answers hidden from the students. A built-in timer can track the time a student works on a problem.

PILOT II is available on disk for $49.95.

Tamarack Software, Inc.
Box 247
Darby, MT 59829
(406) 821-4596

Games For The Commodore 64

Tronix has produced four new games for play on the Commodore 64. The games, *Waterline*, *Suicide Strike*, *Motocross*, and *Slalom*, will be available initially on disk with cartridge versions to follow.

In *Waterline*, the player, a ship's captain, must choose between saving the passengers of his sinking ship or salvaging the ship's store of gold.

In *Suicide Strike*, the player flies his plane through waves of enemy planes enroute to his military target. The pilot has a limited amount of time and fuel to complete the mission. The game features a rear-view mirror that allows the player to see enemy action behind him.

Slalom and *Motocross* are graphic representations of the real sports. The games attempt to reproduce the drama and excitement found in skiing and cycle riding.

Disk versions of these games sell for $34.95; cartridge versions will be available for $39.95.

Tronix Publishing, Inc.
8295 S. La Cienega Blvd.
Inglewood, CA 90301
(213) 215-0529

Memory Saver For VIC

A 16K VIC-20 expansion board with a built-in battery backup is available from Abaris.

The 16K Memory Plus includes full-block switching, a reset switch, and a write-protect switch. Also included is a Nickel-Cadmium battery backup circuit. Programs housed in the expander are retained for up to four weeks, even in the event of an unexpected power failure.

Once programs are loaded into memory, the expander can be removed and transported to another VIC. In addition, custom routines can be loaded into block 5 memory and executed automatically on power up.

The 16K Memory Plus is available for $89 plus $3 for shipping.

Abaris, Inc.
Box 2501
Vancouver, WA 98668
(206) 694-3455

Flight Simulator II *shows both the flight instrumentation and a panoramic view of the surrounding area.*

Flight Simulator For Commodore 64

Sublogic's *Flight Simulator II* puts you at the controls of a Piper 181 Cherokee Archer with full flight instrumentation and a panoramic view.

The program lets you practice takeoffs, landings, and even aerobatics. It features more than 80 airports. Day, dusk, and night flying modes are available, and

weather conditions are user adjustable.

The program, which is available for the Commodore 64 as well as Apple and Atari, also includes an air battle game that you can use to test your skills.

Flight Simulator II is available for $49.95 plus $1.50 for shipping.

Sublogic Corporation
713 Edgebrook Drive
Champaign, IL 61820
(800) 637-4983

Programs For Preschoolers

Kidbit Software has developed a line of software designed to be used by preschool children on the VIC-20.

Wormsical Count is a counting game. A worm crawls out of one of several apples on the ground and tries to make it across a field patrolled by hungry birds. Count the apples, count the birds, cheer the worm on to safety.

Small Wizard/Capital Wizard is a game that teaches the relationship between small and capital letters of the alphabet.

Same/Not Same Game teaches youngsters to match like colors, shapes, and letters. This game includes several play levels.

Alpha-Bee Sequence features a bee that sings the alphabet. When he gets stuck the child helps him along. When the alphabet is completed, the bee dances over a field of flowers to the tune of "Flight of the Bumblebee."

These programs are available on cassette for $9.95 each, two for $15.95, or four for $29.95.

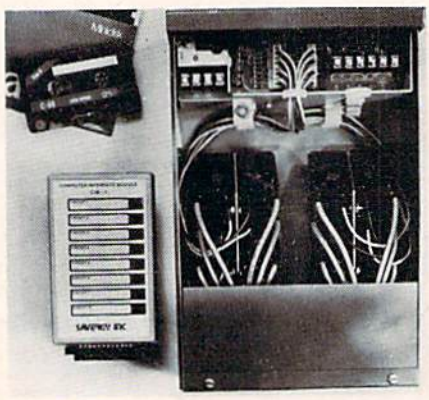
Kidbit Software
6116 Merced Ave. #175
Oakland, CA 94611

Home Computer-Appliance Interface

Savergy has produced an interface that links a Commodore 64 or VIC-20 to a variety of appliances to allow computer control of such things as lighting, heating and cooling, and lawn watering.

The Computer Interface Module 112 can communicate user-programmed information to one or more switching units having eight high- or low-voltage relays. The relay unit can be mounted up to 500 feet from the computer.

The package, which sells for $450, consists of the interface module, which plugs into the computer's User Port, a wall-mounted switching unit (about the size of a thick telephone book), and software on tape or disk.



Savergy's computer-appliance interface package consists of software, the interface module, left, and a switching unit, right.

Savergy, Inc.
1404 Webster Ave.
Fort Collins, CO 80524
(303) 221-4200

Games Converted For VIC

Sierra On-Line has converted nine of its action games for use on VIC-20 home computers.

Ultima II: Escape From Mt. Drash is an action/adventure game in which the player, a captive in the dungeons of Mt. Drash, attempts to escape through a maze of twisting subterranean corridors and tunnels. *Flip-N-Match* is a memory game in which players battle the clock while trying to match shapes. These games sell for $19.95.

In *Cannonball Blitz*, based on an American Revolution theme, the player tries to storm and overtake a Redcoat fortress. In *Jawbreaker*, players must eat their way through a horizontal maze of constantly moving walls. *Threshold* is a space shoot-em-up, complicated by overheating engines and the need to watch the fuel supply.

In *Crossfire*, the player must defend his city from aliens that approach from all directions. *Lunar Leepers* is a rescue game set on the moon. The player must save his men while avoiding the jaws of the voracious leepers. *Creepy Corridors* is a hunt for

diamonds in twisting passageways filled with crawling creatures of all sorts. These games are available for $29.95.

In *Frogger*, the player must help a frog across a busy highway and a dangerous river enroute to the peace and quiet of his home. It sells for $34.95.

Sierra On-Line, Inc.
Sierra On-Line Building
Coarsegold, CA 93614
(209) 683-6858

Software For Photographers

Shutterbug 64 is a software package designed for those who want to combine the hobbies of computing and photography.

The program, available from Quality Input, includes modules that allow the photographer to obtain detailed information on film selection, film processing, and film characteristics. The program also can create, maintain, and display photo and equipment inventory files on screen or printer.

Shutterbug 64 is available for $79.95.

Quality Input also produces several Commodore 64 programs for educational environments. They include: *Q-Stat*, a sophisticated program to assist in statistical research, $269.95; *ACT/SAT Review Package*, a program to help prepare students for college entrance examinations, $799.95; *Student Scheduler*, a program to assign student schedules in junior high and high schools, $299.95; *Student Filer*, a student record program for school administrators, $149.95; and *Learning in Spanish*, a program written in Spanish for Hispanic students, $119.95.

Quality Input, Inc.
309 West Beaufort, Suite 8
Normal, IL 61761
(309) 454-1061

Commodore 64 Tutorial

Cyberia has released the second volume of its *Commodore 64 Tutorial Series*. This program, available on disk, teaches the fundamentals of computer sound and graphics using audiovisual aids and an interactive approach.

The program, which sells for $24.95, includes a sound generator and a sprite editor, and makes extensive use of quizzes to reinforce subject matter.

Cyberia also has released *Galactic Battles*, a new Commodore 64 game. The game involves an explorer ship traveling through an alien universe dominated by robots which control time, energy, and matter. The game, which is available on disk or tape for $24.95, includes three different scenarios and ten screens.

Cyberia, Inc.
Box 784
Ames, IA 50010
(515) 292-7634

Card File For Commodore 64

Info-Manager from Pyramid Software International is an electronic index card file for the Commodore 64 designed for home or small business use.

The program, a data base, was written to resemble a traditional card file. Each record is referred to as a card, and the user has the options of searching through them or sorting them in ascending or descending order.

Command options are presented in menu format. The print option allows printing of all or part of the information in each record. In addition, any two lines of information in the record can be printed as one line.

Info-Manager is available on tape or disk for $39.95 plus $2.50 for shipping.

Another program available from Pyramid is *Memory Twister*, a computer version of the TV game show *Concentration*. *Memory Twister* comes on tape or disk for $18.95 plus $2.50 for shipping.

Pyramid Software International
30-A Fairfax St.
San Rafael, CA 94901
(415) 459-1333

COMPUTE!'s GAZETTE *welcomes announcements of new products for VIC-20 and Commodore 64 computers, especially products aimed at beginning to intermediate users. Please send press releases and photos well in advance to: Tony Roberts, Assistant Managing Editor,* COMPUTE!'s GAZETTE, *P.O. Box 5406, Greensboro, NC 27403.*

New product releases are selected from submissions for reasons of timeliness, available space, and general interest to our readers. We regret that we are unable to select all new product submissions for publication. Readers should be aware that we present here some edited version of material submitted by vendors and are unable to vouch for its accuracy at time of publication.

A Beginner's Guide To Typing In Programs

What Is A Program?

A computer cannot perform any task by itself. Like a car without gas, a computer has *potential*, but without a program, it isn't going anywhere. Most of the programs published in *COMPUTE!'s Gazette* for Commodore are written in a computer language called BASIC. BASIC is easy to learn and is built into all VIC-20s and Commodore 64s.

BASIC Programs

Each month, *COMPUTE!'s Gazette* for Commodore publishes programs for both the VIC and 64. To start out, type in only programs written for your machine, e.g., "VIC Version" if you have a VIC-20. Later, when you gain experience with your computer's BASIC, you can try typing in and converting certain programs from another computer to yours.

Computers can be picky. Unlike the English language, which is full of ambiguities, BASIC usually has only one "right way" of stating something. Every letter, character, or number is significant. A common mistake is substituting a letter such as "O" for the numeral "0", a lowercase "l" for the numeral "1", or an uppercase "B" for the numeral "8". Also, you must enter all punctuation such as colons and commas just as they appear in the magazine. Spacing can be important. To be safe, type in the listings *exactly* as they appear.

Brackets And Special Characters

The exception to this typing rule is when you see the curved bracket, such as "{DOWN}". Anything within a set of brackets is a special character or characters that cannot easily be listed on a printer. When you come across such a special statement, refer to "How To Type In *COMPUTE!'s Gazette* Programs."

About DATA Statements

Some programs contain a section or sections of DATA statements. These lines provide information needed by the program. Some DATA statements contain actual programs (called machine language); others contain graphics codes. These lines are especially sensitive to errors.

If a single number in any one DATA statement is mistyped, your machine could "lock up," or "crash." The keyboard and STOP key may seem "dead," and the screen may go blank. Don't panic – no damage is done. To regain control, you have to turn off your computer, then turn it back on. This will erase whatever program was in memory, *so always SAVE a copy of your program before you RUN it*. If your computer crashes, you can LOAD the program and look for your mistake.

Sometimes a mistyped DATA statement will cause an error message when the program is RUN. The error message may refer to the program line that READs the data. *The error is still in the DATA statements, though.*

Get To Know Your Machine

You should familiarize yourself with your computer before attempting to type in a program. Learn the statements you use to store and retrieve programs from tape or disk. You'll want to save a copy of your program, so that you won't have to type it in every time you want to use it. Learn to use your machine's editing functions. How do you change a line if you made a mistake? You can always retype the line, but you at least need to know how to backspace. Do you know how to enter inverse video, lowercase, and control characters? It's all explained in your computer's manuals.

A Quick Review

1) Type in the program a line at a time, in order. Press RETURN at the end of each line. Use backspace or the back arrow to correct mistakes.

2) Check the line you've typed against the line in the magazine. You can check the entire program again if you get an error when you RUN the program.

3) Make sure you've entered statements in brackets as the appropriate control key (see "How To Type *COMPUTE!'s Gazette* Programs" elsewhere in the magazine.)

We regret that we are not able to respond to individual inquiries about programs, products, or services appearing in COMPUTE!'s Gazette *for Commodore due to increasing publication activity. On those infrequent occasions when a published program contains a typo, the correction will appear in the magazine, usually within eight weeks. If you have specific questions about items or programs which you've seen in* COMPUTE!'s Gazette *for Commodore, please send them to Gazette Feedback, P.O. Box 5406, Greensboro, NC 27403.*

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, ⟨⟩, 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:
{CLEAR}	SHIFT + CLR/HOME	
{HOME}	CLR/HOME	
{UP}	SHIFT + CRSR ↕	
{DOWN}	CRSR ↕	
{LEFT}	SHIFT + CRSR ↔	
{RIGHT}	CRSR ↔	
{RVS}	CTRL + 9	
{OFF}	CTRL + 0	
{BLK}	CTRL + 1	
{WHT}	CTRL + 2	
{RED}	CTRL + 3	
{CYN}	CTRL + 4	
{PUR}	CTRL + 5	
{GRN}	CTRL + 6	
{BLU}	CTRL + 7	
{YEL}	CTRL + 8	
⟨1⟩	C= + 1	
⟨2⟩	C= + 2	
⟨3⟩	C= + 3	
⟨4⟩	C= + 4	
⟨5⟩	C= + 5	
⟨6⟩	C= + 6	
⟨7⟩	C= + 7	
⟨8⟩	C= + 8	
{F1}	f1	
{F2}	SHIFT + f1	
{F3}	f3	
{F4}	SHIFT + f3	
{F5}	f5	
{F6}	SHIFT + f5	
{F7}	f7	
{F8}	SHIFT + f7	

The Automatic Proofreader

"The Automatic Proofreader" will help you type in program listings from COMPUTE!'s Gazette without typing mistakes. It is a short error-checking program that hides itself in memory. When activated, it lets you know immediately after typing a line from a program listing if you have made a mistake. Please read these instructions carefully before typing any programs in COMPUTE!'s Gazette.

Preparing The Proofreader

1. Using the listing below, type in the Proofreader. The same program works on both the VIC-20 and Commodore 64. Be very careful when entering the DATA statements — don't type an l instead of a 1, an O instead of a 0, extra commas, etc.

2. SAVE the Proofreader on tape or disk at least twice *before running it for the first time.* This is very important because the Proofreader erases this part of itself when you first type RUN.

3. After the Proofreader is SAVEd, type RUN. It will check itself for typing errors in the DATA statements and warn you if there's a mistake. Correct any errors and SAVE the corrected version. Keep a copy in a safe place — you'll need it again and again, every time you enter a program from COMPUTE!'s Gazette.

4. When a correct version of the Proofreader is RUN, it activates itself. You are now ready to enter a program listing. If you press RUN/STOP-RESTORE, the Proofreader is disabled. To reactivate it, just type the command SYS 886 and press RETURN.

Using The Proofreader

All VIC and 64 listings in COMPUTE!'s Gazette now have a *checksum number* appended to the end of each line, for example ":rem 123". *Don't enter this statement when typing in a program.* It is just for your information. The rem makes the number harmless if someone does type it in. It will, however, use up memory if you enter it, and it will confuse the Proofreader, even if you entered the rest of the line correctly.

When you type in a line from a program listing and press RETURN, the Proofreader displays a number at the top of your screen. *This checksum number must match the checksum number in the printed listing.* If it doesn't, it means you typed the line differently than the way it is listed. Immediately recheck your typing. Remember, don't type the rem statement with the checksum number; it is published only so you can check it against the number which appears on your screen.

The Proofreader is not picky with spaces. It will not notice extra spaces or missing ones. This is for your convenience, since spacing is generally not important. But occasionally proper spacing *is* important, so be extra careful with spaces, since the Proofreader will catch practically everything else that can go wrong.

There's another thing to watch out for: if you enter the line by using abbreviations for commands, the checksum will not match up. But there is a way to make the Proofreader check it. After entering the line, LIST it. This eliminates the abbreviations. Then move the cursor up to the line and press RETURN. It should now match the checksum. You can check whole groups of lines this way.

Special Tape SAVE Instructions

When you're done typing a listing, you must disable the Proofreader before SAVEing the program on tape. Disable the Proofreader by pressing RUN/STOP-RESTORE (hold down the RUN/STOP key and sharply hit the RESTORE key). This procedure is not necessary for disk SAVEs, *but you must disable the Proofreader this way before a tape SAVE.*

SAVE to tape erases the Proofreader from memory, so you'll have to LOAD and RUN it again if you want to type another listing. SAVE to disk does not erase the Proofreader.

Replace Original Proofreader

If you typed in the original version of the Proofreader (October 1983 issue), you should replace it with the improved version below. We added a POKE to the original version to protect it from being erased when you LOAD another program from tape. The POKE does protect the Proofreader, and the Proofreader itself was not affected. However, a quirk in the VIC-20's operating system means that programs typed in with the Proofreader and SAVEd on tape cannot be LOADed properly later. If you LOAD a program SAVEd while the Proofreader was in memory, you see ?LOAD ERROR. This applies only to VIC tape SAVEs (disk SAVEs work OK, and the quirk was fixed in the Commodore 64).

If you have a program typed in with the original Proofreader and SAVEd on tape, follow this special LOAD procedure:

1. Turn the power off, then on.
2. LOAD the program from tape (disregard the ?LOAD ERROR).
3. Enter: POKE 45,PEEK(174):POKE 46,PEEK(175):CLR
4. ReSAVE the program to tape.

The program will LOAD fine in the future. We strongly recommend that you type in the new version of the Proofreader and discard the old one.

Automatic Proofreader For VIC And 64

```
100 PRINT"{CLR}PLEASE WAIT...":FORI=886TO
    1018:READA:CK=CK+A:POKEI,A:NEXT
110 IF CK<>17539 THEN PRINT"{DOWN}YOU MAD
    E AN ERROR":PRINT"IN DATA STATEMENTS.
    ":END
120 SYS886:PRINT"{CLR}{2 DOWN}PROOFREADER
     ACTIVATED.":NEW
886 DATA 173,036,003,201,150,208
892 DATA 001,096,141,151,003,173
898 DATA 037,003,141,152,003,169
904 DATA 150,141,036,003,169,003
910 DATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,251
958 DATA 003,206,251,003,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 151,003,192,087,208,006
994 DATA 032,205,189,076,235,003
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 251,003,133,214,076,173
1018 DATA 003
```

MLX Machine Language Entry Program

For Commodore 64 And VIC-20

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost failsafe entry of machine language programs published in COMPUTE!'s GAZETTE. You need to know nothing about machine language to use MLX—it was designed for everyone. There are separate versions for the Commodore 64 and expanded VIC-20 (at least 8K). MLX was conceived and written by Program Editor Charles Brannon. Important: MLX is required to type in the machine language programs in this issue.

MLX is a new way to enter long machine language (ML) programs with a minimum of fuss. MLX lets you enter the numbers from a special list that looks similar to BASIC DATA statements. It checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers. It won't let you enter numbers greater than 255 (forbidden in ML). It won't let you enter the wrong numbers on the wrong line. In addition, MLX creates a ready-to-use tape or disk file. You can then use the LOAD command to read the program into the computer, as with any program:

```
LOAD "filename",1,1   (for tape)
LOAD "filename",8,1   (for disk)
```

To start the program, you enter a SYS command that transfers control from BASIC to machine language. The starting SYS number always appears in the appropriate article.

Using MLX

Type in and save the correct version of MLX for your computer (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. MLX asks you for two numbers: the starting address and the ending address. These numbers are given in the article accompanying the ML program.

You'll see a prompt corresponding to the starting address. The prompt is the current line you are entering from the listing. It increases by six each time you enter a line. That's because each line has seven numbers—six actual data numbers plus a *checksum number*. The checksum verifies that you typed the previous six numbers correctly. If you enter any of the six numbers wrong, or enter the checksum wrong, the computer rings a buzzer and prompts you to reenter the line. If you enter it correctly, a bell tone sounds and you continue to the next line.

MLX accepts only numbers as input. It you make a typing error, press the INST/DEL key; the entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as listed, the computer automatically prints the comma and goes on to accept the next number. If you enter less than three digits, you can press either the comma, SPACE bar, or RETURN key to advance to the next number. The checksum automatically appears in inverse video for emphasis.

MLX Commands

When you finish typing an ML listing (assuming you type it all in one session) you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you've made a typo when entering the MLX program itself.

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later. MLX recognizes these commands:

SHIFT-S: Save	**SHIFT-N: New Address**
SHIFT-L: Load	**SHIFT-D: Display**

When you enter a command, MLX jumps out of the line you've been typing, so we recommend you do it at a new prompt. Use the Save command to save what you've been working on. It will save on tape or disk as if you've finished, but the tape or disk won't work, of course, until you finish the typing. Remember what address you stop at. The next time you run MLX, answer all the prompts as you did before, then insert the disk or tape. When you get to the entry prompt, press SHIFT-L to reload the partly completed file into memory. Then use the New Address command to resume typing.

To use the New Address command, press SHIFT-N and enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the special listing, or else the checksum won't work. The Display command lets you display a section of your typing. After you press SHIFT-D, enter two addresses within the line number range of the listing. You can abort the listing by pressing any key.

The special MLX commands may seem a bit confusing, but as you work with MLX, they will become valuable. For example, what if you forgot where you stopped typing? Use the Display command to scan memory from the beginning to the end of the program. When you reach the end of your typing, the lines will contain a random pattern of numbers. When you see the end of your typing, press any key to stop the listing. Use the New Address command to continue typing from the proper location.

You can use the Save and Load commands to make copies of the completed program. Use Load to reload the tape or disk, then insert a new tape or disk and use Save to make a new copy.

Be sure to save MLX; it will be used for future ML programs in COMPUTE!'s GAZETTE.

See program listings on page 184.

Bug-Swatter:
Modifications And Corrections

● If you used MLX to type in "Spike"(December), you may have had a problem as portions of your typing seemed to change as you entered the program. A solution is the POKE statement below, which will move the top of BASIC memory to below the Spike program, so BASIC does not write over Spike as you type it in.

```
POKE 52, 128: POKE 56, 128: CLR
```

Enter this line in direct mode (without a line number). Then LOAD and RUN MLX. Next Spike should be loaded in. By using the Display command of MLX, you can check to see which parts of Spike have been overwritten by BASIC. You can then determine which sections you need to retype.

Additionally, there is a correction in the text of the MLX article on page 164. The article mentions that by scanning memory from the beginning to the end of the program, the memory locations where you have not typed in numbers will be filled with 170s. This is incorrect. These areas will be filled with random patterns of numbers.

●"64 Basic Aid" (January) is subject to the same problem, as described above. Before typing it in, you must protect the top of memory with POKE 52,154: POKE 56,154: CLR. Without these POKEs, BASIC will overwrite the machine language program.

●Tape users have had problems using "Automatic Proofreader" with programs typed in more than one sitting (or after a safety SAVE). Automatic Proofreader is a machine language program stored in the cassette buffer, and when a program is SAVEd or LOADed from tape, the buffer is cleared. This makes it impossible to reload that part of the program you had previously entered and saved, and Automatic Proofreader at the same time.

The following modification will allow you to load Automatic Proofreader while a program is in memory:

1. LOAD and RUN Automatic Proofreader. This will put the machine language program into the cassette buffer.

2. Type the following lines in direct mode (without line numbers):

```
A$="PROOFREADER.T": B$="{10 SPACES}": FOR
   X = 1 TO 4: A$=A$+B$:NEXTX
FORX= 886 TO 1018: A$ = A$ + CHR$(PEEK(X)
  ):NEXTX
OPEN1,1,1,A$:CLOSE1
```

After you type the last line, you will be asked to press RECORD and PLAY. We recommend that you start at the beginning of a new tape.

You now have a new version of Automatic Proofreader. Turn your computer off and on, then LOAD the program you were working on. Put the cassette containing PROOFREADER.T into the tape drive and type:

```
OPEN1:CLOSE1
```

You can now get into Proofreader by typing SYS 886. To test this, PRINT PEEK (886) should return the number 173. If it does not, repeat the steps above, making sure that A$ (PROOFREADER.T) contains 13 characters and that B$ contains 10 spaces.

The new version of Automatic Proofreader will load itself into the cassette buffer whenever you type OPEN1: CLOSE1 and PROOFREADER.T is the next program on your tape. It will not disturb the contents of BASIC memory.

The lines above convert the machine language program into characters that are concatenated into a string. When you open a tape file, using the string as the name of the file, the tape header contains the ML program (disguised as the name of the file). Opening and closing the tape file loads the header into the cassette buffer, but does not disturb BASIC programs that are already in memory.

●A programming error in "64 Aardvark Attack" (November) prevents players from defending against bombs falling on city zero. To remedy this, readers James V. Powell and Sheldon S. Cantor suggest changing line 460:

```
460 IFVAL(G$)-1<>ZANDG$<>"0"ORZ<>9)THEN4
    80
```

The error was introduced when the game was translated from the VIC to the Commodore 64. The VIC-20 version (October) does not contain this problem.

●There are two typos in the Assembler program ("Machine Language For Beginners," November): In line 6060, change ZO$ to ZA$. And, in line 20000, change BCC8114 to BCC8144. Thanks to reader Jim Tobias for pointing this out.

PRODUCT MART

Numeric key pad for Commodore VIC-20 and 64

With full cursor control and special function keys. No software interaction.

$59.95 Retail

Bear Technologies INC.
4321 Airwest S.E.
Grand Rapids, MI 49508
(616) 698-5000
VISA and Mastercard accepted

---

for Commodore 64™ and VIC 20™
Creative and hilarious FUN HOUSE

Full of Surprises!

- Great as a gift and for having fun at parties
- Has color graphics fun sounds & accepts words
- Find special keys for extra fun

On cassette. Just $22. Get $2. off with the word "GAZETTE" in your order. Specify computer. Send check or money order to:

CONTINENTAL CONCEPTS
P.O.Box 7631 NEWPORT BEACH, CA 92660

CA add sales tax. Postage extra if fast delivery desired.
VIC 20 version uses 16K expansion. Order today!

"VIC 20" and "Commodore 64" are trademarks of Commodore Business Machines

---

MAC PAC

MACPAC is a full-featured package for machine language programming on the VIC-20.™

- Memory expansion NOT required.
- User Guide includes introduction to 6502 Assembly Language.

Available for $25 -- specify tape or disk -- or write for free brochure.

LOW CLASS ENTERPRISES
777 KAPIOLANI BLVD.
SUITE 3301
HONOLULU, HI 96813

DEVELOP YOUR CHILD'S SPELLING & MATH SKILLS!

Make the drudgery of learning fun and easy with Spelling Tutor II and Math Tutor II. A unique way to advance your child's ability in learning.

MATH TUTOR II

- Menu Driven
- Addition/Subtraction
- Multiplication/Division
- Tape $15.95
- Disk $16.95

SPELLING TUTOR II

- Menu Driven
- Data Base File
- Create Spelling Words from Student's Spelling Lesson
- Tape $17.95
- Disk $19.95

Expansion Unit Unnecessary
User Friendly
VIC-20 CHECK OR MONEY ORDER C-64
IL RESIDENTS ADD 5% SALES TAX

SPH SOFTWARE
R.R. #1
E. PEORIA, IL 61611

"SLOT MACHINE for COMMODORE 64!"*

NO JOYSTICK NEEDED!

GREAT for PARTIES!

by:

VEGAS STYLE FUN!

"MA & PA SOFTWARE, ETC."

SATISFACTION AND QUALITY ALWAYS GUARANTEED!!

This program is intended for amusement only. "Actual" gambling, or awarding of prizes may be unlawful.

Please indicate cassete, or disk.
SEND $10.00 CHECK OR MONEY ORDER TO:
MA & PA SOFTWARE
672 Ave. De La Plata
Newbury Park, CA 91320
California residents add 6%.
*Trademark of Commodore Electronics LTD.

GUITAR PROGRAM

For your Commodore 64!!!
No musical experience needed.
35 common chords - drum and cymbal keys - transpose up and down. UNLOCKED SOFTWARE - modify it all you want! **Send $12.00.**
SPECIFY disk or tape:
DREWS PROGRAMS
3120 Corona Trail #206
Boulder, Colorado 80301

BridgePro®

Enjoy the card game of Bridge by yourself — your computer will play the other hands.

- Easy to learn — illegal bids and plays prevented
- Cards dealt randomly — millions of different hands
- Fast machine language speed
- 2-player game options
- Complete Contract Bridge scoring
- Bidding "help" feature for beginners
- Save the score and continue later
- May repeat hands, if desired
- Option to receive the best hand
- Play "duplicate" with a friend
- Demonstration feature plays all 4 hands

Available for Commodore 64* — Diskette $35

California residents add 6.5% tax

Computer Management Corporation
2424 Exbourne Court
Walnut Creek, CA 94596

VIC & 64

AT LAST! The information you need, without always going back to the manual. These durable plastic coated overlays contain program starting locations, function key labeling, commands and additional aids in center cutout.

