

loop slowly decreases the volume, fading out the train sound.

The UFO sound is unusual. It's a series of tone bursts and silences in which the pitch of the tone sweeps up and down. Without the silences between each tone, the effect is less interesting.

Morse code is simulated by a series of tone bursts and silences of random duration.

Here are the functions of the program lines:

Lines 100-120 color the screen and print the title.

Lines 140, 260, 270, 330, 340, 450, 460, 490, 500, 530, 540, and 550 are story captions.

Lines 150-190 are the musical introduction.

Line 190 trills the last note of the intro.

Lines 200-220 sweep the noise rapidly downward and flash the screen white (to simulate lightning).

Line 230 produces a low rumble that fades out (to simulate thunder).

Lines 240-250 perform another lightning-and-thunder subroutine.

Lines 290-320 are the train sound.

Lines 360-440 are the UFO sound, interspersed with screen-color changes.

Lines 470-480 simulate Morse code.

Lines 510-520 simulate a dropping ping-pong ball.

Lines 570-590 display a night sky full of stars.

Lines 600-630 simulate crickets and PRINT "The End."

Line 640 holds The End on the screen for a few seconds, then resets the screen to its normal condition.

## The Sound Of Snow

Type in and run the following program. It's the sound of someone walking in deep snow. As a walker steps into snow, the snow gives way with a noise burst. As the walker puts his full weight down, the snow becomes compacted and the bursts increase in frequency. In other words, a footstep or crunch is a series of noise bursts that accelerate like a train, except much faster.

```
10 POKE36878,15:S=36877
20 FORL=15TO1STEP-1
30 POKES,200:FORL=1TOL:NEXT:POKES,0:FORL=
  1TOL:NEXT:NEXT
40 FORL=1TO200:NEXT:GOTO20
```

If you'd rather not type in the program, send \$3, a blank cassette with Sound Story printed on the label, and a self-addressed stamped envelope to:

Bruce Bartlett  
51941 Jenny Lane  
Elkhart, Indiana 46514

See program listings on page 155. @

## 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.

**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.

Back issues of July and August 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 Back Issues**  
P.O. Box 5406  
Greensboro, NC 27403

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



# Joystick Control For The 64

Sterling N. Augustine

This tutorial on how to use the 64's joystick ports includes a two-player game, "Chase And Tag," which illustrates the way the joystick ports are read.

The joystick ports on the 64 transmit data to registers at memory locations 56321 (port 1) and 56320 (port 2). A register is a memory location used for input or output instead of storing data. A simple PEEK at one of these locations gives you a value which corresponds to the position of the joystick. Try plugging a joystick into port 1 and RUNNING the following:

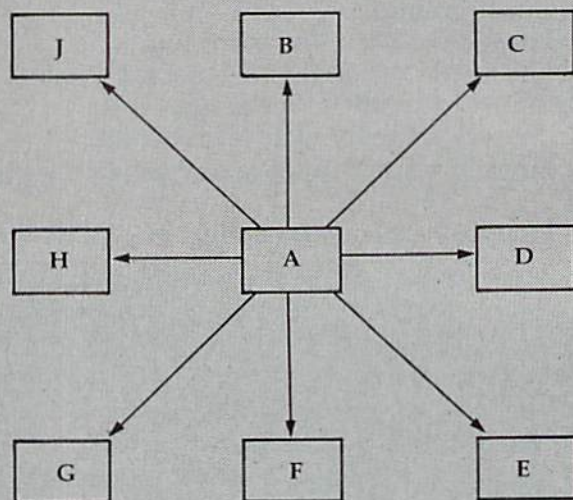
```
10 PRINT PEEK (56321)
20 GOTO 10
```

Note how fast the value changes as you move the joystick. Now hold the stick in one position and press the fire button. If you play around a bit you will see that each position has two possible values depending on whether or not the fire button is depressed. Below are a table and figure which summarize all possible values for the two port registers, and the joystick fire button status which they represent.

Joystick Control Values		
Control Port 1		Control Port 2
	W/O Fire	W/ Fire
A	255	239
B	254	238
C	246	230
D	247	231
E	245	229
F	253	237
G	249	233
H	251	235
J	250	234

Control Port 2		Control Port 1
	W/O Fire	W/ Fire
A	127	111
B	126	110
C	118	102
D	119	103
E	117	101
F	125	109
G	121	105
H	123	107
J	122	106

Joystick Movement Schematic



## Chase And Tag

"Chase and Tag" is a two-player game using simple keyboard graphics which illustrates the way the joystick ports are read. The object of the game is to score points by touching the opposing player while you are *it*, and to avoid the opposing player while he is *it*. The left player (controlled by the joystick in port 2) is *it* first. The computer keeps track of who is *it*, automatically switching after a point is scored, or after each player has made 100 moves without a tag, whichever comes first. When the player who is *it* catches the other player, a colorful explosion takes place. The game ends when one player scores 20 points.

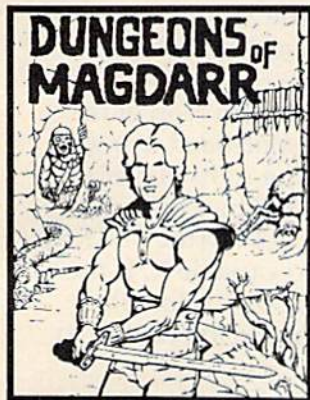
Here's a breakdown of the program's organization:

Line 40 POKes the screen border color to purple, the screen background color to light blue, the screen character color to white, erases all variables in memory, and branches to line 700.



# AARDVARK LTD.

## VIDEO ADVENTURES™



**DUNGEONS OF MAGDARR** - Serious D of D for up to 8 players. Features full 3d GRAPHICS! You get a choice of several characters that grow from game to game and are interchangeable with characters from our famous Dungeons of Death game. A real dungeon with level after level of monsters to conquer and treasures to find - all in hi-res 3d graphics.

Available On: TRS80C, IBM PC, CMD64

TAPE \$19.95

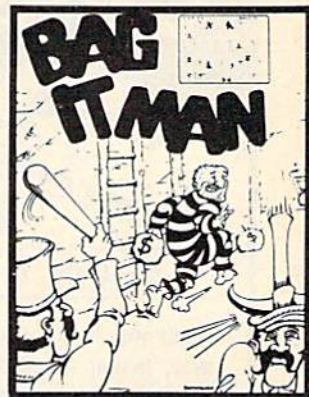
DISK \$24.95

**BAG-IT-MAN** - The ultimate arcade game for TRS80C or MCD64. This one has three screens full of BAGS OF GOLD, CARTS & ELEVATORS TO RIDE IN, MINE SHAFTS, and TWO NASTY GUARDS. Great sound and color and continuous excitement.

Available On: TRS80C 32K, CMD64

TAPE \$19.95

DISK \$24.95



**QUEST** - A different kind of Graphic Adventure, it is played on a computer generated map of Alesia. You'll have to build an army and feed them through combat, bargaining, exploration of ruins and temples, and outright banditry! Takes 2 - 5 hours to play and is different each time.

Available On: TRS80C 16K, CMD64, VIC20 13K, MC10 16K, T199 (EXT. BASIC), IBMPC

TAPE \$14.95

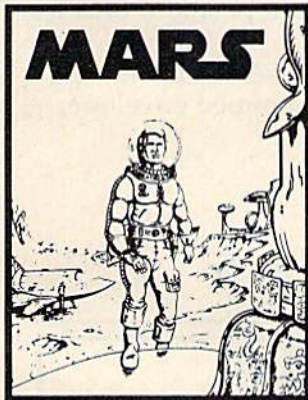
DISK \$19.95

**STARFIRE** - If you enjoyed Star Raiders or Star Wars, you will love Starfire. It is not a copy, but the best shoot-em-up, see them in the window space game on the CMD64 or TRS80C. The fantastic graphics will put you right in the control room as you hyperspace from quadrant to quadrant fighting the aliens and protecting your bases.

Available On: TRS80C 16K, CMD64

TAPE \$19.95

DISK \$24.95



**MARS** - Your ship crashed on the Red Planet and you have to get home. You will have to explore a Martian City, repair your ship, and deal with possibly hostile aliens to get home again. This is recommended as a first Adventure. It is in no way simple - playing time normally runs from 30 to 50 hours, but it lets you try out Adventuring before you battle the really tough ones. Full Graphics Adventure.

Available On: TRS80C, CMD 64, IBM PC

TAPE \$19.95

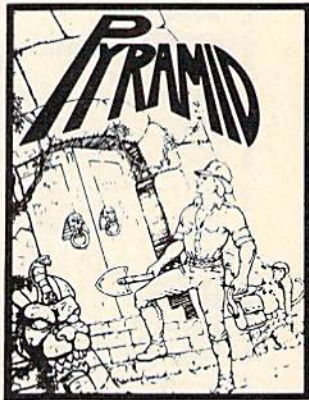
DISK \$24.95

**PYRAMID** - ONE OF THE TOUGHEST ADVENTURES. Average time through the pyramid is 50 - 70 hours. Clues are everywhere and some ingenious problems make this popular around the world. FULL GRAPHIC ADVENTURE.

Available On: TRS80C 16K, CMD64, MC10 16K, IBM PC

TAPE \$19.95

DISK \$24.95



## NEW! GRAPHIC ADVENTURES

AARDVARK offers over 120 original high quality programs. Send one dollar for a current catalog and receive a \$1.00 gift certificate good towards your next purchase.

Authors - AARDVARK pays top dollar for high quality programs. Send a copy today for a personal review and editorial help.

TO ORDER: Send amount indicated plus \$2.00 shipping, per order. Include quantity desired and your preference of tape or disk. Be sure to indicate type of system and amount of memory. When using charge card to order by mail, be sure to include expiration date.



CHARGE CARDS  
WELCOME



1-313-669-3110

PHONE ORDERS ACCEPTED

8:00 a.m. to 8:00 p.m. E.S.T., MON-FRI

AARDVARK *Action Software*

2352 S. COMMERCE • WALLED LAKE, MI 48088 • (313) 669-3110



# YORK 10<sup>TM</sup> CASSETTES



## COMPUTER GRADE DATA TRAC BLANK CASSETTES

C-05, C-06, C-10, C-12, C-20, C-24, C-32

From the leading supplier of Computer Cassettes, new, longer length C-12's (6 minutes per side) provide the extra few feet needed for some 16K programs.



**BASF-LHD (DPS) world standard tape.**  
Premium 5 screw shell with leader.



**Internationally acclaimed.**  
Thousands of repeat users.



**Error Free • Money back guarantee.**

**\* 500 C-12's or C-10's — 38¢ each**  
(w/labels, add 4¢ • Shipping \$17./500  
500 Boxes 13¢ ea. • Shipping \$10./500  
(Free Caddy offer does not apply)



**TRACTOR FEED • DIE-CUT  
BLANK CASSETTE LABELS**  
WHITE: \$3.00/100 \$20.00/1000  
**COLORED LABELS • Pastels -**  
Red, Blue, Green, Yellow, Lavender  
\$4.00/100 \$30.00/1000

**CASSETTE STORAGE CADDY**  
Holds 12 cassettes  
w/o boxes  
Includes edge labels  
and index card  
**\$2.95**



**FREE**  
**1 CADDY WITH EVERY  
4 DOZ. CASSETTES PURCHASED**  
(does not apply to 500 quantity offer)



**Call: 213/700-0330** for IMMEDIATE SHIPMENT  
on Credit Card Orders



**ORDER  
NOW...**



**YORK 10<sup>TM</sup>** 9525 Vassar Ave. #G  
Chatsworth, CA 91311

### ORDER FORM

ITEM	1 DOZEN	2 DOZEN	TOTAL
C-05	<input type="checkbox"/> 7.00	<input type="checkbox"/> 13.00	
C-06	<input type="checkbox"/> 7.00	<input type="checkbox"/> 13.00	
C-10	<input type="checkbox"/> 7.50	<input type="checkbox"/> 14.00	
C-12	<input type="checkbox"/> 7.50	<input type="checkbox"/> 14.00	
C-20	<input type="checkbox"/> 8.75	<input type="checkbox"/> 17.00	
C-24	<input type="checkbox"/> 9.00	<input type="checkbox"/> 17.00	
C-32	<input type="checkbox"/> 11.00	<input type="checkbox"/> 21.00	
Hard Box	<input type="checkbox"/> 2.50	<input type="checkbox"/> 4.00	
White Labels	<input type="checkbox"/> 3.00/100	<input type="checkbox"/> 20.00/1000	
Colored Labels	<input type="checkbox"/> 4.00/100	<input type="checkbox"/> 30.00/1000	
DESCRIPTION	PRICE	QUANTITY	
Storage Caddy	2.95		
		<b>SUB TOTAL</b>	
Calif. residents add sales tax			
Shipping/handling (any quantity — using prices above)			3.50
Outside 48 Continental States — Additional \$1 per caddy; per doz. cassettes or boxes.			
		<b>TOTAL</b>	

Each cassette includes two YORK 10 labels only. Boxes are sold separately. We prefer to ship by UPS as being the fastest and safest. If you need shipment by Parcel Post, check here ☐.

**NOTE:** Additional charges outside 48 Continental States. Shipments to AK, HI, and USA possessions go by Priority Mail; Canada & Mexico — Airmail; All others — Sea Mail.

**Ask about our  
DUPLICATING  
SERVICE**

CHECK OR M.O. Charge to  
ENCLOSED ☐ Credit Card: VISA ☐ MASTERCARD ☐  
☐ PLEASE SEND QUANTITY DISCOUNTS

Card No. \_\_\_\_\_ Exp. \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State/Zip \_\_\_\_\_

Signature \_\_\_\_\_ Phone \_\_\_\_\_

Computer make & model \_\_\_\_\_ Disk? (y/n) \_\_\_\_\_

Lines 700–810 allow the two players to print their initials on the screen above their scores. Values are assigned to several other variables used in the program.

Lines 900–1230 print the title of the game on the screen, draw the playing field, reset both scores to zero, and branch to line 55.

Lines 55–70 POKE the movement values into memory.

Lines 100–290 make up the main loop of the program. Player movement is controlled using the indirect address method. The right player's direction is determined in line 140 by PEEKing location 56321, adding a value 52000 to the PEEKed value and assigning the total to variable X. If the fire button is pressed, the value of X is less than 52245, so the program waits for the fire button to be released.

This trick can be used to confuse the other player so that you can change direction when you are about to be tagged. If the fire button is not pressed, the value 41 is subtracted from the value of X. The result is assigned to the variable M; this is the actual movement step.

For example, assume that the right player's first move will be up and to the left. Joystick position J (see the figure) will put the value 250 into register 56321. PEEKing at this register and adding 52000 to the result will give a value of 52250 to X. PEEKing location 52250 and subtracting 41 from the result (0) assigns the value of -41 to M. Since the move in this example will not take the player outside the border, lines 180 to 200 will allow the player to move to his new screen position:  $1422 + (-41) = 1371$ .

For a copy of the program, send a blank tape or diskette, a self-addressed, stamped envelope, and \$3 to:

*Sterling N. Augustine*  
115½ Parkway  
Schuylkill Haven, PA 17972

See program listing on page 156.

## 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

**916-726-8793**

VISA, MASTERCARD, AND MONEY ORDERS  
CA RESIDENTS ADD 6% SALES TAX.  
VIC IS A TRADEMARK OF COMMODORE



# Print Sound For The VIC-20

Alejandro A. Kapauan

**This utility translates letters into music. All you supply is a simple PRINT statement.**

"Print Sound" is a machine language (ML) utility that enables you to produce musical tones on the VIC with simple PRINT statements. Using the program requires no knowledge of ML programming, although intermediate ML programmers may be interested in examining the code.

To use Print Sound, type in the BASIC program. Be very careful with the DATA statements, and SAVE the program before RUNning it. When you RUN the program, it prints a greeting, pauses, then plays a few notes. It then plays a short classical tune. If the program fails to work, or if the VIC locks up, LOAD your SAVED copy of the program, LIST it, and check the DATA statements. Make the necessary corrections and again SAVE the program.

When you RUN the program successfully, VIC device 2 (normally the RS-232 port) is re-defined, so that you can print strings of letters to it to play your own music.

## Letters Equal Notes

If you examine the program, you will see that logical file 1, which is opened to send data to device 2, is not closed. After RUNning the program, type the command `PRINT#1,"ABCDEFGHIJKLM"` in the immediate mode. You will hear a chromatic scale. A chromatic scale is the series of 12 notes that are sounded when you play one octave of successive black and white keys on the piano, for example C, C#, D, D#, and so on to the next C. Printing a letter from A to Y will cause a note in the chromatic scale to be played. The letter Z is

silent, and produces a rest. Experiment further by printing various alphabetic strings with the command `PRINT#1,"string"`.

To incorporate the utility into your own programs, just include lines 500 to 640, and include a `GOSUB630` at the beginning of your program.

The `PRINT#1` commands are completed even before the VIC is finished playing all the notes. This is because the notes are placed in the 255 character buffer usually reserved for the RS-232 transmit channel, and an interrupt routine does the actual transfer of data to the VIC sound registers. To synchronize the notes with your program, you may examine the contents of location 983. This location contains the current number of notes in the buffer. Line 50 of the sample program illustrates how to do the synchronization. If you want to close the file to device 2 in your program, wait until the buffer is empty first. Most of the time you can just leave the file open.