Please send me the following Leroy's Cheatsheet™ keyboard overlays

20	64		20	64	
☐	☐	Programmer's Aid[1]	☐	☐	Graphic printer (1515 & 1525)[1]
☐	☐	Vicmon[1]	☐	☐	UMI Wordcraft 20[2]
☐	☐	Super Expander[1]	☐	☐	HES Vic Forth[3]
☐	☐	Vic Typewriter[1]	☐	☐	HES Writer[3]
☐	☐	Victerm 1[1]	☐	☐	Wordpro 3 plus
☐	☐	Term 64[1]	☐	☐	Easy Script[1]
☐	☐	Quick Brown Fox	☐	☐	Basic

CG184

Send check or money order plus $1.00 (postage and handling)
PA residents add 6% sales tax.

Name _____
Address _____
City _____ State _____ Zip _____

(1) Products of Commodore Business Machines, Inc. (2) Product of United Microware Industries, Inc.
(3) Products of Human Engineered Software. VIC-20 is a Trademark of Commodore Business Machines, Inc.

CHEATSHEET PRODUCTS™
P.O. Box 8299 Pittsburgh PA. 15218 (412) 456-7420

COMMODORE 64

VIC-20

	C-64 Tape	C-64 Disk
BASIC TUTOR part 1 . . . . . Learn BASIC programming from your computer, not books and manuals.	$14.95*	$17.95*
TRIVIA BOWL I . . . . . Over 1,000 questions; fast-paced game.	N/A	14.95*
GAMES PACKAGE I—CASINO 3 games for the price of 1—Blackjack, Craps, Poker.	9.95*	12.95*
MAILING LIST—64 . . . . . Over 500 names; alpha search & retrieval.	N/A	9.95
AMORTIZATION	9.95*	12.95*
CHARACTER GENERATOR	9.95*	12.95*
SPRITE CREATOR—64	9.95	12.95

*denotes programs also available for VIC-20 with at least 8K expansion memory.

Add $1.50 per order shipping & handling.
Arkansas residents add 4% state sales tax.

Spottsware

P.O. Box 1912 • Little Rock, AR 72203

Commodore 64 and VIC 20 are registered trademarks of Commodore Electronics, Ltd.

CONVERSE WITH YOUR COMPUTER

AT LAST! A FULL IMPLEMENTATION of the original ELIZA program is now available to run on your Commodore 64!

Created at MIT in 1966, ELIZA has become the world's most celebrated artificial intelligence demonstration program. ELIZA is a non-directive psychotherapist who analyzes each statement as you type it in and then responds with her own comment or question – and her remarks are often amazingly appropriate!

Designed to run on a large mainframe, ELIZA has never before been available to personal computer users except in greatly stripped down versions lacking the sophistication which made the original program so fascinating.

Now, our new Commodore 64 version possessing the **FULL** power and range of expression of the original is being offered at the introductory price of only $25. And if you want to find out how she does it (or teach her to do more) we will include the complete **SOURCE PROGRAM** for only $20 additional.

Order your copy of ELIZA today and you'll never again wonder how to respond when you hear someone say, "Okay, let's see what this computer of yours can actually do!"

ELIZA IS AVAILABLE IN THE FOLLOWING FORMATS:
(Please specify Disk or Cassette)

1. Protected Version $25
(Protected Version can be run but not listed or modified)
2. Un-protected Commodore 64 BASIC Source Version $45
(Source Version can be listed and modified as well as run)
Both versions include a six page user manual.

Please add $2.00 shipping and handling to all orders
(California residents please add 6% sales tax)

ARTIFICIAL INTELLIGENCE RESEARCH GROUP
921 North La Jolla Avenue, Dept. G
Los Angeles, CA 90046
(213) 656-7368 (213) 654-2214
MC, VISA and checks accepted

VISA MasterCard

FREE OFFER!

COMPUTER CASSETTES 58¢

FREE
"Party Trivia Game"
with each order of
20 or more C-10's

Specify VIC-20 or Commodore 64

- C-10 Length
- 5 Screw Shell/Free Labels
- Lifetime money back guarantee
- Storage Box add 12¢ each
- $2.00 shipping charge — any quantity (Canadian orders $4.00 shipping)
- NJ Residents add 6% sales tax
- Send check or money order to

PARALLEL SYSTEMS
Box 772
Blackwood, NJ 08012
609-227-9634

FAMILY TREE

A NEW COMPUTER SOFTWARE PACKAGE TO HELP THE AMATEUR AND PROFESSIONAL GENEALOGIST USE THE COMMODORE 64 OR VIC AS A DYNAMIC SYSTEM TO CONTROL DATA ON THE FAMILY TREE.

FEATURES:

- 664 NAMES PER DATA DISK
- FULLY INDEXED
- EASY EDITING AND UPDATING
- SEARCH FUNCTIONS
- PRODUCES FAMILY GROUP SHEETS
- PRODUCES PEDIGREE CHARTS
- OUTPUT TO SCREEN OR PRINTER
- COMPLETE MANUAL

BY HELPING YOU TO ORGANIZE YOUR FAMILY TREE IT WILL AID YOU IN DETERMINING THE AREAS OF YOUR RESEARCH

PRICE $39.95 US — $49.95 CANADIAN (MICHIGAN & ONTARIO RESIDENCE ADD TAX)

GENEALOGY SOFTWARE
PHONE 519-344-3990

P.O. BOX 1151
PORT HURON, MICHIGAN 48061

1046 PARKWOOD AVE
SARNIA, ONTARIO N7V 3T9

VIC & 64

BE A COPY C.A.D.
(CASSETTE AIDED DUPLICATOR) NOW YOU CAN MAKE BACKUP COPIES OF ALL THE COSTLY, NON-SAVEABLE CASSETTE PROGRAMS YOU BOUGHT.

OUR BACKUP V1.0 UTILITY PROGRAM WILL LET YOU MAKE DUPLICATES THAT RUN.

BACKUP V1.0 WILL WORK WITH A STANDARD 5K UNEXPANDED VIC. MEMORY EXPANSION IS REQUIRED TO COPY PROGRAMS LONGER THAN 3K BYTES.

$24.95
PLUS $2.00 SHIPPING & HANDLING

SOFTWARE PLUS
6201 SUITE C
GREENBACK LANE
CITRUS HEIGHTS, CA 95610

VISA, MASTERCARD, AND MONEY ORDERS
CA. RESIDENTS ADD 6% SALES TAX.

VIC IS A TRADEMARK OF COMMODORE

916-726-8793

MICRO-CONSOLE

Route 5 Box 925V Canyon Lake, Texas 78130
(512) 964-2390

COMMODORE OWNERS . . . . GET ORGANIZED!!!
Micro-console puts it all together

- Holds Keyboard
- Holds Monitor
- Holds 2 Disk Drives or Recorders
- Hides Cable Mess
- Heavy Guage Aluminum Console Reduces Interference Between Peripherals
- Attractive Finish Matches Vic and 64

Models available for most popular Home Computers
Dealer Inquiries Invited

EXPAND YOUR VIC 20™ MEMORY!

Affordable Memory Expander lets you add 2K RAM Circuits as your needs increase. (Up to 35K.)

Mother Board with Instructions	$39.95
Mother Board & Sockets with Instructions	$54.95
Complete Kit with Cabinet (35K)	$159.95
Assembled & Tested Expander (35K)	$189.95
Each Add'l 2K Chip (Up to 280K paged RAM) [$1 Shipping]	$8.95

Send Check or Money Order to

PERIPHERAL DEVELOPMENT
P. O. Box 28247
St. Louis, MO 63132

(Add $5.5 % Missouri residence add 5½% tax)
Prices subject to change without notice

Vic 20 is a trademark of Commodore Electronics, Ltd.

Soft Cellars

PRESENTS

GAMES AND UTILITIES FOR THE VIC & 64
on cassette or disk

Digital Derby—Pari-mutuel betting with galloping graphics and sound.
Unex. Vic Cn1 . . . . . $14.95

High Risk—A possible mission fraught with audio, visual & mental gymnastics. Joystick required.
Unex. Vic only $19.95

Super Cipher—Decipher color or symbol codes. Select length and time. Infinite levels of difficulty. 1 or 2 players.
Unex. Vic Cn1 . . . . . $12.95

Program Cellar—Pixel-by-pixel movement techniques in BASIC. Auto renumber delete. Easy entry BASIC program lines. Sub-routine library.
Vic Cn1 . . . . . $14.95

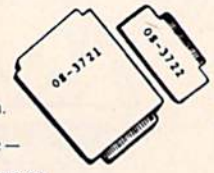
Data Cellar—Over 600 records per disk. Random access. Menu prompted. Alpha-numeric sorts. Easily tailored to your needs.
Cn1 only. Disk required $19.95

SEND CHECK OR MONEY ORDER TO:
SOFT CELLARS, INC.
828 RUE ROYAL SUITE 535
NEW ORLEANS, LA. 70116

ADD $2.00 for disk versions

Vic Cn1 are Commodore trademarks

MAKE YOUR OWN CARTRIDGES!



- Complete System • Programs 8K Cartridge to "AUTO-RUN" (opt) at Power-up — YOUR BASIC or Machine-Code Prgm.
- Prototype GAMES
- EDUCATIONAL use — NO Load Delay
- OS-3721 plugs into VIC-20
- Expansion socket on board
- Carts. Erasable & BLK Locatable

OS-3723 ZIF socket module Allows programming 2764, 2732A, 2732, 2716, 2564, 2532, 2516 EPROMS	$24.50
OS-37A System (Pgmr., 8K Cart)	$88.50
OS-37B System (Pgmr., ZIF mod) (Software Tape incl. Disk add $2)	$88.50
OS-3722 Blank 8K Carts.	$24.50
Blank 2764 EPROMS	$11.50

OTTO SYSTEMS
8135 ENGINEER ROAD
SAN DIEGO, CA 92111
(619) 569-5665

Add $4 Shipg. & Hndlg. — CA res. 6% tax

VIC-20 USERS
CARTRIDGE BACK-UP

- SYSTEM IS AN EASY TO USE PROGRAM AND A HIGH QUALITY CARTRIDGE INTERFACE BOARD
- BACK-UP YOUR CARTRIDGES ONTO CASSETTE OR DISK
- 8K RAM REQUIRED

$49.95 POST PAID

CASSETTE BACK-UP

- EASY TO USE PROGRAMS
- BACK-UP ANY CASSETTE TAPE ONTO TAPE OR DISK
- REQUIRES **NO USER MEMORY**

$19.95 POST PAID

NOW CARRYING OTHER EQUIPMENT
PLEASE CALL OR WRITE FOR PRICE LIST OF COMMODORE COMPATIBLE EQUIPMENT AND SOFTWARE

VISA/MASTERCARD ORDERS:
PHONE (215) 269-4803

MAIL-CHECK OR MONEY ORDER TO:
E-M TECHNOLOGIES
P.O. BOX 185
DOWNINGTOWN, PA 19335

PA. RESIDENTS ADD 6%
6 MONTH REPLACEMENT GUARANTEE

VIC 20™
KIDBIT SOFTWARE
PRESENTS:

THE PLAYSCHOOL TAPES

Pre-school learning programs, on cassettes for the unexpanded VIC

- ☐ WORMSICAL COUNT
- ☐ SAME/NOT SAME GAME
- ☐ SMALL WIZARD/CAPITAL WIZARD
- ☐ ALPHA-BEE SEQUENCE

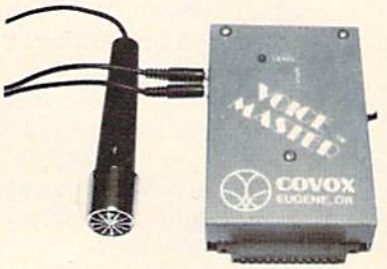
$ 9.95 ea. **$15.95 for 2**
$29.95 all 4 ☐ **FREE BROCHURE**

CA residents add 6½% sales tax.
Personal check or money order accepted.

Kidbit Software
7001 Sunkist Drive
Oakland, CA 94605
(415) 638-3529

TALK OR SING-The "64" responds

IN YOUR OWN VOICE

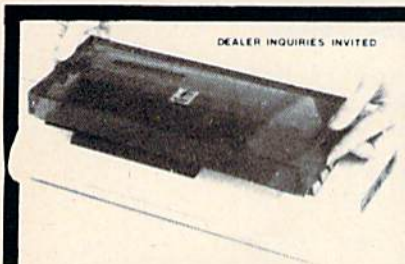


Enter up to 150 of your own words and phrases. Compute response with BASIC. Store word sets on tape or disk for unlimited selections. Easy for anyone to set up and use. Complete with cassette software (transferrable to disk) with demonstration programs for talking clock, calculator, and black jack. How to define every key as a spoken phrase or song note or other sound. There are so many applications and special effects it boggles the mind! Soon available for other popular computers. Software for word recognition to be available.

ONLY $119.95

WE CAN DEMONSTRATE OVER THE TELEPHONE!!

COVOX CO.
675-D Conger St. Eugene, OR 97402
Tel: (503) 342-1271, Telex 706017
Check, money order, or VISA/MC
(Dealer inquiries invited)



DEALER INQUIRIES INVITED

INTRODUCING **K-COVER** AN ANTI-STATIC UNBREAKABLE **COMMADORE** KEYBOARD PROTECTOR MADE OF ATTRACTIVE SMOKED GREY PLASTIC WITH MAR-RESISTANT RUBBER FEET. **K-COVER** PROTECTS YOUR KEYBOARD FROM DUST, DIRT, AND WANDERING FINGERS. **K-COVER** ALSO DOUBLES AS A TILT RISER FOR COMPUTER OR MONITOR AND COMES WITH LIFETIME WARRANTY AGAINST BREAKAGE. ONLY **$7.95 PLUS $1.00 POSTAGE AND HANDLING.** CHECK, MONEY ORDER, VISA AND MASTER CARD ACCEPTED.

PENGUIN PRODUCTS
P.O. BOX 7008
ROSEVILLE, MICHIGAN 48305-7008
1 (800) 732-0614
1(313) 293-5210

DISK DUPLICATOR FOR COMMODORE SINGLE DISK DRIVES

(1540, 1541 and 2031 used with a VIC-20, commodore 64, CBM 4000 or CBM 8000 computer)

DISK DUPLICATOR provides you a fast and easy way to make back-up copies of your precious, irreplaceable diskettes. Enjoy the convenience of a dual disk drive without the expense. DISK DUPLICATOR is 100% MACHINE LANGUAGE, 100% FAST, and most importantly, 100% AFFORDABLE!

Don't let an accident or mistake catch you without back-up copies of all your diskettes. ORDER "DISK DUPLICATOR" TODAY at the special introductory price of only $14.95 postage paid (check or money order only please).

H&H ENTERPRISES DEPT. 123G
5056 NORTH 41st STREET
MILWAUKEE, WISCONSIN 53209

VIC-20 & C-64
SOFTWARE

All programs run on a Vic-20 computer. Each program is on an individual cassette tape except for specials. All programs are guaranteed to run or will be replaced by a new tape if defective tape is returned within 10 days. Individual programs $3.00 each.

SET A: Mathematics
Add/Sub/Div/Mul ... 0101
Math Instructor ... 0102
Add Fast ... 0103
Reverse/Numbers ... 0104
Guessing Game ... 0105

SET B: Science
Bio (printer req.) ... 0201
Chemistry ... 0202
Density ... 0203
Life Espec. (16K) ... 0204
Zodiac (16K) ... 0205

SET C: Social Studies
States & Capitals 16K 0301
Geography (16K) ... 0302
Hawaii (16K) ... 0303
Presidents Quiz (16K) 0304
World Capitals (16K) 0305

SET D: Language
Spelling ... 0401
Scramble Word (16K) 0402
Letter Squares (16K) 0403
Contractions #2 ... 0404
Syno/Anonims ... 0405

SET E: Home-Management
Home Budget ... 0501
Inventory ... 0502
Investments ... 0503
Income Exp. Budget 0504
Phone Directory ... 0505

SET F: Home-Management
Mail List ... 0506
Sheet Music ... 0507
Check Book ... 0508
Loan Calcul. ... 0509
Decis.-Maker ... 0510
Address Book ... 0511

SET G: Space Games
Star Wars ... 0601
Asteroids ... 0602
Invaders ... 0603
Lunar Lander ... 0604
Star Chaser ... 0605

SET H: Gambling Games
Acie Ducie Game ... 0606
Bandit #1 ... 0607
Solitaire Poker ... 0608
Quick Draw ... 0609
Vic Slots ... 0610

SET I: Maze Games
Peackman ... 0611
Maze Chase ... 0612
Tele-Mazeman ... 0613
Dragon Maze ... 0614
Maze (16K) ... 0615

SET J: Auto Racing
Grand Prix ... 0616
Forest Driver ... 0617
Trail Ride ... 0618
Car Race ... 0619
Road Race ... 0620

SET K: War Games
Tankers/UFO ... 0621
Artillary ... 0622
Depth Charge ... 0623
Out Post ... 0624
War Fish ... 0625

SET L: Sports
Pinball ... 0626
Miniature Golf ... 0627
Rugby ... 0628
Ping Pong ... 0629
Round Fishing ... 0630

"SPECIALS"
S001 You select any 1 set ... 9.95
S002 You select any 4 sets ... 19.95
S003 You select any 8 sets ... 29.95
S004 All 12 sets ... 39.95

ORDER FROM: **Satellite Technology**
Software Division A1
4955 Monterey Dr.
Arcata, CA 95521

SHIPPING CHARGES: Add $1.50 shipping and handling ($3.50 for C.O.D.). If California residence add 6% sales tax. When ordering request stock number.

500 different programs available. For additional information about C-64 software and other programs, send a self-addressed stamped envelope to SATELLITE SOFTWARE, P.O. Box 4842, Arcata, CA 95521. Dealer discounts available.

TIRED OF TRYING TO LEARN C-64* OR VIC20* BASIC FROM A BOOK?
Try the Next Best Thing To A Classroom... VIDEO TAPE !!
INCLUDES:
- **BINARY NUMBER SYSTEM**
- **ALL BASIC COMMANDS**
- **GRAPHICS** *including* **PROGRAM. CHARACTERS & SPRITES** *(C-64 only)*
- **SOUND**
- **HINTS & TECHNIQUES**

8 HOURS only $89.95
(IL residents add 5% sales tax)
PLEASE SPECIFY:
VHS *OR* **BETA** *AND*
C-64* *OR* **VIC 20*** **BASIC**
SEND CHECK OR MONEY ORDER TO:

TECH COM
PO BOX 1654
PEO, IL 61607
(309) 697-1393

** C-64 and VIC 20 are registered trademarks of COMMODORE INTERN.*

PROFESSIONAL FOOTBALL

AN ADVENTURE TYPE GAME FOR
Vic + 16K & Commodore 64

Program Features
11 Offenses & 11 Defenses
Full Feature Scoreboard
Includes down, distance, field position, clock, # 1st downs, and # timeouts.
Computer selects its plays based on down, distance, field position & score & will run a 2 min drill.
Sudden Death (for ties)
Complete Statistics Given at Half and at end of game.
** Includes Users Manual **

(Cassette)
$16.95 ppd

Personal Checks Accepted
VISA & MC Include Card #
Exp. Date & Signature

CMS SOFTWARE
PO Box 4876
Topeka, Kansas 66604—0876

MLX For VIC And 64

(Article on page 163.)

BEFORE TYPING...

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

Program 1: MLX—64 Version

```
100 PRINT"{CLR}{CYN}";CHR$(142);CHR$(8);:
    POKE53281,1:POKE53280,1          :rem 73
101 POKE 788,52:REM DISABLE RUN/STOP
                                     :rem 119
110 PRINT"{RVS}{40 SPACES}";         :rem 176
120 PRINT"{RVS}{15 SPACES}{RIGHT}{OFF}
    {*}£{RVS}{RIGHT} {RIGHT}{2 SPACES}
    {*}{OFF}{*}£{RVS}£{RVS}
    {13 SPACES}";                    :rem 250
130 PRINT"{RVS}{15 SPACES}{RIGHT} {G}
    {RIGHT} {2 RIGHT} {OFF}£{RVS}£{*}
    {OFF}{*}{RVS}{13 SPACES}";       :rem 35
140 PRINT"{RVS}{40 SPACES}"          :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK}{3 SPACES}A F
    AILSAFE MACHINE LANGUAGE EDITOR
    {5 DOWN}"                        :rem 130
210 PRINT"{5}{2 UP}STARTING ADDRESS?
    {8 SPACES}{9 LEFT}";             :rem 143
215 INPUTS:F=1-F:C$=CHR$(31+119*F:rem 125
220 IFS<256OR(S>40960ANDS<49152)ORS>53247
    THENGOSUB3000:GOTO210            :rem 235
225 PRINT:PRINT:PRINT                :rem 180
230 PRINT"{5}{2 UP}ENDING ADDRESS?
    {8 SPACES}{9 LEFT}";:INPUTE:F=1-F:C$=
    CHR$(31+119*F)                   :rem 20
240 IFE<256OR(E>40960ANDE<49152)ORE>53247
    THENGOSUB3000:GOTO230            :rem 183
250 IFE<STHENPRINTC$;"{RVS}ENDING < START
    {2 SPACES}":GOSUB1000:GOTO 230
                                     :rem 176
260 PRINT:PRINT:PRINT                :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S:POKEV+21,0
                                     :rem 225
310 PRINTRIGHT$("0000"+MID$(STR$(AD),2),5
    );":";:FORJ=1TO6                 :rem 234
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                     :rem 228
390 IFN=-211THEN 710                 :rem 62
400 IFN=-204THEN 790                 :rem 64
410 IFN=-206THENPRINT:INPUT"{DOWN}ENTER N
    EW ADDRESS";ZZ                   :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}OUT OF RANGE":GOSUB1000:GOTO410
                                     :rem 225
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                     :rem 238
420 IF N<>-196 THEN 480              :rem 133
430 PRINT:INPUT"DISPLAY:FROM";F:PRINT,"TO
    ";:INPUTT                        :rem 234
440 IFF<SORF>EORT<SORT>ETHENPRINT"AT LEAS
    T";S;"{LEFT}, NOT MORE THAN";E:GOTO43
    0                                :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":";       :rem 30
```

```
451 FORK=ØTO5:N=PEEK(I+K):PRINTRIGHT$("ØØ
    "+MID$(STR$(N),2),3);",";          :rem 66
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO31Ø
                                       :rem 25
470 NEXTK:PRINTCHR$(2Ø);:NEXTI:PRINT:PRIN
    T:GOTO31Ø                          :rem 5Ø
480 IFN<Ø THEN PRINT:GOTO31Ø          :rem 168
490 A(J)=N:NEXTJ                      :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT       :rem 200
510 PRINTCHR$(18);:GOSUB57Ø:PRINTCHR$(2Ø)
                                      :rem 234
515 IFN=CKSUMTHEN53Ø                  :rem 255
520 PRINT:PRINT"LINE ENTERED WRONG : RE-E
    NTER":PRINT:GOSUB1ØØØ:GOTO31Ø:rem 176
530 GOSUB2ØØØ                         :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54
    272,Ø:POKE54273,Ø                 :rem 227
550 AD=AD+6:IF AD<E THEN 31Ø          :rem 212
560 GOTO 71Ø                          :rem 108
570 N=Ø:Z=Ø                            :rem 88
580 PRINT"[+]";                        :rem 79
581 GETA$:IFA$=""THEN581               :rem 95
585 PRINTCHR$(2Ø);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN67Ø                     :rem 229
590 IFA>128THENN=-A:RETURN            :rem 137
600 IFA<>2Ø THEN 63Ø                   :rem 10
610 GOSUB69Ø:IFI=1ANDT=44THENN=-1:PRINT"
    {LEFT} {LEFT}";:GOTO69Ø           :rem 172
620 GOTO57Ø                           :rem 109
630 IFA<48ORA>57THEN58Ø               :rem 105
640 PRINTA$;:N=N*1Ø+A-48              :rem 106
650 IFN>255 THEN A=2Ø:GOSUB1ØØØ:GOTO6ØØ
                                      :rem 229
660 Z=Z+1:IFZ<3THEN58Ø                 :rem 71
670 IFZ=ØTHENGOSUB1ØØØ:GOTO57Ø        :rem 114
680 PRINT",";:RETURN                  :rem 240
690 S%=PEEK(2Ø9)+256*PEEK(21Ø)+PEEK(211)
                                      :rem 149
691 FORI=1TO3:T=PEEK(S%-I)             :rem 67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                      :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1);:RETURN
                                        :rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***{3 DOWN}"
                                      :rem 236
720 INPUT"{DOWN} FILENAME";F$         :rem 228
730 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
    {RVS}D{OFF}ISK: (T/D)"            :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"THEN74Ø
                                       :rem 36
750 DV=1-7*(A$="D"):IFDV=8THENF$="Ø:"+F$
                                      :rem 158
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                    :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE78Ø,LEN(
    T$):SYS65469                      :rem 109
763 POKE78Ø,1:POKE781,DV:POKE782,1:SYS654
    66                                 :rem 69
765 POKE254,S/256:POKE253,S-PEEK(254)*256
    :POKE78Ø,253                       :rem 12
766 POKE782,E/256:POKE781,E-PEEK(782)*256
    :SYS65496                         :rem 124
770 IF(PEEK(783)AND1)OR(ST AND191)THEN78Ø
                                      :rem 111
775 PRINT"{DOWN}DONE.":END            :rem 106
780 PRINT"{DOWN}ERROR ON SAVE.{2 SPACES}T
    RY AGAIN.":IFDV=1THEN72Ø          :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO72Ø              :rem 103
790 PRINT"{CLR}{RVS}*** LOAD ***{2 DOWN}"
                                      :rem 212
800 INPUT"{2 DOWN} FILENAME";F$       :rem 244
810 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
    {RVS}D{OFF}ISK: (T/D)"            :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"THEN82Ø
                                       :rem 34
830 DV=1-7*(A$="D"):IFDV=8THENF$="Ø:"+F$
                                      :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                    :rem 2
841 POKE781,ZK-PEEK(782)*256:POKE78Ø,LEN(
    T$):SYS65469                      :rem 107
845 POKE78Ø,1:POKE781,DV:POKE782,1:SYS654
    66                                 :rem 70
850 POKE78Ø,Ø:SYS65493                 :rem 11
860 IF(PEEK(783)AND1)OR(ST AND191)THEN87Ø
                                      :rem 111
865 PRINT"{DOWN}DONE.":GOTO31Ø         :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.{2 SPACES}T
    RY AGAIN.{DOWN}":IFDV=1THEN8ØØ
                                      :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO8ØØ              :rem 102
1000 REM BUZZER                       :rem 135
1001 POKE54296,15:POKE54277,45:POKE54278,
     165                              :rem 207
1002 POKE54276,33:POKE 54273,6:POKE54272,
     5                                 :rem 42
1003 FORT=1TO2ØØ:NEXT:POKE54276,32:POKE54
     273,Ø:POKE54272,Ø:RETURN         :rem 202
2000 REM BELL SOUND                    :rem 78
2001 POKE54296,15:POKE54277,Ø:POKE54278,2
     47                               :rem 152
2002 POKE 54276,17:POKE54273,4Ø:POKE54272
     ,Ø                                :rem 86
2003 FORT=1TO1ØØ:NEXT:POKE54276,16:RETURN
                                       :rem 57
3000 PRINTC$;"{RVS}NOT ZERO PAGE OR ROM":
     GOTO1ØØØ                          :rem 89
```