The rate at which notes are played can be modified by changing the thirteenth number in line 570 (third from the end), which is normally a 10, to some other value. This number represents the duration of each note in jiffies (1/60 second). Try changing the value to a 5. You'll see how fast the VIC can play a tune.

Since the ML program resides in the cassette buffer, you can issue a `NEW` command and the utility will not disappear. It will function until you warm start the VIC by holding down the `RUN/STOP` key and hitting `RESTORE`. A word of warning for cassette users: You must not perform cassette operations with the utility in place. Warm start the VIC before doing a cassette `LOAD` or `SAVE`. Disk users do not have this problem.

See program listing on page 157. ☐



## Sound On The VIC-20

This month we'll be discussing the fundamentals of sound on the VIC-20. We'll explore the VIC chip, and give you the basics of producing sound and music. So turn up the volume on your monitor and away we'll go.

Sound effects and music can add a lot to a computer program. Think how boring it would be to play a game like *Defender* if you couldn't hear the sounds of lasers blasting and enemy ships exploding. In business applications a "raspberry" sound can warn a user of bad input. Without a warning sound there would often be a potential for compounded errors.

You can communicate with the VIC chip inside your VIC-20 via memory locations 36864 to 36879. It is responsible for controlling the video and audio functions of your VIC. Specifically, memory locations 36874 to 36879 control sound, and by POKEing various values into these locations, you can produce almost infinite combinations of sounds and music on your VIC.

### POKEing Values For Sound

The VIC has three separate tone generators, a white noise generator, and a volume control. Each tone generator covers a range of three octaves, but because they overlap, the three together produce a range of five octaves. Sounds are produced on the VIC by POKEing the volume control location and then POKEing any value between 128 and 255 into one of the four sound generators. For ease of reference, these tone generators are often called speakers.

POKEing values below 128 (0-127) into the tone generating locations will produce no sound. This technique can be used to turn off an individual tone generator without turning down the volume. The volume control affects all three tone generators and the white noise generator. You can set the volume to any value between 0 and 15, with 15 being maximum volume. Here's a chart of the memory locations that control sound, and what they do.

Location	Voice Number	Noise Generated	Poke Values	Tone Range
36874	1	tone	128-255	low
36875	2	tone	128-255	medium
36876	3	tone	128-255	high
36877	-	white noise	128-255	noise
36878	-	volume control	0-15	-

Now that you have the basic information, let's POKE around a little and produce some sounds on the VIC. Before we continue, make sure that the volume setting on your TV or monitor is up. If you're using a monitor, make sure that all of the audio connections are made properly.

Enter POKE 36878,15 then press RETURN. This will put the volume at the maximum setting, but you still don't hear anything. Simply turning on the volume doesn't produce sound. To do that, we have to POKE a value into one of the tone generators. Now enter POKE 36874, 128. You should hear a very low tone, the lowest note that can be achieved on the VIC. Why? It's the lowest because we POKEd the lowest legal value (128) into the lowest of the three tone generators (36874). If you POKE values lower than 128 into any of the four sound producing registers, no



sound will be produced. To demonstrate this, POKE 36874,127 with the low sound still on.

Now press RUN/STOP—RESTORE. Among other things, this resets the VIC chip, setting all of the sound registers and the volume control to zero.

## Combining Sounds

When producing sound on the VIC, you are by no means limited to using one speaker at a time. To see how to combine sounds, we'll turn on one, two, then three of the speakers simultaneously.

First, let's turn on the volume by POKEing 36878,15. Now, let's turn on the speakers one at a time. POKE 36874,130 produces a very low tone by turning on the lowest speaker. POKE 36875,175 turns on the second (middle range) speaker, and you should be able to hear both tones at the same time. Now we'll turn on the third or highest speaker by POKEing 36876, 240. This adds a very high tone to the other two.

You can produce some interesting sound effects by using all three speakers simultaneously. You can turn off all three at once by pressing RUN/STOP—RESTORE or by turning off the volume with a POKE 36878,0. When you POKE off the volume, the speaker registers still contain sound producing values, but they're not audible with the volume off.

When working with sound, you are not limited to producing monotones. The frequency (note value) of the sound can be varied along with the volume. Enter and RUN this program:

```
10 POKE36878,15
20 FORA=128TO254:POKE36876,A:NEXT
30 FORA=254TO127STEP-1:POKE36876,A:NEXT
```

Unlike the other three speakers which produce musical tones, the white noise generator produces a hissing sound. For a demonstration, POKE 36878,15:POKE 36877,240. You hear a high-pitched hissing sound. This is known as *white noise*. The white noise speaker operates under the same rules as the other three speakers (turn on volume POKE values between 128 and 255, etc.). The noise speaker can be used to produce sounds such as an explosion or the thrust of a jet plane taking off. Here's a quick demonstration of an explosion:

```
5 POKE36878,15
10 FORA=254TO210STEP-1:POKE36876,A:FORT=1
  TO20:NEXTT:NEXTA
30 POKE36876,0
40 POKE36877,150:FORT=1TO500:NEXTT
50 FORA=150TO0STEP-1:POKE36878,A:FORT=1TO1
  50:NEXTT:NEXTA
```

Now that you have the basics, let's type in a few small sound demonstration programs and see what the VIC can do. On pages 136 through 138 of *Personal Computing on the VIC-20* (the in-

struction book that came with your VIC), you'll find many sound demo programs. A few of them—like #10: birds chirping, #9: phone ringing, and #12: ocean waves—are especially good. Studying the techniques used in these demonstration programs will teach you a lot about sound on the VIC.

## Programming Musical Notes

When you tire of the sounds of exploding spaceships, you can use the VIC to play tunes. Here is a chart of musical notes, and the values you need to POKE to produce them:

C	131	G	214
C#	140	G#	216
D	145	A	218
D#	151	A#	220
E	158	B	222
F	161	C	224
F#	166	C#	226
G	173	D	227
G#	178	D#	229
A	181	E	231
A#	185	F	232
B	189	F#	233
C	192	G	234
C#	197	G#	235
D	200	A	236
D#	203	A#	237
E	206	B	238
F	208	C	239
F#	211	C#	240

You'll notice that although a few of the notes have one value listed, the majority need two. In some cases, to produce a more pleasant and musically accurate note on the VIC you have to merge two tones. For example, the first G on the chart is achieved by using the two values 173 and 174.

This is done by alternately POKEing a speaker with two different values. For instance, to produce this G you would first POKE in the volume, then POKE the speaker with a value of 173, then a 174, a 173, and so on. This alternate beating of two notes in one speaker serves to correct notes that are slightly sharp or flat.

Here's a recognizable tune to show you how it's done:

```
10 POKE36878,15:S3=36876
20 READA,B,T:IFA=999THENPOKE36878,0:END
30 FORZ=1TOT
40 POKES3,A:POKES3,B:NEXT
50 GOTO20
60 DATA 224,224,100,239,240,50,218,219,50
70 DATA 214,214,50,206,207,50,214,214,150
80 DATA 200,200,150,999,999,999
```

Although the tune might be rather simple, study the programming techniques involved. In this program, both the value (frequency) of the notes




and the length of time the notes are played are READ via DATA statements. During each program cycle three values (A, B, and T) are READ. The note values (A and B) are POKEd into the speaker (S3=36876), and the time delay value (T) is used in the FOR/NEXT time delay loop. The program stops as soon as it reads a value of 999.

## Programming Hints

To save yourself some programming time as well as BASIC memory, set your speaker and volume registers as variables at the beginning of the program. For example:

```
10 S1=36874:S2=36875:S3=36876:S4=36877:V=36878
```

In this example, the variables S1 through S4 stand for speakers one through four, and V is the volume. After setting the registers as variables, referencing them in the body of the program is easy. For example, to turn on the volume, POKE V,15, or to POKE a value into one of the speakers, POKE S2,200. This not only saves programming memory, but also simplifies the program logic.

Although the VIC-20 does not have the sophisticated SID (sound interface device) like the 64, it can produce a myriad of sounds. With these techniques in mind you can add sound to that program you're writing, or even compose a symphony. 

## Modem with Printer Interface for Atari, Commodore, and Vic



Direct connect, autodial, autoanswer modem, also runs a parallel printer. 300 baud modem has built in Centronics printer port. Simultaneously prints whatever appears on your screen as it comes over the telephone.

One box does it all so you can receive the most from your computer.

Auto-Print Microconnection retails for \$149.95.