Program 2: MLX—VIC Version

```
100 PRINT"{CLR}{PUR}";CHR$(142);CHR$(8);
                                      :rem 181
101 POKE 788,194:REM DISABLE RUN/STOP
                                      :rem 174
110 PRINT"{RVS}{14 SPACES}"           :rem 117
120 PRINT"{RVS} {RIGHT}?{OFF}[*]£{RVS}
    {RIGHT} {RIGHT}{2 SPACES}[*]{OFF}
    [*]£{RVS}£{RVS} "                 :rem 191
130 PRINT"{RVS} {RIGHT} [G]{RIGHT}
    {2 RIGHT} {OFF}£{RVS}£[*]{OFF}
    [*]{RVS} "                        :rem 232
140 PRINT"{RVS}{14 SPACES}"           :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK}A FAILSAFE MA
    CHINE":PRINT"LANGUAGE EDITOR{5 DOWN}"
                                      :rem 141
210 PRINT"{BLK}{3 UP}STARTING ADDRESS":IN
    PUTS:F=1-F:C$=CHR$(31+119*F)       :rem 97
220 IFS<256ORS>32767THENGOSUB3ØØØ:GOTO21Ø
                                        :rem 2
225 PRINT:PRINT:PRINT:PRINT           :rem 123
230 PRINT"{BLK}{3 UP}ENDING ADDRESS":INPU
    TE:F=1-F:C$=CHR$(31+119*F)        :rem 158
240 IFE<256ORE>32767THENGOSUB3ØØØ:GOTO23Ø
                                      :rem 234
250 IFE<STHENPRINTC$;"{RVS}ENDING < START
    {2 SPACES}":GOSUB1ØØØ:GOTO 23Ø
                                      :rem 176
260 PRINT:PRINT:PRINT                 :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S         :rem 56
```

```
310 PRINTRIGHT$("0000"+MID$(STR$(AD),2),5
    );":";:FORJ=1TO6                :rem 234
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                    :rem 228
390 IFN=-211THEN 710                :rem 62
400 IFN=-204THEN 790                :rem 64
410 IFN=-206THENPRINT:INPUT"{DOWN}ENTER N
    EW ADDRESS";ZZ                  :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}OUT OF RANGE":GOSUB1000:GOTO410
                                    :rem 225
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                    :rem 238
420 IF N<>-196 THEN 480             :rem 133
430 PRINT:INPUT"DISPLAY:FROM";F:PRINT,"TO
    ";:INPUTT                       :rem 234
440 IFF<SORF>EORT<SORT>ETHENPRINT"AT LEAS
    T";S;"{LEFT}, NOT MORE THAN",E:GOTO43
    0                               :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":";      :rem 30
455 FORK=0TO5:N=PEEK(I+K):IFK=3THENPRINTS
    PC(10);                         :rem 34
457 PRINTRIGHT$("00"+MID$(STR$(N),2),3);"
    ,";                             :rem 157
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                                    :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
    T:GOTO310                       :rem 50
480 IFN<0 THEN PRINT:GOTO310        :rem 168
490 A(J)=N:NEXTJ                    :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT     :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20)
                                    :rem 234
515 IFN=CKSUMTHEN530                :rem 255
520 PRINT:PRINT"LINE ENTERED WRONG":PRINT
    "RE-ENTER":PRINT:GOSUB1000:GOTO310
                                    :rem 129
530 GOSUB2000                       :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:rem 80
550 AD=AD+6:IF AD<E THEN 310        :rem 212
560 GOTO 710                        :rem 108
570 N=0:Z=0                         :rem 88
580 PRINT"[+]";                     :rem 79
581 GETA$:IFA$=""THEN581            :rem 95
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN670                   :rem 229
590 IFA>128THENN=-A:RETURN          :rem 137
600 IFA<>20 THEN 630                :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
    {LEFT} {LEFT}";:GOTO690         :rem 172
620 GOTO570                         :rem 109
630 IFA<48ORA>57THEN580             :rem 105
640 PRINTA$;:N=N*10+A-48            :rem 106
650 IFN>255 THEN A=20:GOSUB1000:GOTO600
                                    :rem 229
660 Z=Z+1:IFZ<3THEN580              :rem 71
670 IFZ=0THENGOSUB1000:GOTO570      :rem 114
680 PRINT",";:RETURN                :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                    :rem 149
692 FORI=1TO3:T=PEEK(S%-I)          :rem 68
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                    :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1);:RETURN
                                    :rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***{3 DOWN}"
                                    :rem 236
720 INPUT"{DOWN} FILENAME";F$       :rem 228
730 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
    {RVS}D{OFF}ISK: (T/D)"          :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"THEN740
                                    :rem 36
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                    :rem 158
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                    :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                              :rem 69
765 POKE254,S/256:POKE253,S-PEEK(254)*256
    :POKE780,253                    :rem 12
766 POKE782,E/256:POKE781,E-PEEK(782)*256
    :SYS65496                       :rem 124
770 IF(PEEK(783)AND1)OR(ST AND191)THEN780
                                    :rem 111
775 PRINT"{DOWN}DONE.":END          :rem 106
780 PRINT"{DOWN}ERROR ON SAVE.{2 SPACES}T
    RY AGAIN.":IFDV=1THEN720        :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO720            :rem 103
782 GOTO720                         :rem 115
790 PRINT"{CLR}{RVS}*** LOAD ***{2 DOWN}"
                                    :rem 212
800 INPUT"{2 DOWN} FILENAME";F$     :rem 244
810 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
    {RVS}D{OFF}ISK: (T/D)"          :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"THEN820
                                    :rem 34
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                    :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                :rem 2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                    :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                              :rem 70
850 POKE780,0:SYS65493              :rem 11
860 IF(PEEK(783)AND1)OR(ST AND191)THEN870
                                    :rem 111
865 PRINT"{DOWN}DONE.":GOTO310      :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.{2 SPACES}T
    RY AGAIN.{DOWN}":IFDV=1THEN800
                                    :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO800            :rem 102
1000 REM BUZZER                     :rem 135
1001 POKE36878,15:POKE36874,190     :rem 206
1002 FORW=1TO300:NEXTW              :rem 117
1003 POKE36878,0:POKE36874,0:RETURN
                                    :rem 74
2000 REM BELL SOUND                 :rem 78
2001 FORW=15TO0STEP-1:POKE36878,W:POKE368
    76,240:NEXTW                    :rem 22
2002 POKE36876,0:RETURN             :rem 119
3000 PRINTC$;"{RVS}NOT ZERO PAGE OR ROM":
    GOTO1000                        :rem 89
```

Speed Reader

(Article on page 82.)

Program 1:

Speed Reader — 64 Version

```
10 POKE 53280,15 : POKE 53281,15 : REM SE
   T BORDER AND BACKGROUND COLORS:rem 111
20 PRINT "{WHT}"                    :rem 56
```

BEFORE TYPING...

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

```
30 CL = 55776 : REM START OF SCREEN COLOR
   MEMORY                              :rem 150
40 DD = 50 : NC = 5 : REM VARIABLES FOR D
   ELAY & NUMBER OF DISPLAYED CHARACTER
                                       :rem 136
50 PRINT "{CLR}" : REM CLEAR SCREEN
                                       :rem 14
60 FOR I = 1 TO 4 : PRINT : NEXT I
                                       :rem 100
70 PRINT TAB(14) "SPEED READER"        :rem 233
80 PRINT : PRINT : PRINT               :rem 131
90 PRINT"HOLD DOWN THESE KEYS TO CHANGE D
   ISPLAY"                             :rem 173
100 PRINT : PRINT "{3 SPACES}F - MAKES RE
    ADING SPEED FASTER"                :rem 61
110 PRINT : PRINT "{3 SPACES}S - MAKES RE
    ADING SPEED SLOWER"                :rem 98
120 PRINT : PRINT "{3 SPACES}> - DISPLAYS
     MORE CHARACTERS ON LINE"          :rem 215
130 PRINT : PRINT "{3 SPACES}< - DISPLAYS
     FEWER CHARACTERS ON LINE";        :rem 87
140 PRINT : PRINT "{3 SPACES}Q - QUIT THE
     PROGRAM"                          :rem 231
150 PRINT : PRINT : PRINT              :rem 177
160 PRINT "PRESS SPACEBAR TO START THE SP
    EED READER":PRINT                  :rem 51
170 S=PEEK(197):IFS<>60ANDS<>62THEN170
                                       :rem 10
175 IFS=62THEN60000                    :rem 73
180 POKE 788,52 : REM DISABLE STOP KEY
                                       :rem 67
190 PRINT "{8}"                        :rem 6
200 READ A$                            :rem 19
210 IF A$ = "EOD" THEN GOTO 60000      :rem 62
230 PRINT"{CLR}{12 DOWN}"A$            :rem 42
240 FOR L = 0 TO 39                    :rem 69
250 POKE CL + L,12 : REM POKE CONTRASTING
     COLOR FOR EACH LETTER IN LINE
                                       :rem 250
260 IF L - NC >= 0 THEN POKE CL + L - NC,
    15 : REM POKE BACK TO BACKGROUND COLO
    R                                  :rem 240
270 FOR D = 0 TO DD : NEXT D           :rem 25
280 NEXT L                             :rem 37
290 S = PEEK(197) : REM SCAN KEYBOARD FOR
     KEYPRESS                          :rem 51
300 IF S = 21 THEN DD = DD - 10 : REM DEC
    REASE DELAY - READ FASTER          :rem 246
310 IF S = 13 THEN DD = DD + 10 : REM INC
    REASE DELAY - READ SLOWER          :rem 27
320 IF S = 44 THEN NC = NC + 2 : IF NC >
    {SPACE}11 THEN NC = NC - 2         :rem 238
330 REM LINE 320 INCREASES THE NUMBER OF
    {SPACE}CHARACTERS DISPLAYED AT ONE TI
    ME                                 :rem 56
340 IF S = 47 THEN NC = NC - 2 : IF NC <
    {SPACE}3 THEN NC = NC + 2          :rem 194
350 REM LINE 340 DECREASES THE NUMBER OF
    {SPACE}CHARACTERS DISPLAYED AT ONE TI
    ME                                 :rem 46
360 IF S = 62 THEN GOTO 60000 : REM END T
    HE PROGRAM                         :rem 108
370 GOTO 200                           :rem 101
400 REM ** TRY THIS SAMPLE DATA - THEN SU
    BSTITUTE YOUR OWN SELECTION{2 SPACES}
    **                                 :rem 148
410 DATA "{2 SPACES}THIS IS A SAMPLE OF T
    HE SPEED READING"                  :rem 171
420 DATA "PROGRAM FOR THE COMMODORE 64.
    {2 SPACES}YOU CAN"                 :rem 224
430 DATA "PUT ANY KIND OF READING MATERIA
    L IN THE"                          :rem 82
440 DATA "DATA STATEMENTS, AND THE COMPUT
    ER WILL"                           :rem 159
450 DATA "DISPLAY ONE LINE AT A TIME FROM
     LEFT TO"                          :rem 30
460 DATA "RIGHT FORCING YOU TO READ WITH
    {SPACE}THE"                        :rem 87
470 DATA "CORRECT EYE HABITS.{2 SPACES}WI
    TH ENOUGH"                         :rem 217
480 DATA "PRACTICE, IT IS POSSIBLE TO DOU
    BLE OR"                            :rem 10
490 DATA "TRIPLE YOUR READING SPEED.
    {2 SPACES}IN THIS"                 :rem 130
500 DATA "PROGRAM YOU CAN USE THE KEYBOAR
    D TO"                              :rem 156
510 DATA "INCREASE YOUR READING RATE WITH
     THE (F)"                          :rem 103
520 DATA "KEY, OR YOU CAN SLOW DOWN THE P
    ROGRAM"                            :rem 240
530 DATA "WITH THE (S) KEY."           :rem 206
540 DATA "{3 SPACES}IF YOU WANT TO SEE MO
    RE WORDS ON THE"                   :rem 125
550 DATA "LINE, HOLD DOWN THE > KEY.
    {2 SPACES}THE <"                   :rem 254
560 DATA "KEY SHOWS FEWER CHARACTERS ON T
    HE LINE AT"                        :rem 10
570 DATA "ONE TIME.{2 SPACES}WHEN YOU ARE
     READY TO QUIT,"                   :rem 199
580 DATA "JUST PRESS THE (Q) KEY."
                                       :rem 104
590 DATA "{2 SPACES}THIS PROGRAM DUPLICAT
    ES SOME OF THE"                    :rem 228
600 DATA "SAME TECHNIQUES USED IN REGULAR
     SPEED"                            :rem 94
610 DATA "READING CLASSES.{2 SPACES}BECAU
    SE WE ARE TRYING"                  :rem 116
620 DATA "TO CORRECT EYE MOVEMENT HABITS,
     WE WILL"                          :rem 180
630 DATA "HAVE TO DEVOTE SOME AMOUNT OF P
    RACTICE"                           :rem 109
640 DATA "TO COR"{RVS}CTING OUR BAD HABIT
    S, BUT THE"                        :rem 152
650 DATA "READING EXPERTS SEE NO REASON W
    HY WE"                             :rem 244
660 DATA "CANNOT MAKE OURSELVES BETTER AN
    D FASTER"                          :rem 1
670 DATA "READERS IF WE WORK TOWARD THAT
    {SPACE}GOAL."                      :rem 194
60000 POKE198,0:POKE 788,49 : REM ENABLE
      {SPACE}THE STOP KEY              :rem 161
60010 PRINT "{CLR}"                    :rem 91
60020 PRINT "{WHT}"                    :rem 206
60030 DATA "EOD"                       :rem 47
```

Program 2:
Speed Reader — VIC Version

```
5 POKE 808,114:REM DISABLE STOP KEY:rem 7
10 POKE 36879,110:REM SET BORDER AND BACK
   GROUND COLORS                       :rem 172
20 PRINT "{WHT}"                       :rem 56
```

```
30 CL = 38576:REM START OF SCREEN COLOR M
   EMORY                              :rem 149
40 DD=50:NC=3:REM VARIABLES FOR DELAY AND
    NUMBER OF DISPLAYED CHARACTERS
                                      :rem 134
50 PRINT"{CLR}":REM CLEAR SCREEN      :rem 14
60 FORI=1TO4:PRINT:NEXTI              :rem 100
70 PRINTTAB(3)"**SPEED READER**"      :rem 95
80 PRINT:PRINT                        :rem 188
90 PRINT"HOLD DOWN KEY:"              :rem 188
100 PRINT:PRINT"{RVS}F{OFF}-FOR FAST READ
    ING{6 SPACES}SPEED"               :rem 192
110 PRINT:PRINT "{RVS}S{OFF}-FOR SLOW REA
    DING{6 SPACES}SPEED"              :rem 229
120 PRINT: PRINT"{RVS}>{OFF}-FOR MORE CHA
    RACTERS{3 SPACES}ON LINE"         :rem 249
130 PRINT:PRINT"{RVS}<{OFF}-FOR FEWER CHA
    RACTERS{2 SPACES}ON LINE"         :rem 62
140 PRINT:PRINT"{RVS}Q{OFF}-QUIT THE PROG
    RAM"                              :rem 139
150 PRINT:PRINT                       :rem 234
160 PRINT"{2 RIGHT}{YEL}{RVS}{RIGHT}SPACE
    BAR TO START{OFF}"                :rem 115
170 S=PEEK(197):IFS<>32ANDS<>48THEN170
                                      :rem 13
175 IFS=48THEN60000                   :rem 77
190 POKE36879,59                      :rem 116
200 READA$                            :rem 19
210 IFA$="EOD"THEN60000               :rem 5
230 PRINT"{CYN}{CLR}{8 DOWN}"A$       :rem 133
240 FORL=0TO21                        :rem 60
250 POKECL+L,0:REM POKE CONTRASTING COLOR
     FOR EACH LETTER IN LINE          :rem 199
260 IFL-NC>=0THEN POKECL+L-NC,3:REM POKE
    {SPACE}BACK TO BACKGROUND COLOR
                                      :rem 189
270 FORD=0TODD:NEXT D                 :rem 25
280 NEXT L                            :rem 37
290 S=PEEK(197):REM SCAN KEYBOARD FOR KEY
    PRESS                             :rem 51
300 IFS=42THENDD=DD-10:REM DECREASE DELAY
    -READ FASTER                      :rem 249
310 IFS=41THENDD=DD+10:REM INCREASE DELAY
    -READ SLOWER                      :rem 28
320 IFS=37THENNC=NC+2:IFNC>11THENNC=NC-2
                                      :rem 240
330 REM LINE 320 INCREASES THE NUMBER OF
    {SPACE}CHARACTERS DISPLAYED AT ONE TI
    ME                                :rem 56
340 IFS=29THENNC=NC-2:IFNC<3THENNC=NC+2
                                      :rem 194
350 REM LINE 340 DECREASES THE NUMBER OF
    {SPACE}CHARACTERS DISPLAYED AT ONE TI
    ME                                :rem 46
360 IFS=48THEN60000:REM END THE PROGRAM
                                      :rem 55
370 GOTO200                           :rem 101
400 REM ** TRY THIS SAMPLE DATA - THEN SU
    BSTITUTE YOUR OWN SELECTION{2 SPACES}
    **                                :rem 148
410 DATA "{2 SPACES}THIS IS A SAMPLE OF"
                                      :rem 95
415 DATA"THE SPEED READING"           :rem 68
420 DATA"PROGRAM FOR THE VIC-20"      :rem 69
425 DATA". YOU CAN PUT ANY KIND"      :rem 253
430 DATA " OF READING MATERIAL"       :rem 211
435 DATA" IN THE DATA"                :rem 140
440 DATA " STATEMENTS, AND THE"       :rem 222
445 DATA" COMPUTER WILL DISPLAY"      :rem 184
450 DATA " ONE LINE AT A TIME"          :rem 6
455 DATA" FROM LEFT TO RIGHT"         :rem 124
460 DATA " FORCING YOU TO READ"       :rem 188
465 DATA" WITH THE CORRECT EYE"        :rem 15
470 DATA "HABITS. WITH ENOUGH"        :rem 228
480 DATA "PRACTICE, IT IS "           :rem 170
485 DATA"POSSIBLE TO DOUBLE OR"        :rem 95
490 DATA "TRIPLE YOUR READING "        :rem 20
495 DATA"SPEED. IN THIS PROGRAM"      :rem 134
500 DATA " YOU CAN USE THE "          :rem 144
505 DATA"KEYBOARD TO INCREASE"         :rem 54
510 DATA " YOUR READING RATE "        :rem 105
515 DATA"WITH THE (F) KEY, OR"         :rem 99
520 DATA " YOU CAN SLOW DOWN THE "    :rem 34
525 DATA"PROGRAM"                      :rem 18
530 DATA "WITH THE (S) KEY."          :rem 206
540 DATA "{3 SPACES}IF YOU WANT TO SEE "
                                      :rem 61
545 DATA"MORE WORDS ON THE LINE"      :rem 100
550 DATA ", HOLD DOWN THE > KEY. "
                                      :rem 185
555 DATA"THE < KEY SHOWS FEWER"        :rem 16
560 DATA " CHARACTERS ON THE "         :rem 87
565 DATA"LINE AT ONE TIME. WHEN"       :rem 44
570 DATA " YOU ARE READY TO QUIT"     :rem 42
580 DATA ", JUST PRESS THE (Q) "      :rem 125
585 DATA"KEY. THIS PROGRAM "          :rem 103
590 DATA "DUPLICATES SOME OF THE"     :rem 148
600 DATA "SAME TECHNIQUES USED "       :rem 68
605 DATA"IN REGULAR SPEED"             :rem 19
610 DATA " READING CLASSES. "          :rem 43
615 DATA"BECAUSE WE ARE TYPING "       :rem 65
620 DATA "TO CORRECT EYE "            :rem 142
625 DATA"MOVEMENT HABITS, WE "         :rem 233
630 DATA "HAVE TO DEVOTE SOME "       :rem 185
635 DATA"AMOUNT OF PRACTICE "         :rem 176
640 DATA "TO CORRECTING OUR BAD "      :rem 72
645 DATA"HABITS, BUT THE"             :rem 176
650 DATA "READING EXPERTS SEE NO"     :rem 152
655 DATA" REASON WHY WE CANNOT "       :rem 29
660 DATA "MAKE OURSELVES BETTER "     :rem 166
665 DATA"AND FASTER READERS IF "       :rem 44
670 DATA "WE WORK TOWARD THAT "       :rem 220
675 DATA"GOAL."                        :rem 81
60000 POKE198,0:POKE 808,112:REM ENABLE T
      HE STOP KEY                     :rem 193
60010 PRINT"{CLR}"                     :rem 91
60020 PRINT"{BLK}"                     :rem 89
60030 DATA"EOD"                        :rem 47
```

VIC Piano

(Article on page 94.)

BEFORE TYPING...

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

```
10 POKE36879,106:PRINT"{CLR}{BLK}";
                                      :rem 205
12 PRINT"{RVS}{YEL}{6 SPACES}VIC
   {2 SPACES}PIANO{28 SPACES}";       :rem 120
20 PRINT"{RVS}{WHT}G{BLK} {WHT}A{BLK}
   {WHT}BC{BLK} {WHT}D{BLK} {WHT}EF{BLK}
   {SPACE}{WHT}G{BLK} {WHT}A{BLK} {WHT}BC
   {BLK} {WHT}D{BLK} {WHT}E";         :rem 52
25 FORT=1TO4                          :rem 231
```

```
30 PRINT"{RVS}{WHT} {BLK} {WHT} {BLK}
   {WHT} {G}{BLK} {WHT} {BLK} {WHT}
   {G}{BLK} {WHT} {BLK} {WHT} {BLK}
   {WHT} {G}{BLK} {WHT} {BLK} {WHT} ";
                                  :rem 178
40 NEXTT                          :rem 247
50 FORT=1TO3                      :rem 228
60 PRINT"{RVS}{WHT} _ _ {G}_ _ {G}_ _
   _ {G}_ _ {OFF}";               :rem 207
70 NEXT                           :rem 166
75 PRINT"{DOWN}{WHT} 2 3{2 SPACES}5 6
   {2 SPACES}8 9 0{2 SPACES}- £ ";
                                  :rem 136
80 PRINT"Q W ER T YU I O P@ * ↑";:rem 101
100 PRINT"{5 DOWN}";              :rem 242
110 PRINT"PLAY EACH NOTE BY THE CHARACTER
    S ABOVE.";                    :rem 18
120 PRINT"PRESS{2 SPACES}SPACE BAR TO RES
    T."                           :rem 65
160 DIM N(55),J(55)               :rem 171
170 FOR I=0 TO 55:J(I)=264:N(I)=0:NEXT I:
    XX=55                         :rem 27
180 FOR I=0 TO 21:READ K,M:J(K-42)=I:N(K-
    42)=M:NEXT I                  :rem 146
190 POKE36878,15                  :rem 107
400 GETA$:IFA$=""THEN400          :rem 75
420 X=ASC(A$)-42:IF X<0 OR X>55 THEN X=55
                                  :rem 167
430 POKE7900+J(XX),32             :rem 157
440 POKE36876,0                   :rem 49
450 POKE36876,N(X)                :rem 249
460 XX=X                          :rem 223
470 POKE7900+J(X),81              :rem 77
480 GOTO400                       :rem 105
800 DATA 81,175,50,179,87,183,51,187
                                  :rem 3
810 DATA 69,191,82,195,53,198,84,201,54,2
    04                            :rem 90
820 DATA 89,207,85,210,56,212,73,215,57,2
    17,79,219,48,221              :rem 15
830 DATA 80,223,64,225,45,227,42,228,92,2
    29,94,231                     :rem 172
```

Checkers

(Article on page 90.)

```
100 DIMX(4),S(7,7):G=-1:X(0)=-99:PRINT"
    {BLK}{CLR}"                   :rem 235
101 SYS65517:IFPEEK(781)=40THEN103
                                  :rem 220
102 POKE36879,59:GOTO110          :rem 114
103 POKE53280,3:POKE53281,3       :rem 239
110 DATA1,0,1,0,0,0,-1,0,0,1,0,0,0,-1,0,-
    1,15                          :rem 95
120 A$="{19 SPACES}":B$="{HOME}{12 DOWN}"
                                  :rem 121
130 FORX=0TO7:FORY=0TO7:READJ:IFJ=15THEN1
    50                            :rem 246
140 S(X,Y)=J:GOTO160              :rem 167
150 RESTORE:READS(X,Y)            :rem 145
160 NEXTY,X:PRINT"{CLR}";         :rem 140
170 FORX=0TO7:FORY=0TO7:IFS(X,Y)>-1THEN20
    0                             :rem 134
180 IFS(X,Y)=-1THENFORA=-1TO1STEP2:B=G:GO
    SUB210:NEXTA                  :rem 127
190 IFS(X,Y)=-2THENFORA=-1TO1STEP2:FORB=-
    1TO1STEP2:GOSUB210:NEXTB,A    :rem 47
200 NEXTY,X:GOTO370               :rem 187
210 U=X+A:V=Y+B:IFU<0ORU>7ORV<0ORV>7THEN2
    60                            :rem 7
220 IFS(U,V)=0THENGOSUB270:GOTO260:rem 94
230 IFS(U,V)<0THEN260             :rem 210
240 U=U+A:V=V+B:IFU<0ORV<0ORU>7ORV>7THEN2
    60                            :rem 4
250 IFS(U,V)=0THENGOSUB270        :rem 86
260 RETURN                        :rem 120
270 IFV=0ANDS(X,Y)=-1THENQ=Q+2    :rem 69
280 IFABS(Y-V)=2THENQ=Q+5         :rem 9
290 IFY=7THENQ=Q-2                :rem 100
300 IFY=0ORU=7THENQ=Q+1           :rem 188
310 FORC=-1TO1STEP2:IFU+C<0ORU+C>7ORV+G<0
    THEN350                       :rem 8
320 IFS(U+C,V+G)<0THENQ=Q+1:GOTO350
                                  :rem 96
330 IFU-C<0ORU-C>7ORV-G>7THEN350  :rem 216
340 IFS(U+C,V+G)>0AND(S(U-C,V-G)=0OR(U-C=
    XANDV-G=Y))THENQ=Q-2          :rem 203
350 NEXTC:IFQ>X(0)THENX(0)=Q:X(1)=X:X(2)=
    Y:X(3)=U:X(4)=V               :rem 135
360 Q=0:RETURN                    :rem 113
370 IFX(0)=-99THEN1040            :rem 210
380 GOSUB1060:PRINT"ME"X(1);","X(2)"TO"X(
    3)","X(4):X(0)=-99            :rem 222
390 FORXX=1TO400:NEXTXX           :rem 1
400 IFX(4)=0THENS(X(3),X(4))=-2:GOTO420
                                  :rem 202
410 S(X(3),X(4))=S(X(1),X(2))     :rem 224
420 S(X(1),X(2))=0:IFABS(X(1)-X(3))<>2THE
    N510                          :rem 204
430 S((X(1)+X(3))/2,(X(2)+X(4))/2)=0
                                  :rem 252
440 X=X(3):Y=X(4):IFS(X,Y)=-1THENB=-2:FOR
    A=-2TO2STEP4:GOSUB480         :rem 65
450 IFS(X,Y)=-2THENFORA=-2TO2STEP4:FORB=-
    2TO2STEP4:GOSUB480:NEXTB      :rem 210
460 NEXTA:IFX(0)<>-99THENPRINT"TO"X(3)","
    X(4);:X(0)=-99:GOTO400        :rem 210
470 GOTO510                       :rem 106
480 U=X+A:V=Y+B:IFU<0ORU>7ORV<0ORV>7THEN5
    00                            :rem 13
490 IFS(U,V)=0ANDS(X+A/2,Y+B/2)>0THENGOSU
    B270                          :rem 185
500 RETURN                        :rem 117
505 FORI=1TO25:PRINT:NEXT         :rem 130
510 PRINT"{BLK}{HOME} ROW":PRINT"{BLK}
    {2 SPACES}{D}{8 I}{F}":FORY=7TO
    0STEP-1:PRINTY;"{LEFT}{RVS}{K}{OFF}
    ";:FORX=0TO7                  :rem 235
520 IFS(X,Y)=0THENIF(X+Y)/2=INT((X+Y)/2)T
    HENPRINT"{RVS} {OFF}";:GOTO580:rem 86
530 IFS(X,Y)=0THENPRINT" ";       :rem 80
540 IFS(X,Y)=1THENPRINT"{RVS}Q{OFF}";:GOT
    O580                          :rem 215
550 IFS(X,Y)=-1THENPRINT"{RVS}W{OFF}";:GO
    TO580                         :rem 11
560 IFS(X,Y)=-2THENPRINT"*";:GOTO580
                                  :rem 188
570 IFS(X,Y)=2THENPRINT"{RVS}*{OFF}";
                                  :rem 36
580 NEXTX:PRINT"{K}":NEXTY:PRINT"
    {2 SPACES}{C}{RVS}{8 I}{OFF}{V}
    ":PRINT"{3 SPACES}01234567 COL"
                                  :rem 112
590 GOSUB1060:PRINT"FROM";        :rem 95
600 GETG$:IFG$=""THEN600          :rem 91
610 IFG$<"0"ORG$>"7"THEN590       :rem 211
620 E=VAL(G$):PRINTE;",";         :rem 17
630 GETG$:IFG$=""THEN630          :rem 9
640 IFG$<"0"ORG$>"7"THEN590       :rem 214
650 H=VAL(G$):PRINTH              :rem 206
```

```
660 X=E:Y=H:IFS(X,Y)<=0THEN590      :rem 78
670 PRINT"TO";                      :rem 76
680 GETG$:IFG$=""THEN680           :rem 107
690 IFG$<"0"ORG$>"7"THEN670        :rem 218
700 A=VAL(G$):PRINTA;",";          :rem 162
710 GETG$:IFG$=""THEN710            :rem 95
720 IFG$<"0"ORG$>"7"THEN670        :rem 212
730 B=VAL(G$):PRINTB               :rem 193
740 X=A:Y=B                        :rem 131
750 IFS(X,Y)=0ANDABS(A-E)<=2ANDABS(A-E)=A
    BS(B-H)THEN770                   :rem 6
760 GOTO590                        :rem 116
770 I=46                           :rem 142
780 S(A,B)=S(E,H):S(E,H)=0:IFABS(E-A)<>2T
    HEN910                         :rem 168
790 S((E+A)/2,(H+B)/2)=0           :rem 167
800 PRINT"+TO";                    :rem 114
810 GETG$:IFG$=""THEN810            :rem 97
820 IFG$=CHR$(13)THEN910            :rem 80
830 IFG$<"0"ORG$>"7"THEN810        :rem 210
840 A1=VAL(G$):PRINTA1;",";           :rem 9
850 GETG$:IFG$=""THEN850           :rem 105
860 IFG$=CHR$(13)THEN910            :rem 84
870 IFG$<"0"ORG$>"7"THEN850        :rem 218
880 B1=VAL(G$):PRINTB1              :rem 41
890 IFS(A1,B1)<>0ORABS(A1-A)<>2ORABS(B1-B
    )<>2THEN800                      :rem 0
900 E=A:H=B:A=A1:B=B1:I=I+15:GOTO780
                                    :rem 95
910 IFB=7THENS(A,B)=2              :rem 208
920 PRINT"{HOME}{11 DOWN}{3 RIGHT}0123456
    7 COL"                         :rem 11
930 PRINT"{2 UP}{2 SPACES}{C}{RVS}
    {8 I}{OFF}{V}{2 UP}"       :rem 223
940 FORY=0TO7:PRINTY;"{LEFT}{RVS}{K}
    {OFF}";:FORX=0TO7              :rem 160
950 IFS(X,Y)=0THENIF(X+Y)/2=INT((X+Y)/2)T
    HENPRINT"{RVS} {OFF}";:GOTO1010
                                   :rem 130
960 IFS(X,Y)=0THENPRINT" ";:GOTO1010
                                   :rem 140
970 IFS(X,Y)=1THENPRINT"{RVS}Q{OFF}";:GOT
    O1010                            :rem 3
980 IFS(X,Y)=-1THENPRINT"{RVS}W{OFF}";:GO
    TO1010                         :rem 55
990 IFS(X,Y)=-2THENPRINT"*";:GOTO1010
                                   :rem 232
1000 IFS(X,Y)=2THENPRINT"{RVS}*{OFF}";
                                    :rem 73
1010 NEXTX:PRINT"{K}{2 UP}":NEXTY
                                   :rem 249
1020 PRINT"{HOME} ROW":PRINT"{2 SPACES}
     {D}{8 I}{F}{2 UP}"             :rem 67
1030 GOTO170                       :rem 149
1040 GOSUB1060:FORI=1TO40:PRINT"Z";:FORJ=
     1TO50:NEXT:NEXT               :rem 222
1050 PRINT"YOU WIN":END            :rem 147
1060 PRINTB$                       :rem 186
1070 FORXX=1TO8:PRINTA$:NEXTXX:PRINTB$:RE
     TURN                          :rem 68
```

Astro-PANIC!

(Article on page 68.)

```
49152 :076,011,193,120,169,127,184
49158 :141,013,220,169,001,141,179
49164 :026,208,169,233,141,018,039
49170 :208,169,027,141,017,208,020
49176 :169,036,141,020,003,169,050
49182 :192,141,021,003,088,096,059
49188 :173,018,208,201,233,208,053
49194 :031,169,000,141,018,208,097
49200 :169,022,141,024,208,169,013
49206 :200,141,022,208,169,012,038
49212 :141,033,208,141,032,208,055
49218 :169,001,141,025,208,076,174
49224 :005,193,169,233,141,018,063
49230 :208,169,030,141,024,208,090
49236 :169,216,141,022,208,169,241
49242 :000,141,032,208,141,033,133
49248 :208,169,001,141,025,208,080
49254 :230,162,032,159,255,173,089
49260 :141,002,013,137,198,240,071
49266 :003,076,005,193,173,089,141
49272 :198,141,000,208,173,016,088
49278 :208,041,254,013,090,198,162
49284 :141,016,208,173,000,220,122
49290 :041,004,208,029,173,090,171
49296 :198,208,007,173,089,198,249
49302 :201,025,144,017,056,173,254
49308 :089,198,233,002,141,089,140
49314 :198,173,090,198,233,000,030
49320 :141,090,198,173,000,220,222
49326 :041,008,208,029,173,090,211
49332 :198,240,007,173,089,198,061
49338 :201,064,176,017,024,173,073
49344 :089,198,105,002,141,089,048
49350 :198,173,090,198,105,000,194
49356 :141,090,198,173,000,220,002
49362 :041,016,208,047,173,088,015
49368 :198,208,042,056,173,089,214
49374 :198,233,024,133,180,173,139
49380 :090,198,233,000,074,102,157
49386 :180,070,180,070,180,238,128
49392 :088,198,024,165,180,105,232
49398 :033,133,251,133,253,169,194
49404 :007,105,000,133,252,105,086
49410 :212,133,254,104,168,104,209
49416 :170,104,064,032,003,192,061
49422 :169,004,133,252,160,000,220
49428 :185,226,196,153,000,056,068
49434 :200,192,008,208,245,160,015
49440 :000,152,153,000,057,200,082
49446 :192,008,208,248,160,000,086
49452 :185,234,196,153,000,058,102
49458 :200,208,247,169,232,141,223
49464 :248,007,160,007,169,233,112
49470 :153,248,007,136,208,250,040
49476 :169,255,141,028,208,169,014
49482 :000,141,029,208,141,023,104
49488 :208,141,016,208,169,003,057
49494 :141,037,208,169,008,141,022
49500 :038,208,169,212,141,001,093
49506 :208,160,000,185,236,197,060
49512 :153,040,208,200,192,007,136
49518 :208,245,169,006,141,039,150
49524 :208,169,147,032,210,255,113
49530 :160,039,169,160,153,152,187
49536 :007,169,005,153,152,219,065
49542 :136,016,243,160,005,162,088
49548 :024,024,032,240,255,160,107
49554 :000,185,244,197,240,006,250
49560 :032,210,255,200,208,245,022
49566 :160,000,162,023,024,032,047
49572 :240,255,160,000,185,025,005
49578 :198,240,006,032,210,255,087
49584 :200,208,245,169,004,141,119
49590 :035,208,169,014,141,036,017
49596 :208,160,024,169,000,153,134
```

```
49602 :000,212,136,016,250,169,209
49608 :255,141,015,212,169,128,096
49614 :141,018,212,169,143,141,006
49620 :024,212,169,015,141,139,144
49626 :198,169,003,141,136,198,039
49632 :169,000,141,088,198,141,193
49638 :137,198,170,142,090,198,141
49644 :169,184,141,089,198,138,131
49650 :010,168,010,010,010,010,204
49656 :024,105,031,153,091,198,082
49662 :169,000,153,092,198,153,251
49668 :003,208,169,060,157,105,194
49674 :198,032,186,195,232,224,053
49680 :007,208,222,169,255,141,250
49686 :021,208,173,030,208,173,067
49692 :031,208,173,030,208,041,207
49698 :001,240,003,076,230,195,011
49704 :173,141,002,208,251,032,079
49710 :228,255,201,136,208,009,059
49716 :169,032,160,000,145,251,041
49722 :076,188,196,173,088,198,209
49728 :208,003,076,027,195,160,221
49734 :000,169,032,145,251,056,211
49740 :165,251,233,040,133,251,125
49746 :133,253,165,252,233,000,094
49752 :133,252,024,105,212,133,179
49758 :254,173,027,212,009,008,009
49764 :145,253,169,000,145,251,039
49770 :173,031,208,041,254,240,029
49776 :103,133,167,141,138,198,224
49782 :162,000,070,167,070,167,242
49788 :144,071,169,032,141,005,174
49794 :212,169,246,141,006,212,092
49800 :169,129,141,004,212,169,192
49806 :234,157,249,007,160,010,191
49812 :173,027,212,157,040,208,197
49818 :140,001,212,165,162,197,007
49824 :162,240,252,136,208,238,116
49830 :189,236,197,157,040,208,169
49836 :169,233,157,249,007,169,132
49842 :128,141,004,212,138,072,105
49848 :189,105,198,073,255,074,054
49854 :074,074,032,201,196,104,103
49860 :170,232,224,007,208,176,189
49866 :173,138,198,073,255,045,060
49872 :021,208,141,021,208,076,115
49878 :226,194,238,088,198,173,051
49884 :088,198,201,021,208,057,225
49890 :160,000,140,088,198,169,213
49896 :032,145,251,173,031,208,048
49902 :173,030,208,173,021,208,027
49908 :041,254,208,035,173,139,070
49914 :198,240,003,206,139,198,210
49920 :160,038,162,023,024,032,183
49926 :240,255,173,139,198,073,060
49932 :015,170,169,000,032,205,091
49938 :189,169,100,032,201,196,137
49944 :076,224,193,174,139,198,004
49950 :160,000,200,208,253,202,029
49956 :208,250,138,010,168,185,227
49962 :091,198,153,002,208,189,115
49968 :105,198,153,003,208,185,132
49974 :092,198,133,168,056,138,071
49980 :168,200,200,169,000,042,071
49986 :136,208,252,133,167,073,011
49992 :255,045,016,208,164,168,160
49998 :240,002,005,167,141,016,137
50004 :208,232,224,007,208,204,143
50010 :162,000,138,010,168,189,245
50016 :105,198,024,125,112,198,090
50022 :201,210,176,004,201,050,176
50028 :176,006,032,186,195,076,011
50034 :178,195,157,105,198,024,203
50040 :185,091,198,121,120,198,009
50046 :133,167,185,092,198,121,254
50052 :121,198,133,168,208,014,206
50058 :165,167,201,031,240,002,176
50064 :176,006,032,186,195,076,047
50070 :178,195,165,168,240,012,084
50076 :165,167,201,064,144,006,135
50082 :032,186,195,076,178,195,000
50088 :165,167,153,091,198,165,083
50094 :168,153,092,198,232,224,217
50100 :007,208,165,076,030,194,092
50106 :134,169,132,170,173,027,223
50112 :212,041,005,170,189,070,111
50118 :198,166,169,157,112,198,174
50124 :173,027,212,041,005,010,160
50130 :168,185,076,198,166,170,149
50136 :157,120,198,185,077,198,127
50142 :157,121,198,164,170,166,174
50148 :169,096,169,235,141,248,006
50154 :007,169,001,141,137,198,119
50160 :169,009,141,005,212,169,177
50166 :160,141,006,212,169,033,199
50172 :141,004,212,162,100,142,245
50178 :001,212,160,000,173,027,063
50184 :212,141,039,208,141,000,237
50190 :212,136,208,244,202,208,200
50196 :236,169,234,141,248,007,031
50202 :169,001,141,029,208,141,203
50208 :023,208,169,032,141,004,097
50214 :212,169,168,141,006,212,178
50220 :169,129,141,004,212,162,093
50226 :100,142,001,212,160,000,153
50232 :140,000,212,173,027,212,052
50238 :141,039,208,136,208,244,014
50244 :202,208,236,169,232,141,232
50250 :248,007,169,006,141,039,172
50256 :208,169,000,141,029,208,067
50262 :141,023,208,169,128,141,128
50268 :004,212,162,100,160,000,218
50274 :136,208,253,202,208,250,075
50280 :169,000,141,137,198,168,149
50286 :153,002,208,200,192,014,111
50292 :208,248,141,016,208,160,073
50298 :000,169,032,145,251,173,124
50304 :030,208,206,158,007,206,175
50310 :136,198,173,136,198,240,191
50316 :003,076,224,193,160,000,028
50322 :185,203,007,217,222,007,219
50328 :240,005,176,011,076,178,070
50334 :196,200,192,006,208,238,174
50340 :076,178,196,160,006,185,197
50346 :202,007,153,221,007,136,128
50352 :208,247,032,159,255,032,085
50358 :228,255,201,136,208,246,176
50364 :160,006,169,048,153,202,158
50370 :007,136,208,250,076,158,005
50376 :193,170,160,006,056,185,202
50382 :202,007,105,000,201,058,011
50388 :144,002,169,048,153,202,162
50394 :007,136,208,239,202,208,194
50400 :233,096,048,016,032,048,185
50406 :016,032,048,016,000,000,086
50412 :000,000,000,000,000,032,012
50418 :000,000,236,000,000,236,202
50424 :000,003,255,000,015,255,008
```

```
50430 :192,015,087,192,015,255,242
50436 :192,063,087,240,255,255,072
50442 :252,255,087,252,255,255,086
50448 :252,239,087,236,239,255,044
50454 :236,236,220,236,236,220,126
50460 :236,236,220,236,252,000,184
50466 :252,000,000,000,000,000,030
50472 :000,000,000,000,000,000,040
50478 :000,000,000,000,000,000,046
50484 :000,000,003,085,192,013,089
50490 :085,112,063,255,252,234,035
50496 :170,171,226,034,043,058,254
50502 :170,172,013,085,112,003,113
50508 :255,192,000,000,000,000,011
50514 :000,000,000,000,000,000,082
50520 :000,000,000,000,000,000,088
50526 :000,000,000,000,000,000,094
50532 :000,000,000,000,000,000,100
50538 :128,032,008,032,160,032,242
50544 :000,168,160,034,162,160,028
50550 :138,168,130,162,197,042,187
50556 :040,115,008,131,190,224,064
50562 :046,188,162,011,127,064,216
50568 :130,201,088,010,186,074,057
50574 :035,190,232,000,141,032,004
50580 :010,038,010,038,166,160,058
50586 :000,168,168,130,138,040,030
50592 :010,003,128,008,032,000,085
50598 :000,002,032,000,000,000,200
50604 :000,000,000,000,000,016,188
50610 :000,000,118,000,000,118,158
50616 :000,001,255,128,007,255,062
50622 :224,007,171,224,007,255,054
50628 :224,031,171,248,127,255,228
50634 :254,127,171,254,127,255,110
50640 :254,119,171,246,119,255,092
50646 :246,118,110,118,118,110,010
50652 :118,118,110,118,126,000,042
50658 :126,000,000,000,000,000,096
50664 :000,000,000,013,006,004,255
50670 :001,007,008,012,013,014,037
50676 :008,031,211,067,079,082,210
50682 :069,058,158,048,048,048,167
50688 :048,048,048,032,032,149,101
50694 :200,073,071,072,032,211,153
50700 :067,079,082,069,058,028,139
50706 :048,048,048,048,048,048,050
50712 :000,018,149,204,073,086,042
50718 :069,083,058,051,029,029,093
50724 :029,029,029,029,029,028,209
50730 :193,083,084,082,079,045,096
50736 :208,193,206,201,195,033,060
50742 :029,029,029,029,029,029,228
50748 :031,204,069,086,069,076,083
50754 :058,048,032,000,001,255,204
50760 :002,254,003,253,001,000,073
50766 :255,255,002,000,254,255,075
50772 :003,000,253,255,013,013,109
```

Typing Derby

(Article on page 86.)

Program 1:

Typing Derby—VIC Version

```
2 PRINTCHR$(147)"{5 RIGHT}{5 DOWN}{RVS}
  {RED}TYPING DERBY{OFF}":PRINT"{6 DOWN}
  {RIGHT}INSTRUCTIONS? (Y=YES)     :rem 32
4 GETA$:IFA$=""THEN4               :rem 139
```

BEFORE TYPING...

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

```
6 IFA$="Y"THENGOSUB401:PRINT"{CLR}"
                                   :rem 230
8 PRINT"{HOME}{15 DOWN}{RIGHT}ENTER LEVEL
  :(1TO13)":INPUTL1:IFL1<1ORL1>13THEN4
                                   :rem 14
10 DIMC(3):C(0)=38488:C(1)=38554:C(2)=386
   20:H=7768:H1=7790:J=8010:R=32 :rem 235
11 L=L1:S=(220*(L1-1))+(10*-(L1>1)):GOTO2
   0                               :rem 179
16 PRINT"{CLR}{3 DOWN}{3 RIGHT}CONT'(Y/N)
   ?"                              :rem 22
17 GETX$:IFX$=""THEN17             :rem 33
18 IFX$="N"THENEND                 :rem 73
20 IFS=2880THENPRINT"{4 RIGHT}THE END":EN
   D                               :rem 144
21 M=0:N=0:N1=0:IFS>220*LTHENL=L+1:L1=L1+
   1:R=32                          :rem 247
22 PRINTCHR$(147):POKE36879,219:PRINT"
   {BLK}SCORE:{RVS}"S"{OFF} LEVEL{RVS}"L1
   "{OFF}"                         :rem 138
30 FORI=0TO3:PRINT"{DOWN}{22 R}":NEXT:P
   RINT"{3 UP}{2 LEFT}{RED}{*}{DOWN}
   {LEFT}{M}"                      :rem 255
40 PRINT"{10 DOWN}{5 RIGHT}{RVS}TYPING DE
   RBY{OFF}{BLK}";                 :rem 48
50 FORI=1TOL:READD$:NEXT:RESTORE:PRINT"
   {HOME}{15 DOWN}{BLU}"D$         :rem 43
52 IFL=6THENPOKE8015,44:POKE8067,44
                                   :rem 71
53 IFL=7THENPOKE8023,44            :rem 118
54 IFL=8THENFORI=0TO6STEP2:POKE8010+I,58:
   NEXT                            :rem 132
55 IFL=8THENPOKE8031,44:POKE8070,58:POKE8
   072,58                          :rem 28
56 IFL=9THENPOKE8046,58            :rem 133
60 FORI=0TO2:FORT=0TO21:POKEC(I)+T,2:NEXT
   T:NEXTI:POKEH,94                :rem 188
70 FORI=0TO2:FORT=0TO21:POKEC(I)+22+T,0:N
   EXTT:NEXTI:POKEH1,94            :rem 123
80 IFN=0ANDM=0ORS=(220*L)+10THENGOSUB300
                                   :rem 153
90 IFM=21ORM=87THENM1=M:M=M+44     :rem 90
100 IFPEEK(H1+M+1)<>32THEN15       :rem 87
110 IFTI>T+RTHENPOKEH1+M1,32:POKEH1+M,32:
    M=M+1:POKEH1+M,94:T=TI         :rem 145
120 GETA$:IFA$=""THEN90            :rem 31
130 IFASC(A$)=PEEK(J+N1)THEN150    :rem 28
140 IFASC(A$)<>PEEK(J+N1)+64THEN90
                                   :rem 194
150 POKE38730+N1,2:N1=N1+1:IFN=21ORN=87TH
    ENN2=N:N=N+45                  :rem 214
160 POKEH+N,32:POKEH+N2,32:N=N+1:IFPEEK(H
    +N)<>32THENS=S+10:R=R-1:GOTO16:rem 68
170 POKEH+N,94:GOTO90              :rem 253
200 DATAFRF FTF FGF FBF FVF FRF FTF FGF F
    BF FVF FRF FTF FGF FBF FVF FR  :rem 7
210 DATADED DCD FRF FTF FGF FBF FVF DED D
    CD FRF FTF FGF FBF FVF DED DC:rem 179
220 DATASWS SXS DED DCD FRF FTF FGF FBF F
    VF SWS SXS DED DCD FRF FTF FG :rem 96
230 DATAAQA AZA SWS SXS DED DCD FRF FTF F
    GF AQA AZA SWS SXS DED DCD FR :rem 92
```

```
240 DATAJUJ JYJ JHJ JNJ JMJ AQA AZA SWS S
    XS DED DCD JUJ JYJ JHJ JNJ JM:rem 146
250 DATAKIK KIK JUJ JYJ JHJ JNJ JMJ AQA A
    ZA SWS SXS DED DCD KIK KIK FR:rem 124
260 DATALOL L.L KIK KIK JUJ JYJ JHJ JNJ J
    MJ AQA AZA SWS SXS FTF LOL L.:rem 125
270 DATA;P; ;/; LOL L.L KIK KIK JUJ JYJ J
    HJ JNJ JMJ AQA AZA SWS SXS ;P :rem 51
280 DATAA11 S22 D33 F44 F55 J66 J77 K88 L
    99 ;ØØ Z11 X22 C33 V44 V55 N6:rem 187
281 DATAIF IF IF{2 SPACES}IT IT IT
    {2 SPACES}IS IS IS TIME TIME TIME IF
    {SPACE}IT IS TIME IF I          :rem 105
282 DATAWE WE WE{2 SPACES}CAN CAN CAN
    {2 SPACES}PLAY PLAY PLAY WE WE WE CAN
     CAN CAN PLA                    :rem 201
287 DATATHAT LITTLE BROWN FOX QUICKLY RUN
    S AND JUMPS OVER THE LAZY DOG :rem 50
290 DATATHIS RACE WILL END THE GAME; IF Y
    OUR TYPING DOES NOT FAIL. BYE :rem 76
300 V=36878:S2=36875:POKEV,15:POKES2,173:
    FORX=1TO150:NEXT:POKEV,Ø        :rem 249
330 POKEV,15:POKES2,194:FORX=1TO150:NEXT:
    POKEV,Ø                         :rem 25
340 POKEV,15:POKES2,206:FORX=1TO150:NEXT:
    POKEV,Ø                         :rem 20
350 FORI=ØTO2:POKEV,15:POKES2,214:FORX=1T
    O150:NEXT:POKEV,Ø:NEXT          :rem 57
360 FORI=ØTO2:POKEV,15:POKES2,206:FORX=1T
    O150:NEXT:POKEV,Ø:NEXT          :rem 59
370 POKEV,15:POKES2,194:FORX=1TO150:NEXT:
    POKEV,Ø                         :rem 29
380 POKEV,15:POKES2,206:FORX=1TO150:NEXT:
    POKEV,Ø                         :rem 24
385 POKEV,15:POKES2,194:FORX=1TO150:NEXT:
    POKEV,Ø                         :rem 35
390 POKEV,15:POKES2,173:FORX=1TO1800:NEXT
    :POKEV,Ø:RETURN                 :rem 105
401 PRINTCHR$(14)CHR$(147)"{5 RIGHT}{RVS}
    {RED}TYPING DERBY{OFF}{BLU}"  :rem 204
402 PRINT"{2 RIGHT}BASIC TOUCH TYPING
    {4 SPACES}TUTOR":PRINT"{RVS}{DOWN}INS
    TRUCTIONS{OFF}:"                :rem 174
403 PRINT"{RVS}{DOWN}1{OFF}.LEARN FINGERS
    ' RANGE ON THE KEYBOARD."       :rem 13
404 PRINT"{RVS}2{OFF}.PLACE FINGERS ON TH
    E'HOME KEYS'.WRISTS LE-VEL,FINGERS SL
    IGHTLY";                        :rem 181
405 PRINT"{2 SPACES}ARCHED,PALMS OFF VIC.
    "                               :rem 37
406 PRINT"{RVS}3{OFF}.TYPE THE EXERCISES
    {2 SPACES}WITHOUT LOOKING AT THE KEYB
    OARD."                          :rem 250
407 PRINT"{RVS}4{OFF}.AT FIRST,ACCURACY I
    S BETTER THAN SPEED."           :rem 178
408 PRINT"{RVS}5{OFF}.BEAT THE BLACK HORS
    E 23 TIMES AND MOVE ON TO THE NEXT LE
    VEL."                           :rem 198
409 PRINT"{DOWN}PRESS A KEY TO GO ON "
                                    :rem 140
410 GETA$:IFA$=""THEN410            :rem 77
411 PRINTCHR$(147)CHR$(142)"{RVS}{RED}
    {5 RIGHT}{DOWN}TYPING DERBY{BLK}{OFF}
    {BLU}"                          :rem 160
412 POKE36879,232                   :rem 154
414 PRINT"{DOWN}{3 RIGHT}{WHT}{RVS}Q
    {13 RIGHT}Q":PRINT"{RVS}{2 RIGHT}Q3Q
    {11 RIGHT}Q8Q"                  :rem 73
415 PRINT"{RVS}{2 RIGHT}2E4{11 RIGHT}6I9"
                                    :rem 89
416 PRINT"{RIGHT}{RVS}QW{RED}D{WHT}5
    {11 RIGHT}7{RED}K{WHT}OQ":PRINT"
    {RIGHT}{RVS}1{RED}S{WHT}CR{11 RIGHT}Y
    ,{RED}L{WHT}Ø                   :rem 18
418 PRINT"{RIGHT}{RVS}QX TQ{9 RIGHT}QU .P
    ":PRINT"{RIGHT}{RVS}{RED}A{WHT}
    {2 SPACES}{RED}F{WHT} {9 RIGHT}{BLK}S
    {WHT}H{2 SPACES}{RED}:{WHT}" :rem 167
420 PRINT"{RIGHT}{RVS}Z{2 SPACES}G{WHT}
    {9 RIGHT}{BLK}P{RED}J{WHT}{2 SPACES}/
    ":PRINT"{RIGHT}{RVS}{3 SPACES}V
    {9 RIGHT}{BLK}C{WHT}M{3 SPACES}"
                                    :rem 117
422 PRINT"{RIGHT}{RVS}{3 SPACES}B
    {9 RIGHT}{BLK}E{WHT}N{3 SPACES}":PRIN
    T"{RIGHT}{RVS}{4 SPACES}{OFF}£
    {9 RIGHT}[*]{RVS}{4 SPACES}"
                                    :rem 114
424 PRINT"{RIGHT}{RVS}[H]{3 SPACES}
    {11 RIGHT}{3 SPACES}[N]"        :rem 55
426 PRINT"{RIGHT}{DOWN}{RVS}LEFT
    {10 RIGHT}RIGHT":PRINT"{RIGHT}FINGER
    {SPACE}RANGE":PRINT"{RIGHT}{RVS}{BLK}
    SPACE BAR"                      :rem 211
428 PRINT"{RIGHT}{RVS}{RED}HOME KEYS{BLU}
    {OFF}":PRINT"{DOWN}PRESS ANY KEY TO G
    O ON";                          :rem 91
430 GETA$:IFA$=""THEN430            :rem 81
432 RETURN                          :rem 121
```