**the microp peripheral corporation**

2565 - 152nd Avenue NE, Redmond, WA 98052  
(206) 881-7544

## ASSEMBLY LANGUAGE

### C-64 PROGRAMMING AID

#### Code With Fewer Errors !

**Introducing** PSI-WARE'S Ultraspeed Macro Library of over 210 tested macros for + - \* / AND OR ON equals POKE PEEK IF and memory move to use with the C-64 macro assembler. 16 and 8 bit integer signed macros provide memory efficiency, speed and maximum versatility.

**Perfect** for games, sound, graphics and more.

**Compact** source code. Your easily inspected macro calls typically generate 20 bytes.

**New programmers** will appreciate how easy it is to program with the library.

**Professionals** develop your program faster and with fewer errors. Let the library do the grunt work.

**\$36.95** CA res. add 6.5%

Send check or M.O. to PSI-WARE, Dept. AG  
Box 1161, San Fernando CA 91341  
or Box 103, Labrador City NFLD Canada A2V 1N7



## ULTRA COPY 64

Make exact backup copies of Commodore 64 disks including DOS protection. Provides programmed instructions to reproduce disk errors used by latest protection schemes.

- Thoroughly tested, fast, reliable, easy to use
- Copy with one or two 1541 disk drives
- Copies EVERYTHING incl. name and ID
- If no disk errors encountered, copy will run
- Analyzes which tracks have data and errors
- Can skip empty tracks to greatly speed copying
- Programmed instructions to put errors 20, 21, 22, 23, 27 & 29 on copy as required

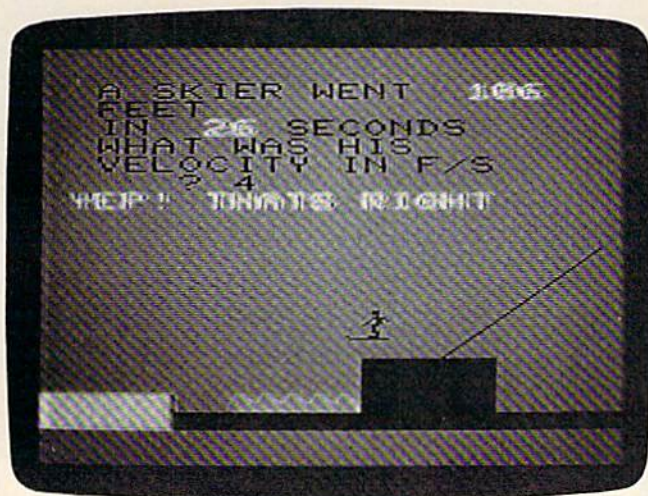
**98% OF SOFTWARE CAN BE ULTRACOPY'ED**

\$39.95 plus \$3 shipping. Mastercard and Visa Satisfaction guaranteed, 10 day return privilege.

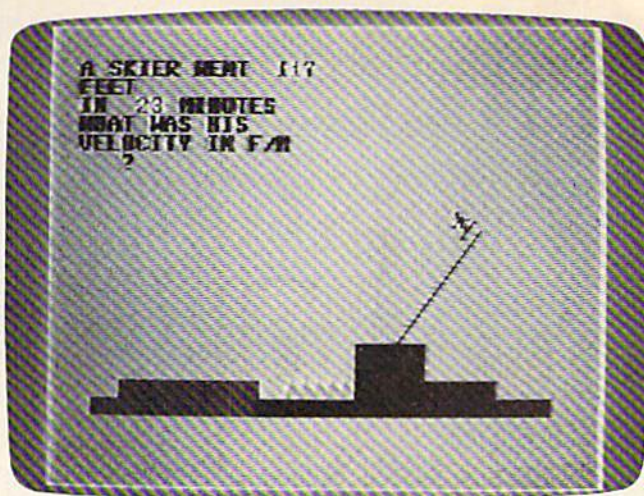
**ULTRABYTE** Call (313) 562-9855

23400 Michigan, Suite 502, Dearborn, MI 48124  
**SOFTWARE AUTHORS EARN UP TO 50% OF PROFITS**





*In the VIC version, a successful ski jump in progress after a correct answer.*



*The skier's fate hangs in the balance in the 64 version.*

# Ski Physics

Gerald and Betty Schueler

"Ski Physics" is an educational game that teaches the relationship between velocity, distance, and time. If you answer questions correctly, you can watch the skier make a perfect jump. If your answer is wrong, the ski jumper lands in a bale of hay. For the VIC-20 or Commodore 64.

"Ski Physics" begins by printing short definitions of time, velocity, and distance. The computer then randomly selects a word problem involving the three concepts. You solve the problem, enter the answer, and press RETURN. If the answer is correct, the ski jumper lands safely on a platform. But if you are wrong, the skier falls short and lands in a pile of hay.

You might want to have paper and pencil (or a calculator) handy; some questions require more effort than others. You should enter only numbers for your answer—not the units such as feet, seconds, miles per hour, etc. You can also ignore remainders in the division problems. The computer expects answers that are integers.

## How The VIC Version Works

Program 1 plays the opening music, sets up the custom characters in a protected area of memory, and prints the instructions. It then automatically loads Program 2.

In Program 2, line 255 selects four random

numbers—A represents the units for each problem (feet, minutes, etc.), B stands for one of three word problems, and variables X and Y are the numbers used in the problems.

Lines 290-300 print the background scene, using the redefined characters from the first program.

Line 315 accepts the player's input (answer). The program jumps to line 360 if the answer is correct; the skier makes a successful jump. If the answer is wrong, the skier fails. The process is then repeated with a new question.

The Commodore 64 version is similar in operation.

## Typing In Ski Physics

If you have a VIC-20 with a tape drive, type in Program 1, SAVE, and VERIFY it. Then type NEW (or turn the computer off and then on again). Enter Program 2 and SAVE it immediately after Program 1. To use Ski Physics, RUN Program 1. After it has set up the custom characters, it will automatically load Program 2.

If you have a VIC and a disk drive, enter and SAVE both programs. Before the first program loads the second, you will be asked if you are using tape or disk.

Commodore 64 owners should simply type in Program 3, SAVE it, and RUN.

See program listings on page 157. @



# PROSYS

The Professional Systems People And

**MICRO WORX**

Present Products From

**commodore**

And

The Software That Makes Them Work!

## 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
1702 Monitor	249.00
1526 Printer	339.00
1600 Modem	69.00
80 Col. Card	59.00
MSD Dual & Single Drives	CALL

## SOFTWARE

**SBSYS** 8032, 8096 & B-Series  
THE SMALL BUSINESS SYSTEM  
Available for twin 8250 systems.  
GL, AP, AR, INV. & payroll  
modules all communicate.

**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.

**PERSYS - 64** ... for the  
'checkbook hacker or the double-entry,  
'serious-stuff' accounting buff.

**PERSYS PROFESSIONAL**  
... the affordable, communicating  
module accounting system for business.

Clear instructions that assume **you a dummy!** Computer assigned debits & credits ... too many reports to list ... either version turns little computers into veritable giants!

**PERSYS 64** Tape: \$69 Disk: \$79

**PERSYS PROFESSIONAL**  
General Ledger - as low as \$129  
AR, AP, INV & Payroll - as low as \$99 ea.

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 806/797-2623, Dallas/Ft. W. 817/589-2622.

Send \$1 for catalog, refundable on first order.  
VISA & MasterCard. Add 3% Surcharge.  
Prices for mail order only, subject to change without notice.

**MICRO**

**WORX**

4210 D 50th St.

Lubbock, TX 79413



# NEWS & PRODUCTS

## 64 Music System

---

With *Studio 64*, a music operating system developed by Entech for the Commodore 64, you can compose music and add your compositions to any other program.

The system operates like a music word processor. Notes are written just as they are played, and they scroll across the screen for easy reading. *Studio 64* requires no programming experience. And it teaches the relationship between how music looks and how it sounds.

Add *Mus'in*, the operating system package of *Studio 64*, adds your compositions to any other program. Simple commands are used. For example, the command "MUSIC" will start the music, "SONG" will start a particular song, "SPEED" will set the tempo, and "FEEL" will change the duration and tempo of the notes.

The *Studio 64* package is available on disk or cassette for \$39.95.

Entech Software  
P.O. Box 881  
Sun Valley, CA 91353  
(213) 768-6646

## SAT English And Math Aids

---

*SAT English I* and *SAT Math I*, Scholastic Aptitude Test preparation aids, have been introduced by Micro Lab for the Commodore 64.

The packages are a part of the company's Micro Learn line of educational products, and include test preparation questions and information designed to aid students who plan to take the SAT test.

The products cost \$30 each.

Micro Lab  
2699 Skokie Valley Road  
Highland Park, IL 60035  
(312) 433-7550

## Games And Word Processing Programs

---

Bröderbund Software has released a game for the VIC-20 and two games and a word processing program for the 64.

*Lode Runner* is a cartridge game for the VIC-20. In *Lode*

*Runner*, you are a galactic commando fighting your way through 24 different game screens, picking up chests of gold hidden in the Bungeling Empire's secret underground treasury. The VIC version also allows you to design screens.

The game has a suggested retail price of \$34.95 and requires a joystick.

*Operation Whirlwind* and *Matchboxes* are two new games for the Commodore 64. The first is a war game which incorporates a chesslike strategy approach as you move your battalion through a series of testing skirmishes and battle actions. *Whirlwind* is priced at \$39.95 on disk and comes with a battle map and user manual.

*Matchboxes* is a memory game in which the player attempts to match identical squares. It has additional modes in which players solve word puzzles and create their own puzzles. It is available for \$29.95 on disk.

*Bank Street Writer* is a word processing program now available for the Commodore 64 on disk. Functions and commands are displayed at the top of the screen so that you don't have to memorize function codes.

Features include universal



search and replace, block move and "unmove," automatic centering and indentation, word wrap, a print format routine, and other capabilities. *Bank Street Writer* comes with a tutorial on the disk that teaches word processing basics. A reference manual and back-up disk are included.

The price for *Bank Street Writer* is \$69.95.

Bröderbund Software, Inc.  
17 Paul Drive  
San Rafael, CA 94903  
(415) 479-1170

## Tax, Data, Writing Programs

Timeworks, Inc., has introduced several new packages for the Commodore 64, plus a cassette-to-disk conversion plan.

*Swiftax* is a menu-driven program that enables the user to prepare annual income tax returns without prior knowledge of computers or accounting. The program guides the user through the tax preparation process, gives instructions on which forms should be completed, and automatically checks tax alternatives such as income averaging.

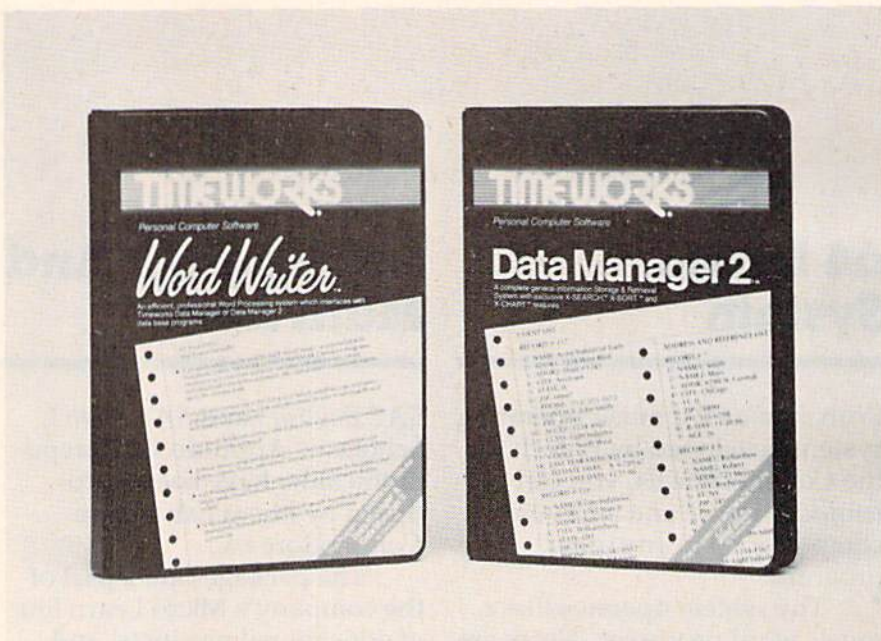
It prints tax information on each IRS schedule and form, and makes itemized statements for lists of dividends, interests,

and other categories that are too long for standard forms.

The suggested retail price for *Swiftax*, which is available on disk, is \$49.95.

*Word Writer* is a word processing package which can be used by itself or interfaced with Timeworks' *Data Manager* or new *Data Manager 2* programs.

The packages allow you to maintain and print out name and address lists, produce customized reports up to 20 columns wide, individually address and print form letters, print name and address files onto standard mailing labels, transfer and print text information onto labels and



Word Writer and Data Manager 2 from Timeworks.



# UNDERWARE

Software that's priced UNDER the competitions'

## COMMODORE 64

Just for You!

### PERSONAL FINANCE

Continental Home Accountant (D)	\$47
Continental Tax Advantage (D)	\$33
Continental FCM First Class Mail	\$29
Timeworks Electronic Checkbook (D&C)	\$19
Timeworks Money Manager (D&C)	\$19
Timeworks SwiftTax (D)	\$39
Creative Household Finance (D)	\$23
Creative Household Finance (C)	\$19
HesWare Finance Manager (D)	\$49
Cardco Tax Survival Program (D)	\$33

### WORD PROCESSING

Blue Sky Script 64 (D)	\$69
Cardco Write Now! (D)	\$39
Timeworks Word Writer (D)	\$39
On-Line Home Word (D)	\$39
Creative Joe's Writer (D)	\$42
HesWare OmniWriter & Spell (D)	\$49
Blue Sky Script 64 & Spell (D)	\$69
WordPro 3 Plus SpellRight (D)	\$69

### SPREADSHEETS

HesWare Multiplan (D)	\$75
HesWare Omnicalc (D)	\$37
MSI Practical (D&C)	\$35
MSI Programmable Spreadsheet (D)	\$55
Home Calc (D&C)	\$26
Creative Jack's Calc (D)	\$42
Handic Calc Result Advanced (D)	\$75

### DATABASES

Creative Fred's Filer (D)	\$42
Timeworks Data Manager (D&C)	\$19
Timeworks Data Manager II (D&C)	\$39
CodeWriter Home Filewriter (D)	\$55
Entech Data Base 64 (D)	\$55
MSI PractiFile (D)	\$39

### HOME APPLICATIONS

Spinnaker Aerobics (D)	\$33
CodeWriter Dialog (D)	\$45
CodeWriter ELF Easy Lang. Form (D)	\$45

### ART & MUSIC

Epyx Fun with Art (CT)	\$27
Epyx Fun with Music (CT)	\$27
HesWare Paintbrush (CT)	\$19
Spinnaker Delta Drawing (CT)	\$26
Delta Music (CT)	\$26
HesWare Synthesound (D)	\$19

### EDUCATION

SPINNAKER LEARNING FUN	
Alf Color Caves (ct) ages 3-6	\$26
Alphabet Zoo (ct) ages 3-8	\$23
Face Maker (ct) ages 3-8	\$23
KinderComp (ct) ages 3-8	\$19
Kids on Keys (ct) ages 3-9	\$23
Bubble Burst (ct) ages 4-8	\$26
Grandma's House (D) ages 4-8	\$23
Story Machine (ct) ages 5-9	\$26
Ranch West (ct) ages 5-10	\$26
Fraction Fever (ct) ages 7-12	\$23
JukeBox (ct) ages 8-adult	\$26
Aegean Voyage (ct) ages 8-adult	\$26
Up For Grabs (ct) ages 8-adult	\$26
Adventure Create (ct) ages 12 & up	\$26

### DESIGNWARE (DISKS)

Crypto Cube ages 8 & up	\$29
Spellcopter ages 6 & up	\$29
Crear. Creator ages 4-8	\$29
Math Maze ages 6-11	\$29
Spellakazam ages 7-14	\$29
SpellaGraph ages 7-14	\$29

### SCHOLASTIC WIZWARE (DISKS)

Turtle Tracks ages 9 & up	\$21
Bannercatch ages 9 & up	\$27
Double Adventure ages 9 & up	\$17
Double Mystery ages 9 & up	\$17
Secret Filer ages 10 & up	\$17
Logo Robot ages 10 & up	\$17
Square Pairs ages 7-12	\$17
HesWare Turtle Graphics II (ct)	\$39

## COMMODORE 64 Cont'd.

Creative I Am Your 64 I (D)	\$19
Creative I Am Your 64 II (D)	\$19
Scarborough Mastertype (D)	\$29
Scarborough SongWriter (D)	\$29
Scarborough Foresman Probe (D)	\$29