Program 2: Typing Derby—64 Version

```
Ø FORL=54272TO54296:POKEL,Ø:NEXTL:POKE532
  81,15:POKE53280,15                :rem 34
1 POKE54296,Ø:PRINT"{CLR}{8 DOWN}
  {14 RIGHT}{RVS}{RED}TYPING DERBY{OFF}
  {BLK}"                            :rem 35
2 PRINT"{5 DOWN}{9 RIGHT} INSTRUCTIONS (Y
  =YES)"                           :rem 234
4 GETA$:IFA$=""THEN4               :rem 139
6 IFA$="Y"THENGOSUB4ØØ              :rem 71
8 PRINT"{CLR}{1Ø DOWN}{4 RIGHT} ENTER LEV
  EL (1-13)";:INPUTL1               :rem 69
9 IFL1<1ORL1>13THEN8               :rem 217
1Ø C(Ø)=55456:C(1)=55576:C(2)=55696:H=118
  4:H1=1224:J=1624                  :rem 211
11 L=L1:S=(22Ø*(L1-1))+(1Ø*-(L1>1))
                                    :rem 222
14 PRINT"{CLR}{11 DOWN}{6 RIGHT} COMPUTER
    SPEED (5-5Ø)"SPC(58)"(5Ø IS THE SLOWE
  ST)";                             :rem 177
15 INPUTR1:R=R1                     :rem 197
16 IFR<5ORR>5ØTHEN14                :rem 227
17 GOTO21                            :rem 4
18 POKE198,Ø:PRINT"{CLR}{1Ø DOWN}
  {4 RIGHT} DO YOU WISH TO CONTINUE? (Y/
  N)";                              :rem 198
19 GETX$:IFX$=""ANDX$<>"N"ANDX$<>"Y"THEN1
  9                                 :rem 230
2Ø IFX$="N"THENEND                  :rem 66
21 M=Ø:N=Ø:N1=Ø:IFS>22Ø*LTHENL=L+1:L1=L1+
  1:R=R1:IFL>13THENPRINT"{CLR} THE END !
  "                                 :rem 56
22 PRINTCHR$(147):POKE53280,3:POKE53281,1
  :PRINT"{UP}{BLK}SCORE:{RVS}"S:IFL1>9TH
  EN24                              :rem 11
23 PRINTTAB(3Ø)"{UP}{OFF} LEVEL{2 RVS}"L1
  "{OFF}":GOTO3Ø                   :rem 169
24 PRINTTAB(3Ø)"{2 UP} LEVEL{RVS}"L1"
  {OFF}"                            :rem 193
```

```
30 FORI=1144TO1183:POKEI,114:POKEI+54272,
   0:NEXTI                              :rem 125
31 FORI=1264TO1303:POKEI,91:POKEI+54272,0
   :NEXTI                               :rem 79
32 FORI=1384TO1423:POKEI,91:POKEI+54272,0
   :NEXTI                               :rem 86
33 FORI=1504TO1543:POKEI,113:POKEI+54272,
   0:NEXTI                              :rem 127
34 POKE1502,103:POKE55774,0:POKE1462,95:P
   OKE55734,2                           :rem 142
35 FORI=1TO3*L-2:READD$:NEXTI:RESTORE:PRI
   NT"{HOME}{15 DOWN}{BLU}"D$           :rem 51
50 FORI=1TO3*L-1:READD$:NEXTI:RESTORE:PRI
   NT"{HOME}{16 DOWN}{BLU}"D$           :rem 64
51 FORI=1TO3*L:READD$:NEXTI:RESTORE:PRINT
   "{HOME}{17 DOWN}{BLU}"D$             :rem 244
53 IFL=7THENPOKE1646,44                 :rem 122
54 IFL=8THENFORI=0TO6STEP2:POKE1624+I,58:
   NEXTI                                :rem 240
55 IFL=8THEN:POKE1663,44:POKE1730,58:POKE
   1732,58                              :rem 82
56 IFL=9THENPOKE1689,58                 :rem 139
60 FORI=0TO2:FORT=0TO39:POKEC(I)+T,2:NEXT
   T:NEXTI:POKEH,94                     :rem 197
70 FORI=0TO2:FORT=0TO39:POKEC(I)+40+T,0:N
   EXTT:NEXTI:POKEH1,94                 :rem 132
80 IFN=0ANDM=0ORS=(300*L)+10THENGOSUB300
                                        :rem 152
90 IFM=39ORM=159THENM1=M:M=M+80          :rem 147
100 IFPEEK(H1+M+1)<>32THEN18            :rem 90
110 IFTI>T+RTHENPOKEH1+M1,32:POKEH1+M,32:
    M=M+1:POKEH1+M,94:T=TI              :rem 145
120 GETA$:IFA$=""THEN90                 :rem 31
130 IFASC(A$)=PEEK(J+N1)THEN150         :rem 28
140 IFASC(A$)<>PEEK(J+N1)+64THEN90
                                        :rem 194
150 POKE55896+N1,2:N1=N1+1:IFN=39ORN=159T
    HENN2=N:N=N+81                      :rem 27
160 POKEH+N,32:POKEH+N2,32:N=N+1:IFPEEK(H
    +N)<>32THENS=S+10:R=R-1:GOTO18:rem 70
170 POKEH+N,94:GOTO90                   :rem 253
200 DATAFRF FTF FGF FBF FVF FRF FTF FGF F
    BF FVF                              :rem 46
201 DATAFRF FTF FGF FBF FVF FRF FTF FGF F
    BF FVF                              :rem 47
202 DATAFRF FTF FGF FBF FVF FRF FTF FGF F
    RTB                                 :rem 174
203 DATADED DCD FRF FTF FGF FBF FVF DED D
    CD FRF                              :rem 254
204 DATAFTF FGF FBF FVF DED DCD FRF FTF F
    GF FBF                              :rem 10
205 DATAFVF DED DCD FRF FTF FGF FBF FVF F
    DEV                                 :rem 145
206 DATASWS SXS DED DCD FRF FTF FGF FBF F
    VF SWS                              :rem 131
207 DATASXS DED DCD FRF FTF FGF FBF FVF S
    WS SXS                              :rem 133
208 DATADED DCD FRF FTF FGF FBF FVF SWS S
    CDX                                 :rem 188
209 DATAAQA AZA SWS SXS DED DCD FRF FTF F
    GF AQA                              :rem 91
210 DATAAZA SWS SXS DED DCD FRF FTF FGF A
    QA AZA                              :rem 92
211 DATASWS SXS DED DCD FRF FTF FGF AQA A
    ZQW                                 :rem 232
212 DATAJUJ JYJ JHJ JNJ JMJ AQA AZA SWS S
    XS DED                              :rem 155
213 DATADCD JUJ JYJ JHJ JNJ JMJ AQA AZA S
    WS SXS                              :rem 154
214 DATADED DCD JUJ JYJ JHJ JNJ JMJ AQA J
    UQX                                 :rem 217
215 DATAKIK KIK JUJ JYJ JHJ JNJ JMJ AQA A
    ZA SWS                              :rem 145
216 DATASXS DED DCD KIK KIK FRF JUJ JYJ J
    NJ JMJ                              :rem 126
217 DATAAQA AZA SWS SXS DED DCD KIK KIK K
    IZD                                 :rem 230
218 DATALOL L.L KIK KIK JUJ JYJ JHJ JNJ J
    MJ AQA                              :rem 104
219 DATAAZA SWS SXS FTF LOL L.L KIK KIK J
    UJ JYJ                              :rem 174
220 DATAJHJ JNJ JMJ AQA AZA SWS SXS FTF L
    ZJM                                 :rem 20
221 DATA;P; ;/; LOL L.L KIK KIK JUJ JYJ J
    HJ JNJ                              :rem 25
222 DATAJMJ AQA AZA SWS SXS ;P; ;/; LOL L
    .L KIK                              :rem 50
223 DATAKIK JUJ JYJ JHJ JNJ JMJ AQA AZA S
    K;Q                                 :rem 222
224 DATAA11 S22 D33 F44 F55 J66 J77 K88 L
    99 ;00                              :rem 150
225 DATAZ11 X22 C33 V44 V55 N66 A11 S22 D
    33 F44                              :rem 190
226 DATAF55 J66 J77 K88 L99 ;00 Z11 X22 C
    3F5                                 :rem 79
227 DATAIF IF{2 SPACES}IT IT IT{2 SPACES}
    IS IS IS TIME TIME IF I             :rem 180
228 DATAF IF IT IS TIME IF IT IS IS IS IT
     IT IT IS                           :rem 102
229 DATAIF IF IS IS TIME TIME TIME IT IT
    {SPACE}IFS                          :rem 182
230 DATAWE WE WE{2 SPACES}CAN CAN CAN
    {2 SPACES}PLAY PLAY PLAY WE         :rem 55
231 DATAWE WE WE CAN CAN CAN PLAY CAN PLA
    Y WE WE                             :rem 112
232 DATACAN CAN WE WE PLAY PLAY WE CAN
    {2 SPACES}CANN                      :rem 135
233 DATATHE LITTLE BROWN FOX QUICKLY RUNS
     AND JU                             :rem 178
234 DATAMPS OVER THE LAZY DOG.DOG IS LAZY
    .FOX IS                             :rem 117
235 DATA JUMPING.DOG IS SLEEPY AND VERY A
    NGRY                                :rem 222
236 DATATHIS RACE WILL END THE GAMES;IF Y
    OUR TYP                             :rem 123
237 DATAING DOES NOT FAIL.BYE.YOUR TYPING
     SHOULD                             :rem 1
238 DATA NOT FAIL. YOUR PRACTICE WAS RIGO
    ROUS                                :rem 241
300 V=54296:V1=54276:AD=54277:SR=54278:FH
    =54273:FL=54272                     :rem 120
301 POKEV,15:POKEFH,70:POKEFL,75 :rem 200
302 POKEAD,26:POKESR,178:FORT=1TO10:POKEV
    1,17                                :rem 34
303 FORQ=1TO150:NEXT:POKEV1,16:NEXT
                                        :rem 234
304 POKE54296,0:RETURN                  :rem 70
400 PRINT"{CLR}{DOWN}{13 RIGHT}{RVS}{RED}
    TYPING DERBY{OFF}{BLK}"             :rem 35
401 PRINT"{DOWN}{6 RIGHT}{RVS}{RED} BASIC
     TOUCH TYPING TUTOR{OFF}{RED}":rem 95
402 PRINT"{2 DOWN}{2 RIGHT}{RVS}1{OFF}
    {WHT} LEARN FINGERS' RANGE ON THE KEY
    -{8 SPACES}BOARD{RED}"              :rem 201
403 PRINT"{DOWN}{2 RIGHT}{RVS}2{OFF}{WHT}
     TYPE WITHOUT LOOKING AT THE KEYBOARD
    {RED}"                              :rem 250
404 PRINT"{DOWN}{2 RIGHT}{RVS}3{OFF}{WHT}
     AT FIRST, ACCURACY IS BETTER THAN
    {7 SPACES}SPEED{RED}"               :rem 62
405 PRINT"{DOWN}{2 RIGHT}{RVS}4{OFF}{WHT}
     BEAT THE BLACK HORSE 23 TIMES AND
```

```
    {7 SPACES}MOVE ON ";              :rem 54
406 PRINT"TO THE NEXT LEVEL":PRINT"
    {3 DOWN}{6 RIGHT}{RVS}{BLK} PRESS ANY
     KEY TO CONTINUE{OFF}"            :rem 44
407 GETF$:IFF$=""THEN407             :rem 99
650 PRINT"{CLR}":POKE53280,9:POKE53281,15
                                     :rem 205
660 PRINT"{CLR} {WHT}":PRINT"{9 SPACES}
    {RVS} Q {OFF}{16 SPACES}{RVS} Q "
                                     :rem 104
670 PRINT"{6 SPACES}{RVS} Q{M} 3{M} Q
     {OFF}{10 SPACES}{RVS} Q{M} 8{M}
    {SPACE}Q "                       :rem 111
680 PRINT"{6 SPACES}{RVS} 2{M} E{M} 4
     {OFF}{10 SPACES}{RVS} 6{M} I{M}
    {SPACE}9 "                       :rem 36
690 PRINT"{3 SPACES}{RVS} Q{N} W{N}
    {5}D{WHT}{M} 5 {OFF}{10 SPACES}
    {RVS} 7{N} {5}K{WHT}{M} O{M}
    {SPACE}Q "                       :rem 150
700 PRINT"{3 SPACES}{RVS} 1{M} {5}S
    {WHT}{M} C{M} R {OFF}{10 SPACES}
    {RVS} Y{M} ,{M} {5}L{WHT}{M}
    {SPACE}0 "                       :rem 92
710 PRINT"{3 SPACES}{RVS} Q{M} X{M}
    {2 SPACES}{M} T{M} Q {OFF}
    {4 SPACES}{RVS} Q{M} U{M}
    {2 SPACES}{M} .{M} P "           :rem 201
720 PRINT"{3 SPACES}{RVS} {5}A{WHT}
    {M}{7 SPACES}{5}F{WHT}{M}
    {3 SPACES}{OFF}{4 SPACES}{RVS} {BLK}S
    {WHT}{M} H{6 SPACES}{M} {5}:
    {WHT} "                          :rem 132
730 PRINT"{3 SPACES}{RVS} Z{M}
    {7 SPACES}G{M}{3 SPACES}{OFF}
    {4 SPACES}{RVS} {BLK}P{WHT}{M}
    {5}J{WHT}{6 SPACES}{M} / "       :rem 89
740 PRINT"{3 SPACES}{RVS}{10 SPACES}V
    {M}{3 SPACES}{OFF}{4 SPACES}{RVS}
    {BLK}C{WHT}{2 SPACES}M{10 SPACES}"
                                     :rem 68
750 PRINT"{3 SPACES}{RVS}{10 SPACES}B
    {4 SPACES}{OFF}{4 SPACES}{RVS} {BLK}E
    {WHT}{2 SPACES}N{10 SPACES}"     :rem 141
760 PRINT"{3 SPACES}{RVS}{14 SPACES}{OFF}
    £{4 SPACES}{*}{RVS}{14 SPACES}"
                                     :rem 172
770 PRINT"{3 SPACES}{*}{RVS}{12 SPACES}
    {OFF}£{6 SPACES}{*}{RVS}
    {12 SPACES}{OFF}£"               :rem 199
780 PRINT"{4 SPACES}{*}{RVS}{10 SPACES}
    {OFF}£{8 SPACES}{*}{RVS}
    {10 SPACES}{OFF}£"               :rem 200
790 PRINT"{5 SPACES}{RVS}{10 SPACES}{OFF}
    {10 SPACES}{RVS}{10 SPACES}"     :rem 39
800 PRINT"{5 SPACES}{RVS}{10 SPACES}{OFF}
    {10 SPACES}{RVS}{10 SPACES}"     :rem 31
810 PRINT:PRINT"{8 SPACES}{RVS}LEFT{OFF}
    {16 SPACES}{RVS}RIGHT"           :rem 144
820 PRINTTAB(14)"FINGER RANGE":PRINT"
    {DOWN}{15 SPACES}{RVS}{BLK}SPACE BAR
    {OFF}"                           :rem 177
830 PRINT"{15 SPACES}{RVS}{5}HOME KEYS
    {OFF}"                           :rem 13
840 PRINT"{DOWN}{7 RIGHT}{WHT}PRESS ANY K
    EY TO CONTINUE";                 :rem 239
850 GETQ$:IFQ$=""THEN850             :rem 125
860 RETURN                           :rem 126
```

Cassette Cataloger

(Article on page 98.)

BEFORE TYPING...

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

```
10 FORI=707TO725:READA:POKEI,A:CK=CK+A:NE
   XT:IFCK<>2384THENPRINT"{WHT}DATA ERROR
   "                                 :rem 38
11 IFCK<>2384THENSTOP                :rem 63
12 BC$=CHR$(14):SYS65517:IFPEEK(781)=40TH
   ENTYPE=64:GOTO14                  :rem 177
13 POKE36879,8:B$="{CLR}":M1=36876:POKE36
   876+2,15:GOTO20                   :rem 250
14 SO=54272:FORT=SOTOSO+24:POKET,0:NEXT
                                     :rem 4
15 POKE53280,0:POKE53281,0:POKESO+24,15:P
   OKESO+5,17:B$="{CLR}{8 RIGHT}":rem 132
16 POKESO+6,250:POKESO,100:POKESO+1,160
                                     :rem 78
20 DIM LOC(50),N$(50),BYTES(50):I=1:J$=CH
   R$(16):CR$=CHR$(13)               :rem 2
25 M$="CASSETTE CATALOG"+CR$         :rem 148
30 LC$=CHR$(15):PRINTB$"{CYN}{RVS}{DOWN}-
   ---TAPE DIRECTORY---{OFF}"        :rem 23
40 INPUT"{WHT}{HOME}{4 DOWN}HARDCOPY (Y/N
   )";H$:IF H$="←" THENEND           :rem 215
50 IFH$="Y"THENINPUT"{RVS}{DOWN}{YEL}PRIN
   TER ON{OFF}";H$:H=1:GOSUB900:OPEN4,4,7
                                     :rem 158
55 S$="ONE":PRINT"{DOWN}{WHT}CASSETTE NAM
   E:";:INPUTCN$:INPUT"{DOWN}SIDE 1 OR 2"
   ;S                                :rem 172
60 INPUT"{DOWN}DATE:";DA$:IFS=2THENS$="TW
   O"                                :rem 51
65 CN$=CN$+"/"+S$                    :rem 82
70 PRINT"{RVS}{DOWN}REWIND TAPE.{OFF}": P
   RINT"{RVS}{DOWN}SET COUNTER.{OFF}": PR
   INT"{RVS}{DOWN}PRESS F7 KEY.{OFF}{YEL}
   "                                 :rem 135
80 GETA$:IFA$<>CHR$(136)THEN80       :rem 216
100 GOSUB900:CLOSE1:IFST=-128THEN700
                                     :rem 198
110 SYS 707:IFST=-128THEN700         :rem 106
120 B1=PEEK(829)+256*PEEK(830):B2=PEEK(83
    1)+256*PEEK(832):BYTES(I)=B2-B1
                                     :rem 13
130 IFI=1THENM$=M$+CN$+CR$+DA$:PRINT"
    {CLR}{DOWN}"M$                   :rem 224
140 IFI=1THENIFH=1THENPRINT#4,BC$;M$;LC$;
    CR$                              :rem 197
200 L$="":GOSUB900:GOSUB900:INPUT"{PUR}
    {RVS}COUNTER{OFF}:";L$:IFL$=""THEN200
                                     :rem 247
210 L=VAL(L$)                        :rem 192
220 IFI=1THENLOC(2)=L:LOC(1)=0:GOTO300
                                     :rem 0
230 LOC(I+1)=L                       :rem 242
300 FORX=0TO15:N$(I)=N$(I)+CHR$(PEEK(833+
    X)):NEXT                         :rem 100
350 PRINT"{RED}{RVS}"LOC(I);TAB(4);"{WHT}
    "N$(I)"{2 RIGHT}{GRN}{RVS}"BYTES(I)"
    {LEFT} BYTES.":GOSUB900          :rem 238
360 IFH=1THENGOSUB920                :rem 40
```

```
370 IFL$="<"THENCLOSE1:GOTO700    :rem 89
400 I=I+1:GOTO100                 :rem 195
700 PRINT"{WHT}I/O STATUS="ST:GOSUB900:CL
    OSE1:LM=I:IFN$(LM)=""THENLM=I-1
                                  :rem 219
800 IFH=1THENPRINT#4:CLOSE4:H=0   :rem 195
810 D$="":INPUT"{WHT}DISPLAY Y/N/H";D$:IF
    D$="N"THENEND                 :rem 161
820 IFD$="H"THEN CLOSE4:OPEN4,4:PRINT#4,B
    C$;M$;LC$:H=1                 :rem 176
830 PRINT"{CLR}{GRN}{RVS}"CN$:FOR I=1TOLM
    :PRINT"{YEL}{RVS}"LOC(I);TAB(4)"{WHT}
    "N$(I)"{6 RIGHT}{GRN}{RVS}"BYTES(I)
                                  :rem 90
840 IF D$="H"THENGOSUB920         :rem 166
845 IFI/10=INT(I/10)THENINPUT"{WHT}MORE Y
    /N";M$:IFM$="N"THEN800        :rem 232
850 IFI/10=INT(I/10)THENINPUT"{WHT}MORE Y
    /N";M$:IFM$="N"THEN810        :rem 229
860 NEXT:GOTO800                  :rem 232
900 IFTYPE=64THEN910              :rem 218
905 POKEM1,232:FORS=1TO50:NEXT:POKEM1,0:R
    ETURN                         :rem 206
910 POKESO+4,17:POKESO+4,16:RETURN
                                  :rem 117
920 PRINT#4,LOC(I);J$+"06";N$(I);J$+"24";
    BYTES(I)J$"29"+"BYTES.":RETURN:rem 29
1000 DATA 169,1,170,160,0,32,186,255,169,
     0,32,189,255,169,1,32,213,255,96
                                  :rem 13
```

Homonym Practice

(Article on page 102.)

BEFORE TYPING...

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

Program 1:
Homonym Practice—VIC And 64 Version

```
2 PRINTCHR$(14):CH$="1":PRINT"{CLR}{DOWN}
  HI, I'M VIC,","{DOWN} WHAT'S YOUR NAME
  ?{4 DOWN}"                      :rem 106
3 GETC$:IFC$=""THEN3              :rem 141
4 N$=CHR$(ASC(C$)OR128)           :rem 0
5 PRINTN$;                        :rem 111
6 GETC$:IFC$=""THEN6              :rem 147
7 IFASC(C$)=13THENN$=N$+"{4 SPACES}":GOTO
  10                              :rem 137
8 IFASC(C$)=133THEN2              :rem 139
9 N$=N$+C$:PRINTC$;:GOTO6         :rem 254
10 PRINT"{CLR}{DOWN} HI, ";N$:PRINT"
   {3 DOWN} TODAY WE'LL PRACTICE":PRINT"
   {DOWN} SOME HOMONYMS."         :rem 157
11 GOSUB 51:PRINT "{CLR}{DOWN} WOULD YOU
   {SPACE}LIKE TO","{DOWN} PRACTICE USING
   ","{2 DOWN}{2 SPACES}1) TO{2 SPACES}TW
   O";                            :rem 221
12 PRINT"{2 SPACES}TOO","{3 DOWN} OR",,"
   {3 DOWN}{2 SPACES}2) THERE{3 SPACES}TH
   EIR {DOWN}{11 SPACES}THEY'RE" :rem 145
13 PRINTTAB(12)"{2 DOWN}? ";      :rem 92
14 GET CH$:IF CH$=""THEN14        :rem 129
15 PRINTCH$:GOSUB51:PRINT"{CLR}{DOWN} IF
   {SPACE}YOU WANT TO SEE","{DOWN} EXAMPL
   ES";                           :rem 82
16 PRINT" OF EACH","{DOWN} WORD USED IN A
   ","{DOWN} SENTENCE,"           :rem 166
17 PRINT"{DOWN} JUST PRESS THE","{DOWN} B
   ROWN BUTTON","{DOWN} MARKED {RVS} F3
   {OFF} .":GOSUB51               :rem 131
18 PRINT"{CLR}{DOWN} YOU MAY USE THE","
   {DOWN} BROWN {RVS} F3 {OFF} BUTTON","
   {DOWN} ANYTIME YOU NEED IT."   :rem 142
19 GOSUB51:PRINT"{CLR}{DOWN} YOU MUST TYP
   E ","{DOWN} THE WORD THAT","{DOWN} GOE
   S IN THE *** ."                :rem 169
20 PRINT"{3 DOWN} PRESS {RVS} RETURN
   {OFF}","{DOWN} AFTER EACH ANSWER.":GOS
   UB51                           :rem 25
21 SC=0:G=0:S=INT((RND(1)*10)+2)  :rem 249
22 W=S:IFASC(CH$)=50THENW=S+16    :rem 53
23 RESTORE:FORT=1TOW:READA$,B$:NEXTT
                                  :rem 128
24 READA$,B$:S=S+1:IFS>17THENS=1:GOTO22
                                  :rem 123
25 C=C+1:IFC>2THENC=0:GOTO24      :rem 186
26 IFASC(C$)=134THENGOSUB42       :rem 112
27 PRINT"{CLR}{5 DOWN}";A$        :rem 194
28 PRINT"{HOME}{14 DOWN}{4 SPACES}*** = "
   ;                              :rem 50
29 GET C$:IF C$=""THEN 29         :rem 253
30 IFASC(C$)=55THENC$="'"         :rem 102
31 IFASC(C$)=13THEN36             :rem 187
32 IFASC(C$)=133THEN2             :rem 184
33 IFASC(C$)=134THEN26            :rem 240
34 IFASC(C$)=20THENAN$=LEFT$(AN$,LEN(AN$)
   -1):PRINTC$;:GOTO29            :rem 74
35 PRINTC$;:AN$=AN$+C$:GOTO29     :rem 228
36 IFAN$=B$THENPRINT"{HOME}{DOWN}VERY GOO
   D,";N$:SC=SC+1:FORT=1TO800:NEXTT:AN$="
   ":GOTO39                       :rem 113
37 PRINT"{HOME}{DOWN}SORRY, TRY AGAIN.":A
   N$="":SC=SC-1                  :rem 166
38 PRINT"{HOME}{14 DOWN}{19 SPACES}":GOTO
   28                             :rem 26
39 G=G+1:IFG<10THEN24             :rem 213
40 PRINT"{CLR}{DOWN} ";N$:PRINT"{2 DOWN}
   {SPACE}YOU GOT "SC" RIGHT","{DOWN}
   {2 SPACES}OUT OF TEN."         :rem 149
41 PRINT"{4 DOWN}{3 SPACES}THAT'S
   {2 SPACES}";100-((10-SC)*10);"%":GOSUB
   51:GOTO2                       :rem 88
42 IFASC(CH$)=50THEN47            :rem 8
43 PRINT"{CLR}{DOWN} TWO",,"{DOWN}
   {4 SPACES}I HAVE TWO TOYS."    :rem 231
44 PRINT"{2 DOWN} TOO",,"{DOWN}{4 SPACES}
   HE ATE TOO MUCH."              :rem 46
45 PRINT"{2 DOWN} TOO",,"{DOWN}{4 SPACES}
   I WANT SOME,TOO."              :rem 126
46 PRINT"{2 DOWN} TO",,"{DOWN}{4 SPACES}G
   O TO THE STORE.","{DOWN}{4 SPACES}I WA
   NT TO SEE IT.":GOTO50          :rem 169
47 PRINT"{CLR}{2 DOWN} THERE",,"{DOWN}
   {4 SPACES}THE BOOK IS OVER{10 SPACES}T
   HERE."                         :rem 32
48 PRINT"{2 DOWN} THEIR",,"{DOWN}
   {4 SPACES}THEY LOST THEIR{12 SPACES}HA
   TS."                           :rem 94
49 PRINT"{2 DOWN} THEY'RE",,"{DOWN}
   {4 SPACES}THEY'RE GOING HOME{8 SPACES}
   NOW."                          :rem 60
50 AN$="":GOSUB51:RETURN          :rem 211
```

```
51 PRINT"{HOME}{21 DOWN}{4 SPACES}{RVS} P
   RESS RETURN {OFF}{2 SPACES}"  :rem 192
52 GETT$:IFT$=""THEN52           :rem 23
53 IFASC(T$)=134THENGOSUB42      :rem 129
54 IFASC(T$)=133THEN2            :rem 205
55 RETURN                        :rem 74
56 DATA1,1                       :rem 19
57 DATA"THAT'S WAY *** MUCH!",TOO:rem 208
58 DATA"I HAD *** MUCH TO EAT{2 SPACES}
   {DOWN}LAST NIGHT.",TOO        :rem 148
59 DATA"WE'RE GOING *** FAST!",TOO :rem 5
60 DATA"LET'S GO OVER *** MY{2 SPACES}
   {DOWN} HOUSE.",TO             :rem 43
61 DATA"MARY WANTS *** COME{4 SPACES}
   {DOWN} OVER HERE.",TO         :rem 27
62 DATA"I DON'T KNOW HOW ***{2 SPACES}
   {DOWN} DO THIS ONE.",TO       :rem 81
63 DATA"PETER THINKS THAT IT'S {DOWN} ***
    FAR TO WALK.",TOO            :rem 102
64 DATA"THERE ARE *** TIGERS{2 SPACES}
   {DOWN} IN THE ZOO.",TWO       :rem 185
65 DATA"WHAT IS *** TIMES{5 SPACES}{DOWN}
    SIXTY-FOUR?.",TWO            :rem 199
66 DATA"WHERE IS TRUDY GOING{3 SPACES}
   {DOWN}*** LOOK FOR IT?",TO    :rem 67
67 DATA"LATONIA WOULD LIKE{4 SPACES}
   {DOWN} SOME ICE CREAM, ***.",TOO
                                 :rem 119
68 DATA"LITTLE JIM CAN COME{3 SPACES}
   {DOWN} ALONG, ***.",TOO       :rem 232
69 DATA"I HOPE THERE WON'T{4 SPACES}
   {DOWN} BE *** MANY.",TOO      :rem 64
70 DATA"HOW MUCH WOULD ***{4 SPACES}
   {DOWN} HAMBURGERS COST?",TWO  :rem 8
71 DATA"THIS WORK IS *** HARD {DOWN} FOR
   {SPACE}ALISA.",TOO            :rem 40
72 DATA"CAN MARK GO TO THE{4 SPACES}
   {DOWN} PARTY, *** ?",TOO      :rem 134
73 DATA"ARE THOSE YOUR BOOKS{2 SPACES}
   {DOWN} OVER ***?",THERE       :rem 30
74 DATA"CAN WE PLAY AT ***{4 SPACES}
   {DOWN} HOUSE?",THEIR          :rem 124
75 DATA"I'M SURE THAT *** NOT {DOWN} HOME
    YET.",THEY'RE                :rem 100
76 DATA"THE CHILDREN PUT ***{2 SPACES}
   {DOWN} BOOKS AWAY.",THEIR     :rem 131
77 DATA"TOM AND SUE SAID ***{2 SPACES}
   {DOWN} COMING LATER.",THEY'RE :rem 179
78 DATA"THE BOYS LOST ***{5 SPACES}{DOWN}
    BALL.",THEIR                 :rem 45
79 DATA"IS KIM SURE THAT ***{2 SPACES}
   {DOWN} COMING TONIGHT?",THEY'RE
                                 :rem 135
80 DATA"IS *** A DRAGON IN{4 SPACES}
   {DOWN} THE CLOSET?",THERE     :rem 146
81 DATA"CAN YOU SEE *** BIG{3 SPACES}
   {DOWN} BLUE EYES?",THEIR      :rem 153
82 DATA"PAUL AND TOM ARE ON{3 SPACES}
   {DOWN} *** WAY.",THEIR        :rem 136
83 DATA"I THINK THAT *** TOO{2 SPACES}
   {DOWN} HIGH TO REACH.",THEY'RE :rem 58
84 DATA"THE BOYS LEFT ***{5 SPACES}{DOWN}
    JUNK ALL OVER!",THEIR        :rem 56
85 DATA"LOOK OVER ***.",THERE    :rem 12
86 DATA"I THINK *** GONE.",THEY'RE:rem 12
87 DATA"GIVE ME *** ADDRESS.",THEIR
                                 :rem 100
88 DATA"CAN MOLLY TAKE ***{4 SPACES}
   {DOWN} PLACE?",THEIR          :rem 45
89 DATA1,1                       :rem 25
```

Program 2: Screen Formatter

```
100 PRINT "{CLR}{4 SPACES}{RVS}22-COLUMN
    {SPACE}PRINT FORMATTER FOR C64":PRINT
                                 :rem 2
110 PRINT "READING DATA"         :rem 119
120 FORI=828TO881:READA:CK=CK+A:POKEI,A:N
    EXT:POKE179,883AND255        :rem 92
130 IF CK<>6032 THEN PRINT "ERROR IN DATA
    :CHECK TYPING.":END          :rem 227
140 PRINT"{DOWN}BEFORE...":SYS 828:PRINT"
    AFTER..."                    :rem 150
150 PRINT "{DOWN}PRESS RUN/STOP-RESTORE";
    :PRINT"TO REGAIN 40 COLUMNS" :rem 228
160 PRINT "{DOWN}ENTER {RVS}SYS 828{OFF}
    {SPACE}TO":PRINT"REACTIVATE, IF":PRIN
    T"NECESSARY."                :rem 115
170 PRINT "{DOWN}DO NOT EDIT ANY":PRINT"L
    INES WHILE IN 22 COL-UMN MODE."
                                 :rem 84
1000 DATA169,71,141,38,3,169,3,141
                                 :rem 180
1010 DATA39,3,96,72,152,72,138,72:rem 141
1020 DATA56,32,240,255,192,9,176,3
                                 :rem 185
1030 DATA76,100,3,192,31,144,15,169
                                 :rem 226
1040 DATA13,32,202,241,56,32,240,255
                                  :rem 9
1050 DATA160,9,24,32,240,255,104,170
                                 :rem 14
1060 DATA104,168,104,76,202,241  :rem 30
```

React

(Article on page 72.)