### LANGUAGES & UTILITIES

HesWare 6502 Pro Devel. Sys. (D)	\$19
HesWare Hesman 64 (CT)	\$26
HesWare 64 Forth (CT)	\$45
HesWare Graphics Basic (CT)	\$39
HesWare HES CAT (D)	\$19
HesWare HES FONT (CT)	\$16
HesWare HES KIT (CT)	\$39
Access Spriemaster (D&C)	\$23
Timeworks Programg. Kit I, II, III (D&C)ea	\$19
Blue Sky Last One (D)	\$79
Blue Sky 80 Column (D)	\$29
Blue Sky Graphic Designer (D)	\$29
Blue Sky 64 Statistics (D)	\$29
Blue Sky Super Basic (D)	\$29
Blue Sky Add On Basic (D)	\$29
Blue Sky Super Copy (D)	\$29

### BUSINESS SOFTWARE

Cymbal General Ledger (D)	\$45
Cymbal Accounts Receivable (D)	\$45
Cymbal Accounts Payable (D)	\$45
Cymbal Inventory Control (D)	\$45
Cymbal Invoice Writer (D)	\$45

### GAMES

Epyx Dragon Riders of Pern (D&C)	\$25
Epyx Silicon Warrior (CT)	\$25
Epyx Pitstop (CT)	\$27
Epyx Temple of Apshai (D&C)	\$27
Epyx Gateway to Apshai (CT)	\$27
Epyx Upper Reaches (D&C)	\$16
Epyx Jumpman Junior (CT)	\$27
Sublogic Pinball (D&C)	\$20
Broderbund Choplifter (CT)	\$27
Sublogic Flight Simulator	\$27
Sega Congo Bongo (CT)	\$25
Sega Star Trek (CT)	\$25
Sega Buck Rogers (CT)	\$25
Access Beach Head (D&C)	\$23
Avalon Hill TelenGuard (C)	\$15
Avalon Hill B-1 Bomber (C)	\$12
Creative Crisis Mountain (CT)	\$25
Synapse Fort Apocalypse (D&C)	\$22
Synapse Blue Max (D&C)	\$22
Synapse Zeppelin (D&C)	\$22
Synapse Sentinel (D&C)	\$22
Synapse ZAXXON (D&C)	\$28
ScreenPlay Pogo Joe (D&C)	\$19
ScreenPlay Asylum (D&C)	\$23
ScreenPlay K. Uston BlackJack (D)	\$57
Quick Silva Ring Power (D&C)	\$23
Quick Silva AquaPlane (D)	\$23
Parker Q*BERT (CT)	\$39
Parker Popeye (CT)	\$39
Parker Frogger (CT)	\$39
Parker Star Wars (CT)	\$39
Parker James Bond 007 (CT)	\$37
First Star Boulder Dash (D&C)	\$25
First Star Flip/Flop (D&C)	\$25

### ELECTRONIC ARTS

Pinball Construction Set (D)	\$40
M.U.L.E. (D)	\$40
Worms? (D)	\$35
Archon (D)	\$40
Hard Hat Mack (D)	\$35
Murder on the Zinderneuf (D)	\$40
The Tesseract Strategy (D)	\$40
Word Flyer (D)	\$35
The Cut & Paste Word Processor (D)	\$50
D-Bug (D)	\$35
Axis Assassin (D)	\$35
Music Construction Set (D)	\$40
Financial Cookbook (D)	\$50
Dr. J. & Larry Bird One on One (D)	\$40



HARDWARE



SOFTWARE



UNDERWARE

**FREE DISKETTE**  
with each purchase of electronic  
arts software

## COMMODORE VIC 20

### PERSONAL PRODUCTIVITY

Creative Home Office (D)	\$22
Creative Home Office (D)	\$19
Creative Household Finance (D)	\$17
Creative Household Finance (C)	\$13
MSI Practical Plus (D)	\$35
MSI Practical Plus (T)	\$33
MSI Practical (D)	\$32
MSI Practical (I)	\$29
HesWare Vic Fourth (CT)	\$39
HesWare Hes Man (CT)	\$26
Cardco Write Now	\$27
Epyx Fun with Art (CT)	\$26
Epyx Fun with Music (CT)	\$26

### 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
WizWare Turtle Tracks (C)	\$19
WizWare Square Pairs (C)	\$19

### GAMES

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

## PERIPHERALS

### GRAPHICS TABLET

CHALKBOARD POWERPAD	\$79
(C64 & Atari)	
Software Packages	
Logic Master	\$27
Leo's Links	\$27
Programmers Kit	\$19
BearJam	\$25
KOALAPAD ATARI C64	\$67

## PRINTERS

NEC 8023A	\$379
STAR GEMINI 10X	\$287
ALPHA COM 81	\$169
ALPHA COM 40	\$99
Cable with AlphaCom	FREE
BMC BX80	\$267
Okidata 82A	\$389
Okidata 83A	\$589
Transstar T315 Color	\$489

## MODEMS

Volkmodem 300 baud	\$79
HesModem	\$47
Anchor Mark 7 300 baud	\$129
Novation J-Cat 300 baud	\$119
Commodore 1600 Modem	\$58
Commodore 1650 Auto M.	\$155

## MONITORS

Amdel Color 1	\$229
BMC 13" Composite Color Plus	\$287
BMC 12" Low Res Green	\$85
BMC 12" Low Res Amber	\$99
BMC 12" Hi Res Green	\$119
BMC 12" Hi Res Amber	\$129
USI 12" Green Hi-Res	\$129
USI 12" Amber Hi-Res	\$139
USI 14" Composite Color	\$279
Monitor cable w/above	\$10

### COMMODORE DISK DRIVES

Concord Disk Drive	\$279
Commodore 1541	\$269

### WICO JOYSTICKS

The Boss	\$14
3 Way Grip Stick	\$21

### CARDCO

Numeric Keypad C64	\$37
Graphic Printer Interface	\$69
Economy Printer Interface	\$39
Commodore 64 5 Slot	\$49
16K Board	\$53
Cassette Interface	\$29

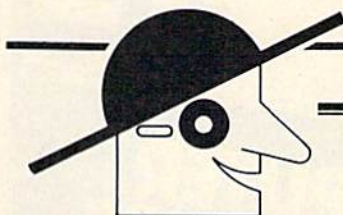
### BLANK DISKETTES

#### MAXELL DISKS

MD1 SS DD Box of Ten	\$27
MD2 DS DD Box of Ten	\$40

#### VERBATIM

ValuLife SS DD Box of Ten	\$28
ValuLife DS DD Box of Ten	\$42



**\$uch A Deal**

12629 N. Tatum Blvd.  
Suite 138  
Phoenix, AZ 85032

**CALL TOLL FREE 1-800-431-8697/602-957-3619**  
For Customer Service Call: 602-955-3857

ORDERING & TERMS: Send check, money order, personal/company checks allow 3 weeks bank clearance. VISA/MasterCard accepted. Provide phone number with order. SHIPPING: Software add \$4.00 for first piece, add \$1.00 each additional piece. Hardware add 3% or \$10.00 whichever is greater. Returns must have authorization number (call 602-958-9128 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. All prices in U.S. dollars. We pay shipping on backorders.

[www.commodore.ca](http://www.commodore.ca)



tags, and calculate numerical data from one column to the next.

The *Word Writer* has a full-screen format, with up to 80 characters per line. Two plastic keyboard overlays are included which place the word processing commands directly onto the keyboard. The program has full word processing features such as automatic search and replace, text block movement, automatic word wrap and page numbering, and other standard capabilities.

The *Word Writer* is priced at \$59.95 and is available on disk.

*Data Manager 2* is a menu-driven information storage package with cross-search features. The program allows you to break

down data in a variety of ways and to arrange data alphabetically, chronologically, or numerically. Automatic calculation of mathematical data allows you to perform payroll functions, cost estimates, and similar operations.

*Data Manager 2* is priced at \$49.95 on disk.

Timeworks is offering an option that will allow Commodore 64 users to convert Timeworks' programs on cassette tape to disk for a charge of \$4 (plus \$.70 postage and handling). Tapes that are returned must be in good playable condition and be an original program published by the company. Timeworks will then send back a disk version of

the program. The disk exchange is for the same program as the one being returned.

Timeworks, Inc.  
405 Lake Cook Road  
Deerfield, IL 60015  
(312) 291-9200

## VIC, 64 Accounting Package

TOTL Software, Inc., has introduced *TOTL.BUSINESS*, a set of accounting tools with integrated

## WAREHOUSE PRICES

Call us toll free for prices directly from our warehouse inventory. Save on our volume purchases. Largest selection of discounted hardware and software.

**800-372-0214**  
IN FLORIDA 800-432-0368

**SUPER SPECIAL**  
ELEPHANT DISK 10/Pack ..... \$16<sup>99</sup>

SOFTWARE	CALL
Flight Simulator II	\$24 <sup>99</sup>
Blue Max D or C	\$69 <sup>99</sup>
Home Accountant	\$64 <sup>99</sup> ea.
Atarisoft pacman/defender/centipede	\$69 <sup>99</sup>
Script-64	\$39 <sup>99</sup>
Practicalc	
HARDWARE	
13-inch Color Monitor	\$199 <sup>99</sup>
80 CPS Dot Matrix Printer	\$209 <sup>99</sup>
Letter Quality Printer	\$399 <sup>99</sup>
Gemini 10 X Printer	\$279 <sup>99</sup>
New Disk Drive for C64	CALL

Send \$1 for our complete catalog.