Program 1: React—VIC Version (Character Loader)

```
10 PRINT"{CLR}{WHT}":POKE36879,107 :rem 8
20 PRINT"DO YOU WANT TO CENTER THE SCREEN
   (Y/N)"                        :rem 41
30 GETA$:IFA$<>"Y"ANDA$<>"N"THEN30
                                 :rem 195
40 IFA$="Y"THENGOSUB10000        :rem 210
50 PRINT"{CLR}":POKE36879,8      :rem 167
60 PRINT"LOADING CHARACTERS{4 SPACES}PLEA
   SE WAIT"                      :rem 4
70 POKE52,28:POKE56,28:CLR       :rem 24
80 FORI=7168TO7679:POKEI,PEEK(I+25600):NE
   XT                            :rem 105
120 FORI=7168TO7175:READA:POKEI,A:NEXT:DA
    TA24,90,36,24,24,24,36,66    :rem 67
130 FORI=7384TO7551:READA:POKEI,A:NEXT
                                 :rem 123
140 DATA0,0,24,24,60,90,60,66    :rem 176
141 DATA60,66,165,129,165,154,66,60
                                 :rem 246
142 DATA255,129,165,129,189,165,189,255
                                 :rem 209
143 DATA0,126,126,126,126,126,126,0
                                 :rem 220
144 DATA170,255,85,255,170,255,85,255:DAT
    A0,0,0,0,0,0,0,0             :rem 105
145 DATA0,6,8,60,126,126,126,60  :rem 29
146 DATA24,60,126,255,255,255,60,60
                                 :rem 238
147 DATA56,60,254,255,255,254,60,56
                                 :rem 250
```

```
148 DATA60,60,255,255,255,126,60,24
                                    :rem 240
149 DATA28,60,127,255,255,127,60,28
                                    :rem 248
150 DATA0,0,56,40,8,20,34,127       :rem 174
151 DATA28,34,42,34,28,20,119,0     :rem 26
152 DATA219,195,36,153,153,36,195,219
                                    :rem 96
153 DATA75,75,75,75,180,180,180,180
                                    :rem 251
154 DATA126,102,66,126,24,36,195,66
                                    :rem 243
155 DATA60,66,189,255,255,189,66,60:rem 9
156 DATA0,0,56,68,146,68,56,0       :rem 199
157 DATA0,60,60,60,60,60,255,0      :rem 229
158 DATA129,195,231,255,255,231,195,129
                                    :rem 198
159 DATA0,102,102,0,66,102,60,24    :rem 62
160 FORI=7632TO7679:READA:POKEI,A:NEXT
                                    :rem 133
161 DATA171,171,173,173,181,181,213,213
                                    :rem 174
162 DATA195,195,0,24,24,0,195,195:rem 143
163 DATA255,145,145,159,249,137,137,255
                                    :rem 200
164 DATA0,0,151,146,242,146,151,0:rem 117
165 DATA165,36,231,24,24,231,36,165
                                    :rem 236
166 DATA195,231,126,60,60,126,231,195
                                    :rem 83
170 POKE198,5:POKE631,78:POKE632,69:POKE6
    33,87:POKE634,13:POKE635,131:END
                                    :rem 22
10000 PRINT"{CLR}CENTER SCREEN WITH
      {4 SPACES}CURSOR KEYS.{2 SPACES}WHE
      N{4 SPACES}FINISHED, HIT <RETURN>"
                                    :rem 65
10001 A=PEEK(197):B=PEEK(653):IFA=15THEN1
      0009                          :rem 42
10002 IFA=31THENA=36865:GOTO10005:rem 134
10003 IFA=23THENA=36864:GOTO10005:rem 135
10004 GOTO10001                     :rem 32
10005 IFB=1THENB=-1                 :rem 65
10006 IFB=0THENB=1                  :rem 20
10007 Q=PEEK(A):IFQ+B<0OR(Q+B>17ANDA=3686
      4)THEN10001                    :rem 3
10008 POKEA,Q+B:GOTO10001          :rem 184
10009 PRINT"{CLR}":RETURN          :rem 120
```

Program 2: React—VIC Version (Main Game)

```
5 POKE36878,15                      :rem 6
10 POKE36879,238:PRINTCHR$(147)CHR$(144)
                                    :rem 10
20 PRINT"RRRRR"                     :rem 205
21 FORI=1TO3:PRINT"R{3 SPACES}R":NEXT
                                    :rem 255
24 PRINT"RRRRR E A C T"             :rem 238
25 PRINT"RR"                        :rem 220
26 PRINT"R R"                       :rem 221
27 PRINT"R{2 SPACES}R"              :rem 222
28 PRINT"R{3 SPACES}R"              :rem 223
30 FORI=8TO248STEP16:FORJ=0TO7:POKE36879,
   I+J                              :rem 2
40 POKE36875,160+J+(I/16):NEXT:NEXT:POKE3
   6875,0:POKE36879,62              :rem 5
50 PRINT:PRINT"PRESS THE FIRE BUTTON TO S
   TART"                            :rem 158
109 CO=36879:A=7901:SM=28:ML=5      :rem 201
110 C=30720:ER=32:MA=0:TR=30:CH=33:WA=31:
   S=36875:V=36878:DD=37154:PA=37137
                                    :rem 213
120 PB=37152:OP=127:TF=255:N1=128:N2=8:N3
   =16:N4=4:NE=1:SC=0:DIMJS(2,2):NM=2:NC
   =1:CL=NC                         :rem 43
130 FORI=0TO2:FORJ=0TO2:READJS(J,I):NEXT:
   NEXT:DATA-23,-22,-21,-1,0,1,21,22,23
                                    :rem 250
135 GOSUB390:IFFR=0THEN135          :rem 75
136 POKES,230:FORI=1TO4:NEXT:POKES,0
                                    :rem 29
160 PRINTCHR$(147):POKECO-10,255:POKECO,8
                                    :rem 194
180 GOSUB430                        :rem 176
190 A=7703:FORI=1TONM               :rem 255
200 W=INT(RND(I)*505)+1:J=INT(RND(I)*6)+2
   :IFPEEK(7680+W)<>32THEN200       :rem 110
205 IFW+7680=ATHEN200               :rem 188
210 POKES,150:POKES+1,230:POKEW+7680,28:P
   OKEW+C+7680,J:POKES,0:POKES+1,0:NEXT
                                    :rem 175
220 FORI=1TONC                      :rem 102
230 W=INT(RND(I)*505)+1:J=INT(RND(I)*6)+2
   :K=INT(RND(I)*64)+1:IFPEEK(7680+W)<>3
   2THEN230                         :rem 224
235 IFK>26ANDK<33ORK=0ORW+7680=ATHEN230
                                    :rem 109
240 POKES,240:POKES+1,200:POKEW+7680,K:PO
   KEW+C+7680,J:POKES,0:POKES+1,0:NEXT
                                    :rem 144
249 TI$="000000"                     :rem 1
250 POKEA,MA:GOSUB400:B=A+JS(X+1,Y+1)
                                    :rem 183
260 IFPEEK(B)=WAORPEEK(B)=TRORPEEK(B)=SMT
   HEN560                           :rem 240
270 IFPEEK(B)=MATHEN290             :rem 117
280 IFPEEK(B)<>ERTHENPOKES,240:SC=SC+50:C
   L=CL-NE:POKES,0:IFCL=.THEN600:rem 141
290 POKES,200:POKES,0:SC=SC+NE:POKEA,TR:A
   =B:GOTO250                       :rem 212
390 P=PEEK(PA):FR=-((PAND32)=0):RETURN
                                    :rem 29
400 POKEDD,OP:S3=-((PEEK(PB)ANDN1)=MA):PO
   KEDD,TF                          :rem 190
410 P=PEEK(PA):S1=-((PANDN2)=MA):S2=((PAN
   DN3)=MA):S0=((PANDN4)=MA)        :rem 122
420 X=S2+S3:Y=S0+S1:RETURN          :rem 125
430 TC=INT(RND(I)*6)+2:FORI=7680TO8185:PO
   KEI+C,TC:NEXT                    :rem 51
440 FORI=7680TO7701:POKEI,WA:POKEI+C,NE:P
   OKEI+485,WA:POKEI+C+485,NE:NEXT:rem 7
450 FORI=7702TO8164STEP22:POKEI,WA:POKEI+
   C,NE:POKEI+21,WA:POKEI+C+21,NE:NEXT
                                    :rem 43
460 W=INT(RND(I)*5)+1               :rem 163
470 ONWGOSUB490,510,540              :rem 51
480 RETURN                          :rem 124
490 FORI=7795TO8095STEP22:POKEI+C,NE:POKE
   I,WA:POKEI+C+11,NE:POKEI+11,WA:NEXT
                                    :rem 60
500 RETURN                          :rem 117
510 FORI=7690TO7822STEP22:POKEI+C,NE:POKE
   I,WA:POKEI+C+NE,NE:POKEI+NE,WA:rem 21
520 POKEI+C+330,NE:POKEI+330,WA:POKEI+C+3
   31,NE:POKEI+331,WA:NEXT          :rem 134
530 FORI=7923TO7928:POKEI+C,NE:POKEI,WA:P
   OKEI+C+14,NE:POKEI+14,WA:NEXT:RETURN
                                    :rem 180
540 FORI=7712TO7822STEP22:POKEI+C,NE:POKE
   I,WA:POKEI+C+NE,NE:POKEI+NE,WA:NEXT
                                    :rem 140
```

```
550 FORI=7840TO8148STEP22:POKEI+C,NE:POKE
    I,WA:POKEI+C+9,NE:POKEI+9,WA:NEXT:RET
    URN                                 :rem 247
560 POKECO,15:IFPEEK(B)=28THENPOKEB,29
                                        :rem 77
570 POKEA,27:FORI=250TO115STEP-.3:POKES+2
    ,I:NEXT:POKES+2,0                   :rem 95
580 ML=ML-1:IFML=.THEN680               :rem 4
590 GOTO620                             :rem 111
600 POKEA,TR:POKEB,MA:FORI=1TO90:POKECO,I
    :POKES,150+I:POKES+1,150+I:POKES-1,15
    0+I                                 :rem 33
601 NEXT:POKES,0:POKES+1,0:POKES-1,0:POKE
    CO,152                              :rem 151
610 BO=100-VAL(TI$):IFBO<0THENBO=0:rem 11
620 PRINTCHR$(147):POKECO,104:PRINTCHR$(5
    ):SC=SC+BO                          :rem 124
630 PRINT"SCORE ";SC:PRINT              :rem 126
640 PRINT"MAYNERDS LEFT ";ML:PRINT:PRINT"
    BONUS ";BO                          :rem 242
650 PRINT:PRINT:PRINT:PRINTCHR$(15)"PRESS
     FIRE BUTTON TO{2 SPACES}CONTINUE"
                                        :rem 16
660 GOSUB390:IFFR=0THEN660              :rem 81
670 POKES,230:PRINTCHR$(147):POKECO,8:NM=
    NM+2:NC=NC+2:CL=NC:POKES,0:BO=0:GOTO1
    80                                  :rem 231
680 PRINTCHR$(147)CHR$(144):POKECO,30
                                        :rem 142
690 PRINT"SCORE ";SC:IFSC>HSTHENHS=SC
                                        :rem 146
700 PRINT:PRINT"+ HIGH SCORE ";HS:PRINT:P
    RINT:PRINT"PRESS THE FIRE BUTTON TO P
    LAY AGAIN"                          :rem 14
710 GOSUB390:IFFR=0THEN710              :rem 73
720 POKES,200:BO=0:SC=0:NC=1:NM=2:CL=NC:P
    RINTCHR$(147):ML=5:POKECO,8:POKES,0:G
    OTO180                              :rem 223
```

Program 3: React—64 Version

```
100 POKE53280,15:POKE53281,15       :rem 82
110 PRINT"{CLR}":POKE53280,15:POKE53281,1
    5                                   :rem 241
120 PRINT"{BLU}{10 DOWN}{2 RIGHT}LOADING
    {SPACE}CHARACTERS{6 F}PLEASE WAIT"
                                        :rem 150
130 PRINTCHR$(142):POKE52,48:POKE56,48:CL
    R                                   :rem 249
140 POKE56334,PEEK(56334)AND254:POKE1,PEE
    K(1)AND251                          :rem 182
150 FORI=0TO1024:POKEI+12288,PEEK(I+53248
    ):NEXT:POKE1,PEEK(1)OR4             :rem 86
160 POKE56334,PEEK(56334)OR1            :rem 68
170 POKE53272,(PEEK(53272)AND240)OR12
                                        :rem 45
180 FORI=12288TO12295:READA:POKEI,A:NEXT:
    DATA24,90,36,24,24,24,36,66         :rem 167
190 FORI=12504TO12671:READA:POKEI,A:NEXT
                                        :rem 214
200 DATA0,0,24,24,60,90,60,66           :rem 173
210 DATA60,66,165,129,165,154,66,60
                                        :rem 243
220 DATA255,129,165,129,189,165,189,255
                                        :rem 206
230 DATA0,126,126,126,126,126,126,0
                                        :rem 217
240 DATA170,255,85,255,170,255,85,255,0,0
    ,0,0,0,0,0,0                        :rem 62
250 DATA0,6,8,60,126,126,126,60         :rem 26
260 DATA24,60,126,255,255,255,60,60
                                        :rem 235
270 DATA56,60,254,255,255,254,60,56
                                        :rem 247
280 DATA60,60,255,255,255,126,60,24
                                        :rem 237
290 DATA28,60,127,255,255,127,60,28
                                        :rem 245
300 DATA0,0,56,40,8,20,34,127           :rem 171
310 DATA28,34,42,34,28,20,119,0         :rem 23
320 DATA219,195,36,153,153,36,195,219
                                        :rem 93
330 DATA75,75,75,75,180,180,180,180
                                        :rem 248
340 DATA126,102,66,126,24,36,195,66
                                        :rem 240
350 DATA60,66,189,255,255,189,66,60:rem 6
360 DATA0,0,56,68,146,68,56,0           :rem 196
370 DATA0,60,60,60,60,60,255,0          :rem 226
380 DATA129,195,231,255,255,231,195,129
                                        :rem 195
390 DATA0,102,102,0,66,102,60,24        :rem 59
400 FORI=12752TO12799:READA:POKEI,A:NEXT
                                        :rem 224
410 DATA171,171,173,173,181,181,213,213
                                        :rem 171
420 DATA195,195,0,24,24,0,195,195:rem 140
430 DATA255,145,145,159,249,137,137,255
                                        :rem 197
440 DATA0,0,151,146,242,146,151,0:rem 114
450 DATA165,36,231,24,24,231,36,165
                                        :rem 233
460 DATA195,231,126,60,60,126,231,195
                                        :rem 80
480 CLR:FORI=54272TO54296:POKEI,0:NEXT:PO
    KE54296,15:POKE54277,17:POKE54278,240
                                        :rem 156
490 POKE54276,33                        :rem 102
500 PRINT"{CLR}":PRINTTAB(12)"RRRRRRRRRR
    RR"                                 :rem 185
510 FORI=1TO7:PRINTTAB(12)"R{11 SPACES}R"
    :NEXT                               :rem 193
520 PRINTTAB(12)"RRRRRRRRRRRR E A C T"
                                        :rem 58
530 PRINTTAB(12)"RR"                    :rem 152
540 PRINTTAB(12)"R R"                   :rem 153
550 PRINTTAB(12)"R{2 SPACES}R"          :rem 154
560 PRINTTAB(12)"R{3 SPACES}R"          :rem 155
570 FORI=5TO7:PRINTTAB(12)"R"TAB(I+12)"R"
    :NEXT                               :rem 14
580 FORI=8TO15:POKE53280,15:POKE53281,I:F
    ORJ=1TO9                            :rem 222
590 POKE54273,J+2*I:POKE54272,I:NEXTJ:NEX
    TI:POKE54276,32                     :rem 129
600 PRINT:PRINT"{3 RIGHT}{3 DOWN}PRESS TH
    E FIRE BUTTON TO START"             :rem 89
610 A=1465:SM=28:ML=5                   :rem 171
620 C=54272:ER=32:MA=0:TR=30:CH=33:WA=31:
    PA=56321                            :rem 40
630 OP=127:TF=255:N1=128:N2=8:N3=16:N4=4:
    NE=1:SC=0:DIMJS(12):NM=2:NC=1:CL=NC
                                        :rem 249
640 JS(1)=-40:JS(2)=40:JS(4)=-1:JS(5)=-39
    :JS(6)=39:JS(8)=1:JS(9)=-41:JS(10)=41
                                        :rem 220
650 GOSUB840:IFFR=16THEN650             :rem 134
660 POKE54276,17:POKE54273,72:POKE54272,2
    00:POKE54276,16                     :rem 153
670 PRINTCHR$(147):POKE53280,0:POKE53281,
    0                                   :rem 167
680 POKE54276,17:POKE54273,81:POKE54273,1
    20:POKE54276,16:GOSUB860            :rem 245
```

```
690 A=1065:FORI=1TONM                    :rem 255
700 W=INT(RND(I)*999)+1:J=INT(RND(I)*14)+
    1:IFPEEK(1024+W)<>32THEN700          :rem 169
710 IFW+1024=ATHEN700                    :rem 180
715 POKE54276,17:POKE54273,40:POKE54272,4
    1:POKE54276,16                       :rem 104
720 POKEW+1024,28:POKEW+C+1024,J:NEXT
                                         :rem 182
730 FORI=1TONC:W=INT(RND(I)*999)+1:rem 37
740 W=INT(RND(I)*999)+1:J=INT(RND(I)*14)+
    1:K=INT(RND(I)*64)+1                 :rem 236
750 IFPEEK(1024+W)<>32THEN740            :rem 147
760 IFK>26ANDK<33ORK=0ORW+1024=ATHEN740
                                         :rem 104
770 POKE54276,17:POKE54273,34:POKE54272,7
    5:POKEW+1024,K:POKEW+C+1024,J        :rem 46
780 NEXT:POKE54276,16:TI$="000000"
                                         :rem 126
790 POKEA,MA:GOSUB850:B=A+JS(J1)         :rem 175
800 IFPEEK(B)=WAORPEEK(B)=TRORPEEK(B)=SMT
    HEN1000                              :rem 22
810 IFPEEK(B)=MATHEN830                  :rem 117
820 IFPEEK(B)<>ERTHENPOKE54276,17:POKE542
    73,40:CL=CL-NE:SC=SC+50:POKE54276,16
                                         :rem 118
825 IFCL=0THEN1040                       :rem 30
830 POKE54276,65:POKE54272,11:POKE54273,2
    :POKE54276,64:SC=SC+NE:POKEA,TR
                                         :rem 20
835 A=B:GOTO790                          :rem 115
840 P=PEEK(PA):FR=PAND16:RETURN          :rem 227
850 P=PEEK(PA):J1=15-(PAND15):RETURN
                                         :rem 170
860 TC=INT(RND(1)*14)+1:FORI=1024TO2023:P
    OKEI+C,TC:NEXT                       :rem 51
870 FORI=1024TO1063:POKEI,WA:POKEI+C,NE:P
    OKEI+960,WA:POKEI+C+960,NE:NEXT
                                         :rem 247
880 FORI=1064TO1984STEP40:POKEI,WA:POKEI+
    C,NE:POKEI+39,WA:POKEI+C+39,NE:NEXT
                                         :rem 66
890 W=INT(RND(1)*5)+1                    :rem 146
900 ONWGOSUB920,940,970                  :rem 61
910 RETURN                               :rem 122
920 FORI=1233TO1780STEP40:POKEI+C,NE:POKE
    I,WA:POKEI+C+11,NE:POKEI+11,WA:NEXT
                                         :rem 33
930 RETURN                               :rem 124
940 FORI=1044TO1284STEP40:POKEI+C,NE:POKE
    I,WA:POKEI+C+NE,NE:POKEI+NE,WA:rem 11
950 POKEI+C+330,NE:POKEI+330,WA:POKEI+C+3
    31,NE:POKEI+331,WA:NEXT              :rem 141
960 FORI=1504TO1516:POKEI+C,NE:POKEI,WA:P
    OKEI+C+14,NE:POKEI+14,WA:NEXT:RETURN
                                         :rem 163
970 FORI=1074TO1314STEP40:POKEI+C,NE:POKE
    I,WA:POKEI+C+NE,NE:POKEI+NE,WA:NEXT
                                         :rem 132
980 FORI=1440TO1920STEP40:POKEI+C,NE:POKE
    I,WA:POKEI+C+12,NE                   :rem 168
990 POKEI+12,WA:NEXT:RETURN              :rem 255
1000 IFPEEK(B)=28THENPOKEB,29            :rem 230
1010 POKEA,27:POKE54276,17:FORI=1TO10:POK
     E54273,RND(0)*200+5                 :rem 195
1020 POKE54272,RND(0)*100+10:NEXT:POKE542
     76,16:ML=ML-1:IFML=0THEN1130        :rem 21
1030 GOTO1070                            :rem 197
1040 POKEA,TR:POKEB,MA:POKE54276,17:FORI=
     1TO30:POKE54273,6*I                 :rem 146
1050 POKE54272,6*I:NEXT:POKE54276,16
                                         :rem 75
1060 BO=100-VAL(TI$):IFBO<0THENBO=0
                                         :rem 59
1070 PRINTCHR$(147)CHR$(158):SC=SC+BO
                                         :rem 146
1080 PRINTTAB(14)"{5 DOWN}SCORE ";SC
                                         :rem 201
1090 PRINTTAB(14)"{DOWN}MAYNERDS LEFT ";M
     L:PRINTTAB(14)"{DOWN}BONUS ";BO
                                         :rem 151
1100 PRINT"{7 DOWN}{5 RIGHT}PRESS FIRE BU
     TTON TO CONTINUE"                   :rem 50
1110 GOSUB840:IFFR=16THEN1110            :rem 214
1120 PRINTCHR$(147):NM=NM+2:NC=NC+2:CL=NC
     :BO=0:GOTO680                       :rem 31
1130 PRINTCHR$(147)CHR$(158)             :rem 48
1140 PRINTTAB(14)"{5 DOWN}SCORE ";SC:IFSC
     >HSTHENHS=SC                        :rem 155
1150 PRINTTAB(14)"{DOWN}+ HIGH SCORE ";HS
                                         :rem 211
1155 PRINT"{7 DOWN}{6 RIGHT}PRESS THE FIR
     E BUTTON TO PLAY"                   :rem 11
1160 GOSUB840:IFFR=16THEN1160            :rem 224
1170 BO=0:SC=0:NC=1:NM=2:CL=NC:PRINTCHR$(
     147):ML=5                           :rem 18
1180 GOTO680                             :rem 161
```

Multicolor Character Generator

(Article on page 124.)

BEFORE TYPING...