**COMPUTER  
WAREHOUSE**  
8764 S.W. 133 ST., MIAMI, FLORIDA 33156



and more



Add 3%

## FREE OFFER! FREE OFFER! COMPUTER CASSETTES

**58¢**

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

Specify VIC-20, Commodore 64, TI-99/4A, TRS-80 CoCo or MC-10

- C-10 Length/Free Labels
- 5 Screw Shell
- Storage Box add 12¢ each
- Lifetime money back guarantee
- \$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 Dept. V  
Blackwood, NJ 08012  
609-227-9634

## ARE YOU TIRED OF PAYING HIGH SOFTWARE PRICES FOR YOUR 64? JOIN THE CLUB!

You receive 120 programs, on disk or tape, each year (10 per month) including business, education, and games each month. You also receive software discounts, free computer consultation, plus much more. All this for \$40 per year or \$60 for 2 years. Send check or money order to: The 64 Club, 1260 Oliver, Pacific Beach, CA 92109.



## Overview

- 0 — Using CodePro-64
- 1 — CBM-64 Keyboard Review

## BASIC Tutorial

- 2 — Introduction to BASIC
- 3 — BASIC Commands
- 4 — BASIC Statements
- 5 — BASIC Functions

## Graphics &amp; Music

- 6 — Keyboard GRAPHICS
- 7 — Introduction to SPRITES
- 8 — SPRITE Generator
- 9 — SPRITE Demonstrator
- A — Introduction to MUSIC
- B — MUSIC Generator
- C — MUSIC Demonstrator

## Other Options

- K — Keyword Inquiry
- R — Run Sample Programs

Select Choice or hit F7 for Default

# A new concept in interactive visual learning.

## CodePro-64™

Now you can learn to code in BASIC and develop advanced programming skills with graphics, sprites and music—*visually*. You learn by interacting with CodePro-64, a new concept in interactive visual learning.

### SEE PROGRAM EXECUTION

Imagine actually seeing BASIC statements execute. CodePro-64 guides you through structured examples of BASIC program segments. You enter the requested data or let CodePro-64 do the typing for you. (It will not let you make a mistake.)

You step through and actually see the execution of sample program statements by simply pressing the space bar. CodePro-64 does the rest. You see statements with corresponding *graphics* and variable value displays.

### EXTENSIVE TUTORIAL

CodePro-64's extensive tutorial guides you through each BASIC command, program statement, and function. You get clear explanations. Where appropriate, you invoke BasicView to see examples execute and watch their flow charts and variables change.

By seeing graphic displays of program segment execution you learn by visual example. You learn faster and grasp programming concepts easier with CodePro-64 because you immediately see the results of your input.

You control your learning. You can go through the tutorial sequentially, or return to the main menu and select different topics, or use keywords to select language elements to study. You can page back and forth between screens within a topic at the touch of a function key.

Once you have practiced and mastered the BASIC language elements you move on to

more advanced concepts. You learn about sprite and music programming.

### SPRITE GENERATOR & DEMONSTRATOR

CodePro-64's sprite generator lets you define your own sprites on the screen. You learn how to define sprites and what data values correspond to your sprite definitions. (You can then save your sprite data to a diskette file for use in your own programs.) You can easily experiment with different definitions and make changes to immediately see the effects.

We also help you learn to program with sprites by giving you a *sprite demonstrator* so you can see the effect of changing register values. You can experiment by moving your sprite around in a screen segment, change its color and see the effects of your changes. You learn by visual examples.

### MUSIC GENERATOR & DEMONSTRATOR

Our Music Generator and Music Demonstrator will provide hours of instruction and creative enjoyment. From the beginning of your instruction you can compose simple tunes on the screen using the generator. Once you've completed a composition you can save the tune and its associated SID parameters to a diskette file. Our music sam-

ple program can be used alone or incorporated into your own programs to read the saved music file and replay your songs.

Our music demonstrator lets you experiment with various combinations of music programming parameters and hear the results. All you do is enter rows of SID parameters on the screen to create a particular sound. Then you hear each sound by playing the "keyboard organ" in real time as you shift from row to row of SID parameters. By seeing your input and hearing the result you quickly learn how to create new musical sounds and special sound effects.

Whether you're a beginning programmer or an experienced professional, CodePro-64 will help you improve your Commodore 64 programming skills. We're sure because CodePro-64 was developed by a team of two professionals with over 25 years of software development experience.

CodePro-64 is a professional quality educational program for the serious student of personal computing. And it's fully guaranteed. Order yours today.

### HOW TO ORDER

Order your copy of CodePro-64 today by mail or phone. Send only \$59.95 plus \$3.00 shipping and handling to:

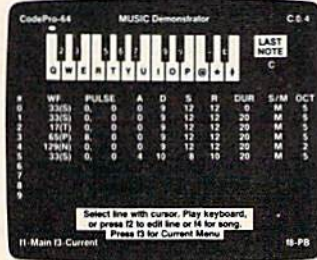
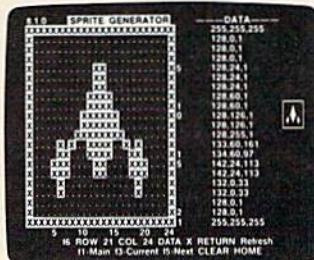
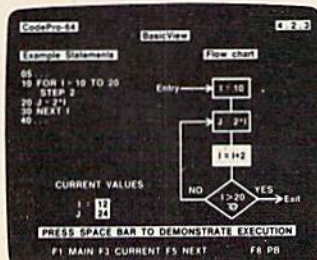
**SYSTEMS MANAGEMENT ASSOCIATES**  
3700 Computer Drive, Dept. G-1  
Raleigh, N.C. 27609

Available on diskette only. MasterCard/-VISA accepted. For faster service on credit card orders, call toll free 1-800 SMA-RUSH. (1-800-762-7874). Dealer inquiries invited.

Commodore 64 is a trademark of Commodore Business Machines, Inc.

### OUR GUARANTEE

We guarantee your satisfaction. You must be satisfied with CodePro-64 for the Commodore-64. Try it for 10 days and if for any reason you are not satisfied return it to us (undamaged) for a full refund. No risk.





programs for the VIC-20 and Commodore 64 computers. Functions and features are divided into two basic categories—accounts receivable and accounts payable.

The accounts receivable category includes invoice processing, customer statements, inventory control, and complete tracking of accounts receivable. The accounts payable category contains purchase order processing, check register and expense category tracking, and complete accounts payable.

*TOTL.BUSINESS* features a disk-based system and *CHICK-SPEED*, machine-language routines for fast loading and disk access. The programs are menu-driven and work with any 80-column printer.

Suggested price for *TOTL.-BUSINESS* is \$95 for the 64 and \$85 for the VIC-20. The VIC version requires a minimum of 24K expansion.

TOTL Software has also released *TOTL.SPELLER*, a companion to the company's word processing program, *TOTL.TEXT*. *TOTL.SPELLER* is a spelling checker with a built-in dictionary that checks and corrects the spelling of any document created with the *TOTL.TEXT* package.

Features include an automatic proofreading option which checks an entire document and flags possible errors, audible cues to facilitate unattended operation, compatibility with 80-column boards, disk utilities, and a fast-verify option which displays each suspect word for alteration or addition to the dictionary. The program has a starter dictionary of more than 10,000 words which can be expanded by the user to up to 24,000 words.

The price of *TOTL.SPELLER* is \$35 on disk.

*TOTL Software, Inc.*  
1555 Third Avenue  
Walnut Creek, CA 94596  
(415) 943-7877

## Word Processor For 64

Mirage Concepts, Inc., has introduced *Word Processor* for the Commodore 64 computer. Written in machine language, this word processing program allows 80-column screen display without additional hardware.

There are more than 70 available single-keystroke commands; printed page, line, and character counters; word wrap; search, replace, block, and column operations; and a 200-page documentation binder.

*Word Processor* is available for \$99.95.

*Mirage Concepts, Inc.*  
2519 W. Shaw Ave. #106  
Fresno, CA 93711  
(209) 227-8369

## Easier Memory For VIC-20

*16K Memory Plus* with battery backup is an expansion device for the VIC-20 which can retain data and programs for up to four weeks in the event of power loss.