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

```
0 PRINT"{CLR}{RVS} MULTICOLOR CHARACTER";
  SPC(7);"GENERATOR"                     :rem 12
1 POKE36869,255:POKE52,28:POKE56,28:CLR:F
  ORA=7168TO7679:POKEA,PEEK(25600+A):NEXT
                                         :rem 70
2 S=7680:C=38400:M=7168:I=36879:DIMC$(15)
  :DEFFND(Q)=X+22*Y:FORA=828TO898:POKEA,0
  :NEXT                                  :rem 35
3 FORA=0TO15:READA$:C$(A)=A$:NEXT:PRINT"
  {BLK}";                                :rem 176
4 PRINT"{CLR}";:FORA=0TO7:FORB=0TO7
                                         :rem 38
5 POKEC+A+22*B,PEEK(828+A+8*B):POKES+A+22
  *B,A+8*B:NEXTB,A:GOTO9                 :rem 192
6 FORA=0TO7:PRINTTAB(8);"{RVS}-":NEXT:PRI
  NT"{RVS}********{X}":PRINT"
  {HOME}";:RETURN                        :rem 0
7 DATABLACK,WHITE,RED,CYAN,PURPLE,GREEN,B
  LUE,YELLOW,ORANGE,LT.OR.,PINK          :rem 197
8 DATALT.CYAN,LT.PUR.,LT.GRN.,LT.BLUE,LT.
  YEL.                                   :rem 173
9 GOSUB6:PRINT:PRINTTAB(11);"{RVS}COMMAND
  S":PRINT                               :rem 177
10 PRINTTAB(10)"{RVS}S-SCREEN":PRINTTAB(1
   0)"{RVS}B-BORDER":PRINTTAB(10)"{RVS}A-
   AUXILIARY"                            :rem 172
11 PRINTTAB(10);"{RVS}N-NEW CHAR":PRINT:P
   RINT:PRINT                            :rem 225
12 X=0:Y=0:POKES,230:POKEC,Q             :rem 206
```

```
13 FORA=ØTO7:PRINT"{RVS}";MID$(STR$(A),2)
   ;"-";C$(A),MID$(STR$(A+8),2); :rem 141
14 PRINT"-";C$(A+8):NEXT:PRINT:PRINT:GOTO
   2Ø                              :rem 214
15 SR=(PEEK(36879)AND24Ø)/16:PRINT"{RVS}S
   CREEN ";C$(SR)                  :rem 148
16 BR=PEEK(36879)AND7:PRINT"{RVS}BORDER "
   ;C$(BR)                         :rem 43
17 AX=(PEEK(36878)AND24Ø)/16:PRINT"{RVS}A
   UXIL. ";C$(AX):RETURN           :rem 136
18 FORA=ØTO63:IFPEEK(828+A)<2THENPOKE828+
   A,Q                             :rem 174
19 NEXT:RETURN                     :rem 195
2Ø GETA$:IFA$=""THEN2Ø             :rem 231
21 SC$="":IFA$="N"THEN5Ø           :rem 47
22 IFA$="S"THENINPUT"{RVS}SCREEN (Ø-15) "
   ;SC$:GOSUB26:POKEI,(PEEK(I)AND7)OR16*S
   C+8:GOSUB29                     :rem 116
23 IFA$="B"THENINPUT"{RVS}BORDER (Ø-7) ";
   SC$:GOSUB26:POKEI,(PEEK(I)AND248)+SC:G
   OSUB29                          :rem 48
24 IFA$="A"THENINPUT"{RVS}AUX. (Ø-15) ";S
   C$:GOSUB26:POKEI-1,16*SC:GOSUB29:rem Ø
25 GOSUB45:GOTO2Ø                  :rem 37
26 SC=VAL(SC$):IFSC>15THENSC=1     :rem 95
27 IFA$="B"ANDSC>7THENSC=Ø         :rem 48
28 RETURN                          :rem 74
29 PRINT"{RVS}{HOME}{2Ø DOWN}{22 SPACES}
   {UP}";                          :rem 129
3Ø IFA$<>"S"THENRETURN             :rem 119
31 IFSC=ØTHENPRINT"{WHT}":Q=1:GOSUB18:GOT
   O4                              :rem 19Ø
32 IFQ=1THENIFSC<>ØTHENQ=Ø:GOSUB18:PRINT"
   {BLK}":GOTO4                    :rem 3
33 RETURN                          :rem 7Ø
34 V=FND(Q):IFFL=1THENPOKES+V,IL:POKES+V+
   1,IL:POKEC+V,CL:POKEC+V+1,CL:RETURN
                                   :rem 2Ø8
35 POKES+V,X+Y*8:POKEC+V,PEEK(828+X+Y*8):
   RETURN                          :rem 2Ø1
36 IFFL=1ANDA$="{RIGHT}"THENX=X+1  :rem 7Ø
37 IFFL=1ANDA$="{LEFT}"THENX=X-1   :rem 2Ø1
38 IFX>7THENX=.                    :rem 185
39 IFY<ØTHENY=7                    :rem 188
4Ø IFX<ØTHENX=7:IFFL=1THENX=6      :rem 117
41 IFY>7THENY=.                    :rem 181
42 V=FND(Q):IL=PEEK(S+V):CL=PEEK(C+V):POK
   ES+V,23Ø:POKEC+V,Q              :rem 177
43 IFFL=1THENPOKES+V+1,23Ø:POKEC+V+1,Q
                                   :rem 75
44 RETURN                          :rem 72
45 IFA$="{RIGHT}"THENGOSUB34:X=X+1:GOSUB3
   6                               :rem 183
46 IFA$="{DOWN}"THENGOSUB34:Y=Y+1:GOSUB36
                                   :rem 174
47 IFA$="{LEFT}"THENGOSUB34:X=X-1:GOSUB36
                                   :rem 59
48 IFA$="{UP}"THENGOSUB34:Y=Y-1:GOSUB36
                                   :rem 5Ø
49 RETURN                          :rem 77
5Ø CH=X+8*Y:FL=1:FORA=M+8*(CH)TOM+8*(CH)+
   7:POKEA,Ø:NEXT:IL=174:CL=Q      :rem 51
51 FORA=M+8*CHTOM+8*CH+7:POKEA,Ø:NEXT
                                   :rem 25
52 PRINT"{CLR}{3 DOWN}";TAB(9);"{RVS}C-CH
   AR.{2 SPACES}DOT":PRINTTAB(9);"{RVS}A-
   AUXIL. DOT"                     :rem 141
53 PRINTTAB(9);"{RVS}B-BORDER DOT"
                                   :rem 251
54 PRINTTAB(9);"{RVS}S-SCREEN DOT":PRINTT
   AB(9);"{RVS}R-RETURN{4 SPACES}":PRINT"
   {HOME}";                        :rem 128
55 FORA=ØTO7:PRINT"{RVS}........":NEXT:X=
   Ø:Y=Ø:POKES,23Ø:POKES+1,23Ø     :rem 52
56 POKE7722,CH:PRINT:PRINT:GOSUB15:PRINT"
   {HOME}";:GOSUB6:FORA=ØTO4:POKE8185+A,2
   52:NEXT                         :rem 169
57 PRINT"{HOME}";TAB(19);"{RVS}[A]*
   [S]{DOWN}{3 LEFT}-{RIGHT}-{DOWN}
   {3 LEFT}[Z]*[X]";:POKE8184,48:POK
   E8190,48:POKE8191,Ø             :rem 25
58 PRINT"{HOME}";TAB(1Ø);"{RVS}{DOWN}COMM
   ANDS":POKE38904,SR:POKE38905,SR
                                   :rem 142
59 PRINT"{HOME}{15 DOWN}";:FORA=ØTO3:PRIN
   T"{RVS}";A;C$(A),A+4;C$(A+4):NEXT
                                   :rem 214
6Ø PRINT:INPUT"{RVS}CHARACTER (Ø-7) ";SC$
   :A$="B":GOSUB26:SC=SC+8:POKE38442,SC:C
   R=SC:GOSUB29                    :rem 251
61 POKE828+CH,SC:PRINT"{7 UP}";:PRINT"
   {RVS}CHAR.{2 SPACES}";C$(SC-8) :rem 68
62 GETA$:IFA$=""THEN62             :rem 243
63 GOSUB45:PS=M+8*CH+Y:IFA$="R"THENFLAG=Ø
   :GOTO71                         :rem 231
64 IFA$="A"THENPOKEPS,PEEK(PS)OR2↑(7-X):P
   OKEPS,PEEK(PS)OR2↑(6-X)::CU=AX:GOSUB69
                                   :rem 61
65 IFA$="C"THENPOKEPS,PEEK(PS)OR2↑(7-X):P
   OKEPS,PEEK(PS)AND(255-2↑(6-X)):CU=CR-8
   :GOSUB69                        :rem 179
66 IFA$="B"THENPOKEPS,PEEK(PS)OR2↑(6-X):P
   OKEPS,PEEK(PS)AND(255-2↑(7-X)):CU=BR:G
   OSUB69                          :rem 77
67 IFA$="S"THENIL=174:POKEPS,PEEK(PS)AND(
   255-(2↑(6-X)+2↑(7-X))):CL=Q:GOSUB34:GO
   SUB7Ø                           :rem 94
68 GOTO62                          :rem 15
69 V=FND(A):POKES+V,127:POKEC+V,CU:POKES+
   V+1,127:POKEC+V+1,CU            :rem 12
7Ø X=X+2:GOSUB36:RETURN            :rem 238
71 PRINT"{CLR}{RVS}";M+8*CH;"TO";M+8*CH+7
   :PRINT:FORA=ØTO7                :rem 18
72 PRINT"{RVS}";PEEK(M+CH*8+A):NEXT:PRINT
   :PRINT"{RVS}";I;PEEK(I)         :rem 169
73 PRINT"{RVS}";I-1;PEEK(I-1):PRINT"{RVS}
   ";C;CR:PRINT"{RVS}";S;CH        :rem 99
74 PRINT:PRINT:PRINT"{RVS}HIT A KEY":POKE
   198,Ø:WAIT198,1:GOTO4           :rem 1Ø3
```

Haunted Mansion

(Article on page 62.)

Program 1:

Haunted Mansion—VIC Version

```
1ØØ IFPEEK(44)<32THENPOKE56,28:POKE52,28
                                   :rem 1ØØ
1Ø5 POKE36879,8:PRINT"{CLR}{GRN}{3 DOWN}*
   ***{RVS}HAUNTED{2 SPACES}HOUSE{OFF}**
   **";                            :rem 229
11Ø FORI=7168TO7679:POKEI,PEEK(25600+I):N
   EXT                             :rem 147
12Ø FORI=832TO936:READA:POKEI,A:NEXT
                                   :rem 17
13Ø FORI=7168+35*8TO7168+45*8+7:READA:POK
   EI,A:NEXT                       :rem 2Ø4
14Ø FORI=7168+27*8TO7168+31*8+7:READA:POK
   EI,A:NEXT                       :rem 2Ø1
15Ø FORI=7168+58*8TO7168+62*8+7:READA:POK
   EI,A:NEXT                       :rem 21Ø
```

```
152 SC=4*(PEEK(36866)AND128)+64*(PEEK(368
    69)AND112):REM FIND SCREEN RAM
                                    :rem 231
153 CM=37888+4*(PEEK(36866)AND128)-SC:REM
     FIND COLOR RAM MINUS SCREEN RAM
                                    :rem 208
155 SH=36876:SL=36874:V=36878:O=0:P=1:Q=2
    2:DIMA(13)                      :rem 164
160 GOSUB800                        :rem 175
165 GOSUB900                        :rem 181
170 POKE36869,PEEK(36869)AND240OR15
                                    :rem 249
175 GOSUB1000:RN=RN+1               :rem 241
200 CL=SC+429                       :rem 190
210 SYS832                           :rem 47
215 TL=CL:Z=TL:ONPEEK(830)GOSUB301,300,30
    3,300,305,300,307,300           :rem 185
220 CL=Z:POKESH,220:POKEV,2:GOSUB400
                                    :rem 233
225 POKESH,0:POKESL,0:POKEV,0:IFDFTHEN500
                                    :rem 178
230 POKETL,32:POKECL,58:POKECM+CL,3+CF
                                     :rem 76
232 GOSUB700:IFCC=1THENFORX=1TO1500:NEXT:
    GOTO170                         :rem 52
235 GOSUB600:IFDFTHEN500           :rem 199
240 GOTO210                         :rem 98
300 RETURN                         :rem 115
301 Z=Z-Q:RETURN                    :rem 29
303 Z=Z+P:RETURN                    :rem 28
305 Z=Z+Q:RETURN                    :rem 31
307 Z=Z-P:RETURN                    :rem 34
400 REM COLLISION CHECK            :rem 130
405 IFPEEK(CL)=44ORPEEK(CL)=45ORPEEK(CL)=
    35THENCL=TL:RETURN              :rem 30
407 IFCFANDCL<SC+439ANDCL>SC+419THEN2000
                                    :rem 162
410 IFCFANDPEEK(CL)=60THENCL=TL:RETURN
                                    :rem 221
415 IFPEEK(CL)=60THENCF=4:POKEV,10:FORI=1
    TO50:NEXT:RETURN               :rem 185
420 IFPEEK(CL)=61ORPEEK(CL)=59THEN1800
                                     :rem 23
425 IFPEEK(CL)=31THEN1900          :rem 201
430 RETURN                         :rem 119
500 REM GOTCHA!                     :rem 80
510 PRINT"{CLR}{BLU}{DOWN}{2 RIGHT}ANOTHE
    R VICTIM!"                      :rem 98
520 POKE36869,PEEK(36869)AND240OR8:POKE36
    879,27                         :rem 209
525 PRINT"{DOWN}{RIGHT}SKILL LEVEL"AA
                                    :rem 20
530 PRINT"{DOWN}{RIGHT}ROUND"RN"SCORE"SR
                                    :rem 36
540 PRINT"{DOWN}{RIGHT}PLAY AGAIN?
    {2 SPACES}{RVS}Y{OFF} OR {RVS}N{OFF}"
                                    :rem 253
550 GETA$:IFA$=""THEN550            :rem 87
560 IFA$="Y"THENCF=0:GOTO570       :rem 157
562 IFA$<>"N"THEN550               :rem 102
565 END                            :rem 119
570 RN=0:SR=0:DF=0                  :rem 38
575 GOTO165                        :rem 118
600 REM MOVE SPIRITS               :rem 223
610 I=INT(RND(1)*(AA*2))+1         :rem 116
620 TL=A(I):Z=TL:POKEA(I),32       :rem 150
630 ONINT(RND(1)*4)+1GOSUB301,303,305,307
                                    :rem 242
635 IFZ>SC+419ANDZ<SC+439THEN660   :rem 185
640 IFPEEK(Z)=58THEN1900           :rem 156
650 IFPEEK(Z)=32THENA(I)=Z          :rem 61
660 POKEA(I),31:POKECM+A(I),4:RETURN
                                    :rem 175
700 PRINT"{HOME}{21 DOWN}{RIGHT}{WHT}ROUN
    D"RN"SCORE"SR"{LEFT} ";:RETURN
                                    :rem 197
800 PRINT"{CLR}YOU ARE ENTERING A
    {4 SPACES}WITCH'S HAUNTED HOUSE.";
                                     :rem 93
815 PRINT"THE WITCH IS AWAY,{4 SPACES}FLY
    ING ON HER BROOM.{2 SPACES}";:rem 246
820 PRINT"SHE HAS CAPTURED{6 SPACES}YELLO
    W CATS AND WILL{2 SPACES}TURN THEM IN
    TO WITCH{2 SPACES}";           :rem 252
825 PRINT"CATS UNLESS YOU RESCUETHEM."
                                    :rem 149
830 PRINT"GUIDE YOURSELF WITH A JOYSTICK.
     PICK UP ONE CAT AT A TIME. BRING
    {2 SPACES}";                    :rem 212
835 PRINT"IT TO THE BOTTOM ROW. WHILE CAR
    RYING A CAT, YOU WILL TURN YELLOW. ";
                                    :rem 109
840 PRINT"YOU CAN PICK UP ONLY{2 SPACES}O
    NE CAT AT A TIME, ANDYOU GET POINTS F
    OR{4 SPACES}";                  :rem 228
845 PRINT"EACH CAT YOU SAVE.{4 SPACES}WHE
    N YOU SAVE 10 CATS,YOU GET A NEW HOUS
    E.{2 SPACES}";                  :rem 192
847 PRINT"{4 SPACES}PRESS ANY KEY
    {5 SPACES}";                     :rem 13
850 GETA$:IFA$=""THEN850            :rem 93
855 PRINT"G{P}F YOU RUN INTO A BAT OR GHO
    ST, YOU'LL JUMP IN FEAR, DROP ANY
    {5 SPACES}";                    :rem 215
860 PRINT"CAT YOU ARE CARRYING{2 SPACES}A
    ND{2 SPACES}LOSE POINTS."       :rem 140
865 PRINT"THE HOUSE IS HAUNTED{2 SPACES}B
    Y EVIL SPIRITS{7 SPACES}FLOATING ALON
    G THE{4 SPACES}HALLWAYS."       :rem 176
866 PRINT"IF YOU TOUCH A SPIRIT,YOU LOSE
    {SPACE}THE GAME!"                :rem 24
867 PRINT" (BOTTOM ROW IS SAFE) ";
                                    :rem 138
870 PRINT"{RED}HIGHER SKILL LEVELS
    {3 SPACES}SCORE MORE POINTS BUT ARE H
    ARDER.{11 SPACES}{OFF}";        :rem 141
890 RETURN                          :rem 129
900 PRINT"ENTER SKILL LEVEL 1-6"    :rem 115
920 GETA$:IFA$=""THEN920             :rem 89
930 AA=VAL(A$):IFAA<1ORAA>6THEN920:rem 13
940 RETURN                          :rem 125
1000 POKE36879,8:PRINT"{CLR}"        :rem 3
1003 PRINT">{YEL}>.>.>>{BLU}-,{YEL}>.>>>>
     >>>>.>";                       :rem 28
1005 PRINT">>>>>>{BLU}-##,{YEL}>.>>$%>.>>
     >>";                           :rem 55
1010 PRINT">>>.>{BLU}-####,{YEL}>>&[£'>>
     .>>";                          :rem 60
1015 PRINT">>.>{BLU}-######,{YEL}>(]↑)>>>
     >>";                           :rem 35
1020 PRINT">>>{BLU}-########,{YEL}>*+>.>>
     >>";                           :rem 174
1025 PRINT">>{BLU}-##########,{YEL}>>>>>>
     >>";                           :rem 180
1030 PRINT">{BLU}-############,{YEL}>>>.>
     >>";                           :rem 106
1035 PRINT">{BLU}##############,{YEL}>.>>
     >>";                            :rem 74
1040 PRINT">{BLU}###############,{YEL}>>>
     .>";                           :rem 43
1045 PRINT">{BLU}################,{WHT}>>
     >>";                           :rem 140
```

```
1050 PRINT">{BLU}#################,{WHT}>
     >>";                              :rem 109
1055 PRINT">{BLU}##################,{WHT}
     >>";                               :rem 87
1060 PRINT">{BLU}###################,
     {WHT}>";                           :rem 56
1065 PRINT">{BLU}####################,
     {WHT}";                            :rem 34
1075 PRINT">{BLU}#####################
     {WHT}";                            :rem 26
1080 PRINT">{BLU}#####################
     {WHT}";                            :rem 22
1085 PRINT">{BLU}#####################
     {WHT}";                            :rem 27
1090 PRINT">{BLU}#####################
     {WHT}";                            :rem 23
1092 PRINT">{BLU}#####################
     {WHT}";                            :rem 25
1093 PRINT">{BLU}#####################
     {WHT}";                            :rem 26
1200 REM BUILD MAZE                     :rem 68
1205 A(Ø)=2:A(1)=-44:A(2)=-2:A(3)=44:WL=3
     5:HL=32:A=SC+42Ø                    :rem 7
121Ø POKEA,4                           :rem 148
122Ø J=INT(RND(1)*4):X=J               :rem 101
123Ø B=A+A(J):IFPEEK(B)=WLTHENPOKEB,J:POK
     EA+A(J)/2,HL:A=B:GOTO122Ø           :rem 8
124Ø J=(J+1)*-(J<3):IFJ<>XTHEN123Ø
                                       :rem 128
125Ø J=PEEK(A):POKEA,HL:IFJ<4THENA=A-A(J)
     :GOTO122Ø                          :rem 34
126Ø FORI=SC+114TOSC+422STEP22:POKEI,32:P
     OKEI+7,32:NEXT                    :rem 138
127Ø FORI=SC+74TOSC+426STEP22:POKEI,32:NE
     XT                                :rem 191
128Ø FORI=SC+282TOSC+436STEP22:POKEI,32:N
     EXT                               :rem 242
129Ø FORI=SC+213TOSC+433STEP22:POKEI,32:N
     EXT                               :rem 234
1291 FORI=SC+2ØØTOSC+212:POKEI,32:NEXT
                                        :rem 66
1292 FORI=SC+332TOSC+35Ø:POKEI,32:POKEI+8
     8,32:NEXT                          :rem 42
13ØØ REM PLACE GAME CHARACTERS           :rem 7
131Ø REMPLACE GHOSTS                   :rem 23Ø
132Ø FORI=PTO3*AA                        :rem 5
133Ø X=INT(RND(1)*374)+SC+22            :rem 22
134Ø BL=Ø:GOSUB17ØØ:IFBLTHEN133Ø       :rem 146
135Ø POKEX,59:POKECM+X,1:NEXT           :rem 6Ø
14ØØ REMPLACE BATS                      :rem 56
142Ø FORI=PTO3*AA                        :rem 6
143Ø X=INT(RND(1)*374)+SC+22            :rem 23
144Ø BL=Ø:GOSUB17ØØ:IFBLTHEN143Ø       :rem 148
145Ø POKEX,61:POKECM+X,5:NEXT           :rem 58
15ØØ REMPLACE CATS                      :rem 58
152Ø CC=11:FORI=PTO1Ø                  :rem 232
153Ø X=INT(RND(1)*374)+SC+22            :rem 24
154Ø BL=Ø:GOSUB17ØØ:IFBLTHEN153Ø       :rem 15Ø
155Ø POKEX,6Ø:POKECM+X,7:NEXT           :rem 6Ø

16ØØ REMPLACE SPIRITS                   :rem 62
162Ø FORI=PTOAA*2                        :rem 7
163Ø X=INT(RND(1)*374)+SC+22            :rem 25
164Ø IFPEEK(X)<>32THEN163Ø               :rem Ø
165Ø POKEX,31:POKECM+X,4:A(I)=X:NEXT:RETU
     RN                                :rem 252
17ØØ IF(PEEK(X)<>32)OR(PEEK(X+P)<>32ANDPE
     EK(X+P)<>35)THENBL=1              :rem 152
171Ø IF(PEEK(X-P)<>32ANDPEEK(X-P)<>35)OR(
     PEEK(X+Q)<>32ANDPEEK(X+Q)<>35)THENBL
     =1                                 :rem 24
172Ø IF(PEEK(X-Q)<>32ANDPEEK(X-Q)<>35)THE
     NBL=1                               :rem 1
173Ø RETURN                            :rem 171
18ØØ REM SCARED!                       :rem 128
181Ø IFCF=OTHEN184Ø                    :rem 1Ø6
182Ø X=INT(RND(1)*374)+SC+22            :rem 26
183Ø IFPEEK(X)<>32THEN182Ø               :rem 2
1835 POKEX,6Ø:POKECM+X,7:CF=O:SR=SR-2*AA↑
     2:IFSR<OTHENSR=O                  :rem 2Ø5
184Ø POKETL,32:POKECL,58:POKECL+CM,1:POKE
     SL,O:POKESH,18Ø:POKEV,9            :rem 62
1845 FORI=1TO4ØØ:NEXT                   :rem 32
185Ø TL=CL:Z=TL:ONINT(RND(1)*4)+1GOSUB3Ø1
     ,3Ø3,3Ø5,3Ø7                       :rem 62
186Ø CL=Z:SR=SR-AA↑2:IFSR<OTHENSR=O
                                        :rem 78
187Ø GOTO4ØØ                           :rem 157
19ØØ REM GOTCHA! SOUND                  :rem 14
19Ø5 POKETL,32:POKECL,58:POKECM+CL,4:DF=1
                                         :rem 5
191Ø POKEV,5:POKESH,Ø:FORI=1TO4:POKESL,25
     5:FORX=1TO15Ø:NEXT:POKESL,18Ø:FORX=1
     TO75:NEXT                         :rem 1Ø1
1913 IFI=1ORI=3THENPOKECL,31:GOTO192Ø
                                        :rem 42
1914 POKECL,58                          :rem 38
192Ø POKESL,Ø:FORX=1TO2ØØ:NEXT:NEXT:FORX=
     1TO5ØØ:NEXT:RETURN                :rem 182
2ØØØ REM LINE UP SAVED CAT             :rem 19Ø
2Ø1Ø X=SC+5ØØ:POKEX-CC,6Ø:POKECM+X-CC,7:C
     C=CC-1:SR=SR+1Ø*AA↑2:CF=O         :rem 255
2Ø2Ø POKEV,1Ø:FORI=1TO5Ø:NEXT:RETURN
                                        :rem 73

9ØØØ DATA12Ø,8,72,152,72,138,72,173,19,14
     5,72,173,34,145,72,169,Ø,141,62,3,14
     1,63,3,169                        :rem 241
9Ø1Ø DATA127,141,34,145,173,32,145,73,255
     ,41,128,42,8,169,195,141,19,145,173,
     17,145,73                         :rem 2Ø7
9Ø2Ø DATA255,41,6Ø,74,74,4Ø,42,168,41,16,
     2Ø1,16,2Ø8,3,141,63,3,152,41,15,162,
     Ø,232,224                         :rem 159
9Ø3Ø DATA9,24Ø,8,221,16Ø,3,2Ø8,246,142,62
     ,3,1Ø4,141,34,145,1Ø4,141,19,145,1Ø4
     ,17Ø,1Ø4                          :rem 1Ø9
9Ø4Ø DATA168,1Ø4,4Ø,88,96,2,3,1,5,4,12,8,
     1Ø                                :rem 1Ø5
1ØØØØ DATA255,255,255,255,255,255,255,255
                                        :rem 31
1ØØ1Ø DATAØ,Ø,Ø,Ø,Ø,15,63,255          :rem 155
1ØØ15 DATAØ,Ø,Ø,Ø,Ø,24Ø,252,255          :rem Ø
1ØØ2Ø DATA1,1,3,3,7,7,7,7              :rem 229
1ØØ3Ø DATA128,128,192,192,224,224,224,224
                                        :rem 16
1ØØ4Ø DATA7,7,7,7,3,3,1,1              :rem 231
1ØØ5Ø DATA224,224,224,224,192,192,128,128
                                        :rem 18
1ØØ6Ø DATA255,63,15,Ø,Ø,Ø,Ø,Ø          :rem 16Ø
1ØØ7Ø DATA255,252,24Ø,Ø,Ø,Ø,Ø,Ø          :rem 1
1ØØ8Ø DATA128,192,224,24Ø,248,252,254,255
                                        :rem 26
1ØØ9Ø DATA1,3,7,15,31,63,127,255        :rem 76
1Ø1ØØ DATA255,255,255,255,25Ø,246,244,224
                                        :rem 21
1Ø11Ø DATA255,255,191,63,15,15,7,63
                                       :rem 233
1Ø12Ø DATA24Ø,249,24Ø,228,Ø,252,255,255
                                       :rem 17Ø
1Ø13Ø DATA255,255,127,35,1,112,63,255
                                        :rem 71
```

```
10140 DATA0,34,0,8,0,0,28,0          :rem 61
10150 DATA56,84,56,16,124,186,40,108
                                     :rem 33
10160 DATA62,42,62,28,28,28,60,120
                                     :rem 179
10170 DATA40,124,85,125,57,57,61,127
                                     :rem 32
10180 DATA0,16,124,254,214,130,0,0
                                     :rem 156
10190 DATA0,0,0,0,0,0,0,0            :rem 201
```

Program 2:

Haunted Mansion—64 Version

```
100 POKE52,48:POKE56,48:CLR          :rem 70
102 POKE53280,0:POKE53281,0          :rem 232
105 PRINT"{CLR}{57}{3 DOWN}*************
    {RVS}HAUNTED{2 SPACES}HOUSE{OFF}*****
    ********";                       :rem 119
107 PRINT"{13 DOWN}{9 SPACES}REDEFINING
    {2 SPACES}CHARACTERS"            :rem 1
108 POKE56334,PEEK(56334)AND254:POKE1,PEE
    K(1)AND251                       :rem 186
110 FORI=0TO511:POKE12288+I,PEEK(53248+I)
    :NEXT                            :rem 224
115 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334
    )OR1                             :rem 134
120 FORI=832TO936:READA:POKEI,A:NEXT
                                     :rem 17
130 FORI=12288+35*8TO12288+45*8+7:READA:P
    OKEI,A:NEXT                      :rem 42
140 FORI=12288+27*8TO12288+31*8+7:READA:P
    OKEI,A:NEXT                      :rem 39
150 FORI=12288+58*8TO12288+62*8+7:READA:P
    OKEI,A:NEXT                      :rem 48
152 SC=1024                          :rem 50
153 CM=54272                         :rem 106
155 SH=54273:SL=54272:V=54296:WF=54276:O=
    0:P=1:Q=40:DIMA(13):POKEV,15     :rem 250
157 POKESL+5,17:POKESL+6,241         :rem 139
160 GOSUB800                         :rem 175
165 GOSUB900                         :rem 181
170 POKE53272,(PEEK(53272)AND240)+12
                                     :rem 183
175 GOSUB1000:RN=RN+1                :rem 241
200 CL=SC+859                        :rem 197
210 JP=15-PEEK(56320)AND15:IFJP=8THENJP=3
    :GOTO215                         :rem 200
211 IFJP=2THENJP=5:GOTO215           :rem 114
212 IFJP=4THENJP=7:GOTO215           :rem 119
213 IFJP=1THEN215                    :rem 244
214 JP=2                             :rem 160
215 TL=CL:Z=TL:ONJPGOSUB301,300,303,300,3
    05,300,307,300                   :rem 66
220 CL=Z:POKEV,15:POKESH,50:POKEWF,17:GOS
    UB400:POKEWF,16                  :rem 33
225 IFDFTHEN500                      :rem 118
230 POKETL,32:POKECL,58:POKECM+CL,3+CF
                                     :rem 76
232 GOSUB700:IFCC=1THENFORX=1TO1500:NEXT:
    GOTO170                          :rem 52
235 GOSUB600:IFDFTHEN500             :rem 199
240 GOTO210                          :rem 98
300 RETURN                           :rem 115
301 Z=Z-Q:RETURN                     :rem 29
303 Z=Z+P:RETURN                     :rem 28
305 Z=Z+Q:RETURN                     :rem 31
307 Z=Z-P:RETURN                     :rem 34
400 REM COLLISION CHECK              :rem 130
405 IFPEEK(CL)=44ORPEEK(CL)=45ORPEEK(CL)=
    35THENCL=TL:RETURN               :rem 30
407 IFCFANDCL<SC+873ANDCL>SC+845THEN2000
                                     :rem 167
410 IFCFANDPEEK(CL)=60THENCL=TL:RETURN
                                     :rem 221
415 IFPEEK(CL)=60THENCF=4:POKEWF,33:FORI=
    1TO100:NEXT:RETURN               :rem 49
420 IFPEEK(CL)=61ORPEEK(CL)=59THEN1800
                                     :rem 23
425 IFPEEK(CL)=31THEN1900            :rem 201
430 RETURN                           :rem 119
500 REM GOTCHA!                      :rem 80
510 PRINT"{CLR}{57}{DOWN}{12 RIGHT}ANOTH
    ER VICTIM!"                      :rem 255
520 POKE53272,21                     :rem 88
525 PRINT"{DOWN}{13 RIGHT}SKILL LEVEL"AA
                                     :rem 112
530 PRINT"{DOWN}{11 RIGHT}ROUND"RN"SCORE"
    SR                               :rem 70
540 PRINT"{DOWN}{10 RIGHT}PLAY AGAIN?
    {2 SPACES}{RVS}Y{OFF} OR {RVS}N{OFF}"
                                     :rem 2
550 GETA$:IFA$=""THEN550             :rem 87
560 IFA$="Y"THENCF=0:GOTO570         :rem 157
562 IFA$<>"N"THEN550                 :rem 102
565 SYS2048                          :rem 109
570 RN=0:SR=0:DF=0                   :rem 38
575 GOTO165                          :rem 118
600 REM MOVE SPIRITS                 :rem 223
610 I=INT(RND(1)*(AA*2))+1           :rem 116
620 TL=A(I):Z=TL:POKEA(I),32         :rem 150
630 ONINT(RND(1)*4)+1GOSUB301,303,305,307
                                     :rem 242
635 IFZ>SC+845ANDZ<SC+873THEN660     :rem 190
640 IFPEEK(Z)=58THEN1900             :rem 156
650 IFPEEK(Z)=32THENA(I)=Z           :rem 61
660 POKEA(I),31:POKECM+A(I),4:RETURN
                                     :rem 175
700 PRINT"{HOME}{23 DOWN}{10 RIGHT}{WHT}R
    OUND"RN"SCORE"SR"{LEFT} ";:RETURN
                                     :rem 236
800 PRINT"{CLR}YOU WILL ENTER A WITCH'S H
    AUNTED HOUSE. ";                 :rem 223
815 PRINT"{DOWN}THE WITCH IS AWAY, FLYING
     ON HER BROOM. ";                :rem 7
820 PRINT"{DOWN}SHE HAS CAPTURED YELLOW C
    ATS AND WILL{3 SPACES}";         :rem 221
825 PRINT"{DOWN}TURN THEM INTO WITCH CATS
     UNLESS YOU{4 SPACES}{DOWN}RESCUE THE
    M."                              :rem 231
830 PRINT"{DOWN}GUIDE YOURSELF WITH A JOY
    STICK. PICK UP ";                :rem 134
832 PRINT"{DOWN}ONE CAT AT A TIME. BRING
    {SPACE}IT TO THE{6 SPACES}";     :rem 58
835 PRINT"{DOWN}BOTTOM ROW.{2 SPACES}WHIL
    E CARRYING A CAT, YOU{2 SPACES}";
                                     :rem 210
837 PRINT"{DOWN}WILL TURN YELLOW.
    {2 SPACES}YOU CAN PICK UP ONLY ";
                                     :rem 39
840 PRINT"{DOWN}ONE CAT AT A TIME, AND YO
    U GET POINTS{3 SPACES}";         :rem 49
845 PRINT"{DOWN}FOR EACH CAT YOU SAVE. WH
    EN YOU SAVE 10 {DOWN}CATS, YOU GET A
    {SPACE}NEW HOUSE."               :rem 142
847 PRINT: PRINT"{13 SPACES}PRESS ANY KEY
    ";                               :rem 212
850 GETA$:IFA$=""THEN850             :rem 93
855 PRINT"{CLR}IF YOU RUN INTO A BAT OR G
    HOST, YOU'LL{2 SPACES}";         :rem 34
```

```
860 PRINT"{DOWN}JUMP IN FEAR, DROP ANY CA
    T YOU ARE{6 SPACES}{DOWN}CARRYING AND
     LOSE POINTS."                :rem 232
865 PRINT"{DOWN}THE HOUSE IS ALSO HAUNTED
     BY EVIL{7 SPACES}";          :rem 196
866 PRINT"{DOWN}SPIRITS FLOATING ALONG TH
    E HALLWAYS."                  :rem 237
867 PRINT"{DOWN}IF YOU TOUCH A SPIRIT, YO
    U LOSE THE{5 SPACES}{DOWN}GAME!"
                                   :rem 59
868 PRINT"{DOWN}(BOTTOM ROW IS SAFE) "
                                   :rem 97
870 PRINT"{YEL}{DOWN}HIGHER SKILL LEVELS
    {SPACE}SCORE MORE POINTS{3 SPACES}
    {DOWN}BUT ARE HARDER.{OFF}"   :rem 246
890 RETURN                        :rem 129
900 PRINT"{6}{DOWN}ENTER SKILL LEVEL 1-
    6"                             :rem 29
920 GETA$:IFA$=""THEN920           :rem 89
930 AA=VAL(A$):IFAA<1ORAA>6THEN920:rem 13
940 RETURN                        :rem 125
1000 POKE53280,0:POKE53281,0:PRINT"{CLR}"
                                  :rem 180
1003 PRINT">{YEL}>.>.>>>>>>>>>>>>>{BLU}-
     ,{YEL}>.>>>>>>>>>.>>>>>>>"; :rem 120
1005 PRINT">>>>>>>.>>.>>>>>>>{BLU}-##,
     {YEL}>.>>$%>.>>>>.>>>>>";     :rem 99
1010 PRINT">>>>.>>>>>>>>>>.>{BLU}-####,
     {YEL}>>&[£'>>.>>>>>>>>";    :rem 136
1015 PRINT">>>>>>>.>>>>>>>>{BLU}-######,
     {YEL}>(]↑)>.>>>.>>>>>";      :rem 95
1020 PRINT">>>>>>>>>>.>>>>{BLU}-########,
     {YEL}>*+>.>>>>>>>>>>";      :rem 250
1025 PRINT">>>>>.>>>>>>>>{BLU}-##########
     ,{YEL}>>>.>>>>>>>.>>";      :rem 224
1030 PRINT">>>>>>>>>>>>>{BLU}-###########
     #,{YEL}>>>.>>>>>>>>>";      :rem 198
1035 PRINT">>>.>>>>>.>>{BLU}-############
     ##,{YEL}>.>>>>>>.>>>";      :rem 101
1040 PRINT">>.>>>>>>>>{BLU}-#############
     ###,{YEL}>>>.>>>>>>>";       :rem 75
1045 PRINT">>>>>>>>>>{BLU}-##############
     ####,{WHT}>>>>>>>>>>";      :rem 161
1050 PRINT">>>>>>>>>{BLU}-###############
     #####,{WHT}>>>>>>>>>";      :rem 103
1055 PRINT">>>>>>>>{BLU}-################
     ######,{WHT}>>>>>>>>";       :rem 54
1060 PRINT">>>>>>>{BLU}-#################
     #######,{WHT}>>>>>>>";      :rem 252
1065 PRINT">>>>>>{BLU}-##################
     ########,{WHT}>>>>>>";      :rem 203
1075 PRINT">>>>>{BLU}-###################
     #########{WHT}>>>>>>";      :rem 168
1080 PRINT">>>>>{BLU}####################
     #########{WHT}>>>>>>";      :rem 154
1085 PRINT">>>>>{BLU}####################
     #########{WHT}>>>>>>";      :rem 159
1090 PRINT">>>>>{BLU}####################
     #########{WHT}>>>>>>";      :rem 155
1092 PRINT">>>>>{BLU}####################
     #########{WHT}>>>>>>";      :rem 157
1093 PRINT">>>>>{BLU}####################
     #########{WHT}>>>>>>";      :rem 158
1094 PRINT">>>>>{BLU}####################
     #########{WHT}>>>>>>";      :rem 159
1095 PRINT">>>>>{BLU}####################
     #########{WHT}>>>>>>";      :rem 160
1200 REM BUILD MAZE                :rem 68
1205 A(0)=2:A(1)=-80:A(2)=-2:A(3)=80:WL=3
     5:HL=32:A=SC+846              :rem 19
1210 POKEA,4                      :rem 148
1220 J=INT(RND(1)*4):X=J          :rem 101
1230 B=A+A(J):IFPEEK(B)=WLTHENPOKEB,J:POK
     EA+A(J)/2,HL:A=B:GOTO1220      :rem 8
1240 J=(J+1)*-(J<3):IFJ<>XTHEN1230
                                  :rem 128
1250 J=PEEK(A):POKEA,HL:IFJ<4THENA=A-A(J)
     :GOTO1220                     :rem 34
1260 FORI=SC+216TOSC+856STEP40:POKEI,32:P
     OKEI+7,32:NEXT               :rem 152
1270 FORI=SC+140TOSC+860STEP40:POKEI,32:N
     EXT                          :rem 235
1280 FORI=SC+372TOSC+852STEP40:POKEI,32:P
     OKEI+15,32:NEXT              :rem 200
1290 FORI=SC+489TOSC+849STEP40:POKEI,32:P
     OKEI+21,32:NEXT              :rem 213
1291 FORI=SC+372TOSC+387:POKEI,32:NEXT
                                   :rem 89
1292 FORI=SC+489TOSC+510:POKEI,32:NEXT
                                   :rem 87
1293 FORI=SC+687TOSC+712:POKEI,32:POKEI+1
     60,32:NEXT                    :rem 97
1300 REM PLACE GAME CHARACTERS      :rem 7
1310 REMPLACE GHOSTS              :rem 230
1320 FORI=PTO3*AA                   :rem 5
1330 X=INT(RND(1)*680)+SC+40       :rem 22
1340 BL=0:GOSUB1700:IFBLTHEN1330 :rem 146
1350 POKEX,59:POKECM+X,1:NEXT      :rem 60
1400 REMPLACE BATS                 :rem 56
1420 FORI=PTO3*AA                   :rem 6
1430 X=INT(RND(1)*680)+SC+40       :rem 23
1440 BL=0:GOSUB1700:IFBLTHEN1430 :rem 148
1450 POKEX,61:POKECM+X,5:NEXT      :rem 58
1500 REMPLACE CATS                 :rem 58
1520 CC=11:FORI=PTO10             :rem 232
1530 X=INT(RND(1)*680)+SC+40       :rem 24
1540 BL=0:GOSUB1700:IFBLTHEN1530 :rem 150
1550 POKEX,60:POKECM+X,7:NEXT      :rem 60
1600 REMPLACE SPIRITS              :rem 62
1620 FORI=PTOAA*2                   :rem 7
1630 X=INT(RND(1)*680)+SC+40       :rem 25
1640 IFPEEK(X)<>32THEN1630          :rem 0
1650 POKEX,31:POKECM+X,4:A(I)=X:NEXT:RETU
     RN                           :rem 252
1700 IF(PEEK(X)<>32)OR(PEEK(X+P)<>32ANDPE
     EK(X+P)<>35)THENBL=1         :rem 152
1710 IF(PEEK(X-P)<>32ANDPEEK(X-P)<>35)OR(
     PEEK(X+Q)<>32ANDPEEK(X+Q)<>35)THENBL
     =1                            :rem 24
1720 IF(PEEK(X-Q)<>32ANDPEEK(X-Q)<>35)THE
     NBL=1                          :rem 1
1730 RETURN                       :rem 171
1800 REM SCARED!                  :rem 128
1810 IFCF=OTHEN1840               :rem 106
1820 X=INT(RND(1)*680)+SC+40       :rem 26
1830 IFPEEK(X)<>32THEN1820          :rem 2
1835 POKEX,60:POKECM+X,7:CF=O:SR=SR-2*AA↑
     2:IFSR<OTHENSR=O             :rem 205
1840 POKETL,32:POKECL,58:POKECL+CM,1
                                  :rem 206
1843 FORI=15TO1STEP-1:POKESL,100:POKESH,1
     00:POKEWF,33:POKEV,I          :rem 99
1844 FORII=1TO25:NEXTII,I          :rem 66
1845 FORI=1TO400:NEXT:POKEWF,32  :rem 183
1850 TL=CL:Z=TL:ONINT(RND(1)*4)+1GOSUB301
     ,303,305,307                  :rem 62
1860 CL=Z:SR=SR-AA↑2:IFSR<OTHENSR=O
                                   :rem 78
1870 GOTO400                      :rem 157
1900 REM GOTCHA! SOUND             :rem 14
1905 POKETL,32:POKECL,58:POKECM+CL,4:DF=1
                                    :rem 5
```

```
1910 POKESL,100:FORI=1TO4:POKEWF,17:POKES
     H,25:FORX=1TO150:NEXT:POKEWF,16
                                    :rem 104
1911 FORX=1TO75:NEXT                 :rem 1
1913 IFI=1ORI=3THENPOKECL,31:GOTO1920
                                    :rem 42
1914 POKECL,58                      :rem 38
1920 POKESL,0:FORX=1TO200:NEXT:NEXT:FORX=
     1TO500:NEXT:RETURN             :rem 182
2000 REM LINE UP SAVED CAT          :rem 190
2010 X=SC+985:POKEX-CC,60:POKECM+X-CC,7:C
     C=CC-1:SR=SR+10*AA↑2:CF=0      :rem 16
2020 POKEWF,33:FORI=1TO50:NEXT:RETURN
                                    :rem 149

9000 DATA120,8,72,152,72,138,72,173,19,14
     5,72,173,34,145,72,169,0,141,62,3,14
     1                              :rem 49
9005 DATA63,3,169                   :rem 124
9010 DATA127,141,34,145,173,32,145,73,255
     ,41,128,42,8,169,195,141,19,145,173,
     17                             :rem 115
9015 DATA145,73                      :rem 25
9020 DATA255,41,60,74,74,40,42,168,41,16,
     201,16,208,3,141,63,3,152,41,15,162,
     0                              :rem 24
9025 DATA232,224                     :rem 69

9030 DATA9,240,8,221,160,3,208,246,142,62
     ,3,104,141,34,145,104,141,19,145,104
                                    :rem 232
9035 DATA170,104                     :rem 68
9040 DATA168,104,40,88,96,2,3,1,5,4,12,8,
     10                             :rem 105
10000 DATA255,255,255,255,255,255,255,255
                                    :rem 31
10010 DATA0,0,0,0,0,15,63,255       :rem 155
10015 DATA0,0,0,0,0,240,252,255      :rem 0
10020 DATA1,1,3,3,7,7,7,7           :rem 229
10030 DATA128,128,192,192,224,224,224,224
                                    :rem 16

10040 DATA7,7,7,7,3,3,1,1           :rem 231
10050 DATA224,224,224,224,192,192,128,128
                                    :rem 18
10060 DATA255,63,15,0,0,0,0,0       :rem 160
10070 DATA255,252,240,0,0,0,0,0      :rem 1
10080 DATA128,192,224,240,248,252,254,255
                                    :rem 26
10090 DATA1,3,7,15,31,63,127,255    :rem 76
10100 DATA255,255,255,255,250,246,244,224
                                    :rem 21

10110 DATA255,255,191,63,15,15,7,63
                                    :rem 233
10120 DATA240,249,240,228,0,252,255,255
                                    :rem 170
10130 DATA255,255,127,35,1,112,63,255
                                    :rem 71
10140 DATA0,102,0,24,0,0,28,0       :rem 151
10150 DATA56,84,56,16,124,186,40,108
                                    :rem 33

10160 DATA62,42,62,28,28,28,60,120
                                    :rem 179
10170 DATA40,124,85,125,57,57,61,127
                                    :rem 32
10180 DATA0,16,124,254,214,130,0,0
                                    :rem 156
10190 DATA0,0,0,0,0,0,0,0           :rem 201
20000 JP=15-PEEK(56320)AND15         :rem 11
20010 PRINTJP:GOTO20000             :rem 127
```

Machine Language For Beginners

(Article on page 129.)

Program 2: VIC Version

```
10 I=12288                          :rem 236
20 READ A:IF A=256 THEN 50           :rem 55
30 POKE I,A:CK=CK+A:I=I+1:GOTO 20:rem 129
40 END                              :rem 59
50 IF CK<>11469 THEN PRINT "ERROR IN DATA
    STATEMENTS":STOP                :rem 196
12288 DATA 160,0,169,8,153,0,148    :rem 44
12295 DATA 153,0,149,200,208,247,160
                                    :rem 236
12302 DATA 0,169,224,153,0,16,153   :rem 75
12309 DATA 228,17,200,192,22,208,245
                                    :rem 236
12316 DATA 169,21,133,71,169,16,133
                                    :rem 191
12323 DATA 72,162,24,160,0,169,224
                                    :rem 132
12330 DATA 145,71,200,145,71,202,240
                                    :rem 219
12337 DATA 16,24,165,71,105,22,133
                                    :rem 132
12344 DATA 71,165,72,105,0,133,72   :rem 83
12351 DATA 76,38,48,169,20,133,204
                                    :rem 145
12358 DATA 32,155,224,164,98,185,149
                                     :rem 5
12365 DATA 15,201,224,240,244,169,90
                                    :rem 237
12372 DATA 153,149,15,198,204,208,235
                                     :rem 42
12379 DATA 96,256                    :rem 88
```

Program 3: 64 Version

```
10 I=49152                          :rem 236
20 READ A:CK=CK+A:IF A=256 THEN 50:rem 54
30 POKE I,A:I=I+1:GOTO 20           :rem 130
40 END                               :rem 59
50 IF CK<>12749 THEN PRINT "ERROR IN DATA
    STATEMENTS":STOP                :rem 198
49152 DATA 160,0,169,8,153,0,216    :rem 40
49159 DATA 153,0,217,153,0,218,153
                                    :rem 142
49166 DATA 0,219,200,208,241,160,0
                                    :rem 130
49173 DATA 169,224,153,0,4,153,192
                                    :rem 148
49180 DATA 7,200,192,40,208,245,169
                                    :rem 198
49187 DATA 39,133,71,169,4,133,72:rem 110
49194 DATA 162,24,160,0,169,224,145
                                    :rem 197
49201 DATA 71,200,145,71,202,240,16
                                    :rem 175
49208 DATA 24,165,71,105,40,133,71
                                    :rem 140
49215 DATA 165,72,105,0,133,72,76   :rem 95
49222 DATA 44,192,169,20,133,204,32
                                    :rem 189
49229 DATA 158,224,164,98,185,168,3
                                    :rem 222
49236 DATA 201,224,240,244,169,90,153
                                     :rem 39
49243 DATA 168,3,198,204,208,235,96,256
                                    :rem 158
```

COMPUTE!'s Gazette for Commodore

AUTHOR GUIDE

COMPUTE!'s Gazette for Commodore is looking for interesting, useful articles aimed at beginning to intermediate VIC-20 and Commodore 64 users. If you have an article idea or a good original program, we'd like to see it. Don't worry if you are not a professional writer. We are more concerned with the content of an article than its style. Simply try to be clear in your writing and check your program for any bugs.

COMPUTE!'s Gazette for Commodore is a consumer-oriented magazine for VIC-20 and Commodore 64 users who want to get the most out of their computers in a non-technical way. It is aimed primarily at home users, not all of whom necessarily want to become expert programmers. If your article covers a more advanced or technical topic, you may choose to submit it to our companion publication, **COMPUTE!**. If you submit an article to one of our magazines and we believe it would be more suitable to the other, we will transfer your submission to the right editors. The basic editorial requirements for publication are the same for both magazines; so are the payment rates.

The following guidelines will permit your good ideas and programs to be more easily edited and published. Most of these suggestions serve to improve the speed and accuracy of publication:

1. The upper left corner of the first page should contain your name, address, telephone number, and the date of submission.

2. The following information should appear in the upper right corner of the first page. If your article is specifically directed to either the VIC-20 or Commodore 64, please state which one. In addition, please indicate the memory requirements of programs.

3. The underlined title of the article should start about 2/3 of the way down the first page.

4. Following pages should be typed normally, except that in the upper right corner there should be an abbreviation of the title, your last name, and the page number. For example: Memory Map/Smith/2.

5. Short programs (under 20 lines) can easily be included within the text. Longer programs should be separate listings. *It is essential that we have a copy of the program, recorded twice, on a tape or disk.* The tape or disk should be labeled with your name and the title of the article. Tapes are fairly sturdy, but disks need to be enclosed within plastic or cardboard mailers (available at photography, stationery, or computer supply stores).

It is far easier for others to type in your program if you use CHR$(X) values and TAB(X) or SPC(X) instead of cursor manipulations to format your output. For five carriage returns, FOR I = 1 TO 5:PRINT:NEXT is far more "portable" to other computers with other BASICs and also easier to type in. And, instead of a dozen right-cursor symbols, why not simply use PRINT SPC(12)? A quick check through your program – making these substitutions – would be greatly appreciated by your editors and by your readers.

6. If your article is accepted and you have since made improvements to the program, please submit an entirely new tape or disk and a new copy of the article reflecting the update. We cannot easily make revisions to programs and articles. It is necessary that you send the revised version as if it were a new submission entirely, but be sure to indicate that your submission is a revised version by writing "Revision" on the envelope and the article.

7. All lines within the text of the article should be spaced so that there is about 1/2 inch between them. A one-inch margin should be left at the right, left, top, and bottom of each page. No hyphens should be used at the ends of lines to break words. And please do not justify. Leave the lines ragged.

8. Standard typing paper should be used (no onionskin or other thin paper) and typing should be on one side of the paper only (upper- and lowercase).

9. Sheets should be attached together with a paper clip. Staples should not be used.

10. A good general rule is to spell out the numbers zero through ten in your article and write higher numbers as numerals (1024). The exceptions to this are: Figure 5, Table 3, TAB(4), etc. Within ordinary text, however, the zero through ten should appear as words, not numbers. Also, symbols and abbreviations should not be used within text: use "and" (not &), "reference" (not ref.), "through" (not thru).

11. For greater clarity, use all capitals when referring to keys (RETURN, TAB, ESC, SHIFT), BASIC words (LIST, RND, GOTO), and three languages (BASIC, APL, PILOT). Headlines and subheads should, however, be initial caps only, and emphasized words are not capitalized. If you wish to emphasize, underline the word and it will be italicized during typesetting.

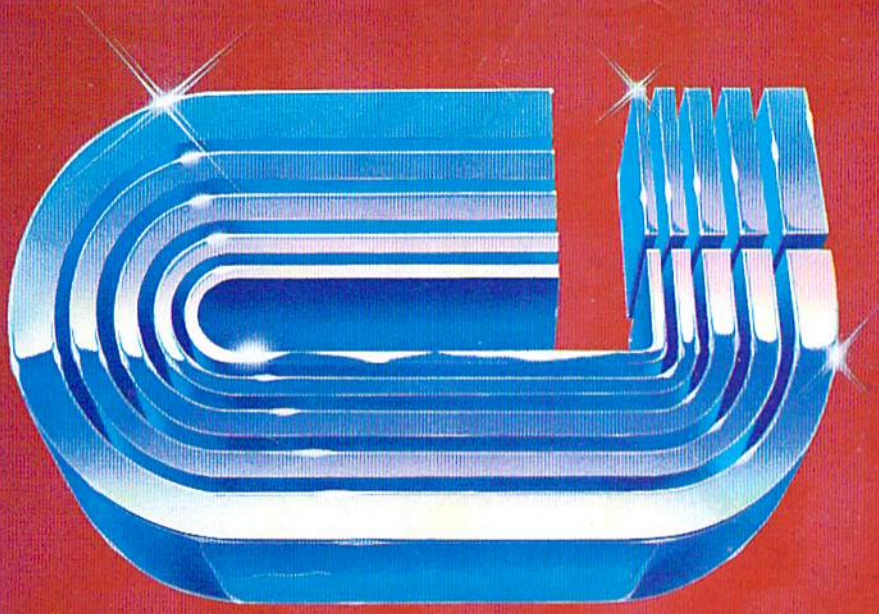
12. *COMPUTE!'s Gazette* for Commodore pays between $75 and $1000 for published articles. In general, the rate reflects the length and quality of the article. Payment is made upon acceptance of an article. Following submission (Editorial Department, *COMPUTE!'s Gazette* for Commodore, P.O. Box 5406, Greensboro, NC 27403) it will take from four to six weeks for us to reply. If your work is accepted, you will be notified by a letter which will include a contract for you to sign and return. Rejected manuscripts are returned to authors who enclose an SASE. We do not consider articles which are multiple submissions. If you wish to send an article to another magazine for consideration, please do not submit it to us.

13. Articles can be of any length – from a single-line routine to a multi-issue series. The average article is about four to eight double-spaced, typed pages.

14. If you want to include photographs, they should be 5x7, black-and-white glossies.

ADVERTISERS INDEX

The ULTIMATE Printer Interface?



We hope so, but because we have 2-1/2 technicians answering four incoming customer service phone lines, we have learned that just when you think the product is perfect some programmer finds a new way to do things and proves you wrong! When we at CARDCO, Inc. are told of a problem, we try to incorporate the cure in all future production. And as our customers will attest, we do not leave owners of older versions out in the cold. When an upgrade is made in the production version of our interface, we make the upgrade available to all owners of that interface, AT NO CHARGE! Free technical support, no charge product upgrades and a lifetime guarantee, we dare anyone to do a better job of customer support.

That's all very nice, but what's all this about the ULTIMATE printer interface? While answering your technical questions our customer service technicians listened to what you wanted. You wanted to be able to print the full Commodore character set with Commodore graphics, reversed characters and reversed graphics. You wanted compatability with Commodore's normal tab functions and high resolution dot space tab functions. You wanted to be able to use the

Commodore high resolution dot addressable graphics commands. And you wanted to run all existing programs without modification and without giving up the extra features and special functions of your printer.

The CARD/?+G has DIP switch selection for the following fine printers:

- Prowriter
- C-Itoh 8510
- NEC 8023
- Epson MX-80/100
- Epson RX-80/100
- Epson FX-80/100
- Star Gemini 10X
- Okidata 82/83/84
- Okidata 92/94
- Axiom GP/100
- Gorilla Banana

In response to your demand CARDCO, Inc. proudly presents the CARD/?+G (CARDPRINT+G). Why is it the ULTIMATE printer interface? Because it is "state of the art" today and because of our strong committment to customer service it will stay that way for all your tomorrows.

The CARD/?+G is available now from your local retailer. Suggested retail $89.95.

If you own a version of the original CARD/?A, we are sorry the CARD/?+G is a totally new product and you will not be allowed a free upgrade. But if you want the capabilities of the new CARD/?+G we do have a trade up policy, please contact our customer service department for details.

If you don't need the graphics capabilities of the CARD/?+G be sure to check out the new CARD/?B. The "B" model offers all of the same features that have made the CARD/?A the #1 selling printer interface in an economy package. The CARD/?B is compatible with programs not requiring graphics functions (ie. Word Processors, Spread Sheets, etc.) and fully supported by our customer service department and the suggested retail price is only $49.95.

313 Mathewson • Wichita, Kansas 67214 • (316) 267-6525

CARDCO, Inc.

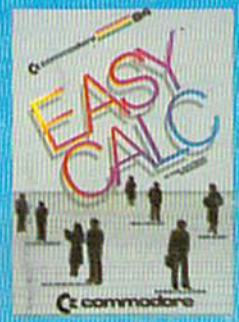
commodore 64

INDISPENSABLE SOFTWARE

For Your Most Important Computing Needs

EasyScript 64
Displays 764 lines × 240 characters. Prints to 130 columns. Works with EasySpell 64.

EasySpell 64
20,000 word Master Dictionary and automatic spelling checker. Works with EasyScript 64.



EasyCalc 64
Multiple electronic spreadsheet with color bar graph feature. 63 columns × 254 rows.

The Manager
Sophisticated database system with 4 built-in applications, or design your own. Text, formulas, graphics.

SuperExpander 64
21 special commands. Combine text with high resolution graphics. Music and game sounds.

Easy Finance I—Loan Analysis
12 loan functions. Bar graph forecasting as well as calculation.

Easy Finance II—Basic Investment Analysis
16 stock investment functions. Investment bar graph.

Easy Finance III—Advanced Investment Analysis
16 capital investment functions. Bar graphs.

Easy Finance IV—Business Management
21 business management features. Bar graphs.

Easy Finance V—Statistics and Forecasting
Assess present/future sales trends with 9 statistics and forecasting functions.

Accounts Payable/ Checkwriting
11 functions. Automatic billing. 50 vendors/disk.

Accounts Receivable/Billing
11 billing functions. Printed statements.

General Ledger
8 general ledger options. Custom income statement, trial balances, reports.

Inventory Management
1000 inventory items. Full reports.

Payroll
24 different payroll functions. Integrated with G/L system.

commodore COMPUTERS

First In Quality Software

www.commodore.ca