Produced by Abaris, Inc., the memory device contains full block switching, reset switch, built-in edgcard receptacle, and

write-protect switch. The battery backup retains the programs. Once a program is loaded into memory, the *16K Memory Plus* can be removed and plugged into another VIC-20. Custom routines can be retained in block five memory and will be executed when the computer is turned on.

The price for the *16K Memory Plus* is \$89, plus a \$3 shipping and handling charge.

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

## Educational Game For VIC

PMI, Inc., has introduced a new educational game for the VIC-20 on tape or disk called *Animated Arithmetic* for young children and pupils in remedial classes.

The game teaches addition using color graphics. Users don't need to be able to read. Included in the package are two programs, *Clown Count* and *Addition Express*.

*Clown Count* teaches children to count by asking them to add the number of boxes the clown is holding. If the answer is correct, the clown juggles the boxes. *Addition Express* teaches addition using express trains traveling along number lines. If the answer is correct, the engine puffs out smoke.

The price for *Animated Arithmetic* is \$10.95.

*PMI, Inc.*  
P.O. Box 87  
Buckfield, ME 04220  
(207) 336-2082 (800) 227-1836



[www.commodore.ca](http://www.commodore.ca)





CodePro-64 is a BASIC tutorial with sprite and music programming instruction for the Commodore 64.

## Commodore 64 Tutorial Package

Systems Management Associates has introduced *CodePro-64*, an integrated software tutorial package for the Commodore 64.

*CodePro-64* combines 12 instructional topics into one menu-driven system of 16 pro-

grams. This BASIC tutorial features instruction on commands, statements, and functions with visual examples. It covers the programming of sprites and music, and provides demonstration programs.

The tutorial is available on two disks, for \$59.95, and includes a 140-page reference manual.

*Systems Management Associates*  
3700 Computer Drive  
Raleigh, NC 27609  
(919) 787-7703

## 64 Spelling Tutorial

*The Spelling Teacher* is an educational spelling tutorial program for the Commodore 64 with disk drive.

The program contains four word files at different levels with 100 words per file. Twenty-five words are presented during each spelling session. Words that are misspelled reappear automatically during the next session.

Parents have the option of adding or deleting words. A graphics bar chart displays the last ten scores, and there is one bar chart for each of the word files.

*The Spelling Teacher* is available for \$39.95.

*Computer Technology Company*  
Computech  
P.O. Box 7000-309  
Redondo Beach, CA 90277  
(213) 375-6391



## Pro Football On Tape

CMS Software has introduced *Professional Football* on cassette tape for the VIC-20 with 16K or for the Commodore 64.

This strategy game pits the player against the offensive or defensive play-calling of the computer. There are 11 offensive play selections (plus punts and field goals), 11 defensive alignments, full-feature scoreboard, sudden-death overtime for tied games, and a user manual.

The computer selects its own offensive and defensive plays based on down, distance, field position, time remaining, and score. Full statistics are given at half-time and at the end of the game.

The game is available on tape for \$16.95 (postage paid).

CMS Software  
P.O. Box 4876  
Topeka, KS 66604-0876  
(913) 267-5864

## Games For Commodore 64

Synapse Software has released six of its games for the Commodore 64 computer.

*Necromancer* is an action game in which you have the power to liberate humanity in a world where evil reigns supreme. *Shamus Case II* is a detec-

tive adventure game involving 38 rooms to be searched.

*Pharaoh's Curse*, a game with three levels of play and 16 screens, pits you against an evil mummy and the ghost of Rana.


In *Zeppelin*, you maneuver an airship through 250 rooms of an underground cavern. *Picnic Paranoia* is played in a series of 90-second rounds in which you kill invading ants while keeping your food from being pushed off the screen.

*Blue Max* uses a 3-D diagonal scrolling effect to put you in the open cockpit of a British World War I biplane. After bombing enemy targets, you must land, refuel, and prepare to take off again.

All of the games are available on tape and on disk for \$34.95 each.

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

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. 

## Software Discounters of America



1-800-225-SOFT

P.O. Box 278-Wildwood, PA 15091

In PA (412) 361-5291

### Commodore 64

<b>Access</b>	
Beach Head (T or D).....	\$23
Neutral Zone (T or D).....	\$23
<b>Artworx</b>	
Bridge 4.0 (T or D).....	\$16
Monkey Math (T or D).....	\$18
Strip Poker (T or D).....	\$21
<b>Batteries Included</b>	
Paperclip (D).....	\$59
Paperclip w/Spelling (D).....	\$79
<b>Broderbund</b>	
Bank St. Writer (D).....	\$43
Choplifter (D or CT).....	\$24
Loderunner (D).....	\$21
Mask of the Sun (D).....	\$25
<b>Continental</b>	
Comm 64 User's Encyclopedia.....	\$13
Home Accountant.....	\$47
<b>Epyx</b>	
Gateway to Apshai (CT).....	\$25
Jumpman (T or D).....	\$25
Mission Impossible (D).....	\$25
Pitstop (CT).....	\$25
Summer Games (D).....	\$25
Temple of Apshai (T or D).....	\$25
<b>HES</b>	
Omnicalc (D).....	\$30
Multiplan (D).....	\$65
<b>Infocom</b>	
Deadline (D).....	\$33
Planetfall (D).....	\$33
Sorcerer (D).....	\$33
<b>Parker Bros.</b>	
Frogger (CT).....	\$33
Popeye (CT).....	\$33
Q-Bert (CT).....	\$33
<b>Professional Software</b>	
Word Pro 3 +/Spell Right + (D).....	\$65
<b>Quick Brown Fox</b>	
QBF (CT).....	\$37
Quickspell (D).....	\$21
Expand-A-Fox 80 (D).....	\$23
<b>Sega</b>	
Buck Rogers (CT).....	\$25
Congo Bongo (CT).....	\$25
Star Trek (CT).....	\$25
<b>Spinnaker</b>	
Alphabet Zoo (CT).....	\$21
Juke Box (CT).....	\$25
Kidwriter (D).....	\$21
Story Machine (CT).....	\$21
<b>Synapse</b>	
Blue Max (T or D).....	\$21
Necromancer (T or D).....	\$21
Pharaoh's Curse (T or D).....	\$21
Zeppelin (T or D).....	\$21
<b>Timeworks</b>	
Data Manager (T or D).....	\$17
Money Manager (T or D).....	\$17
Programming Kit 1,2 or 3 (T or D).....	\$17
<b>Accessories</b>	
Alien Group Voice Box.....	\$89
BASF SS, DD.....	\$17 Bx
Cardco Accessories.....	Call
Compuserve Starter Kit.....	\$25
HES Modem I.....	\$49
Koala Pad.....	\$67
Wico Boss.....	\$13
Wico Trackball.....	\$29

**Ordering & Terms:** Orders with cashier check or money order shipped immediately. Personal/Co. checks allow 3 weeks clearance. VISA/MASTERCARD accepted with no additional charge. **Shipping:** Orders under \$100 add \$3.00; free shipping on orders over \$100. **PA. residents add 6% sales tax. Returns:** Defective merchandise will be replaced with same merchandise—no credits! Returns must have authorization number (412) 361-5291. Prices subject to change without notice.



# Bug-Swatter:

## Modifications And Corrections

• The VIC version of "Trenchfire" (March) contains incorrect instructions. Trenchfire was written to run on an unexpanded VIC, but must be entered using Tiny MLX (also in that issue, p. 165) and an 8K (or more) expander. When expansion memory is added, the VIC automatically moves screen memory and start-of-BASIC. The instructions in step one (POKE44,24:POKE24\*256,0:NEW) will move start-of-BASIC, but leave screen memory at 4096-4608. As a result, the machine language at 4352-4608 can become garbled by the screen (at the same locations). The correct instructions should be:

Insert the 8K expander, turn on your VIC and enter this line:

```
POKE648,24:SYS58648:POKE642,26:SYS58232
```

Readers who have already typed in the VIC version of Trenchfire do not have to retype the whole program to fix it. To create a patch, follow the instructions above, LOAD Tiny MLX and use a starting address (S) of 4352 and ending address (E) of 4610. Enter the first 43 lines and SAVE to tape or disk. To RUN the game, turn off your VIC, unplug the memory expansion, and turn it back on. Tape users can now LOAD "TRENCHFIRE",1,1 followed by LOAD "PATCH",1,1. If you have a disk drive, substitute ,8,1. You can then start the game with SYS 4352.

• The VIC version of "Typing Derby" (February) fits very snugly into memory, with only a dozen bytes to spare. Readers who encounter an OUT OF MEMORY error should remove any extraneous spaces. Also, the process of crunching the program into memory required the removal of certain lines. Line 100 targets such a line and should be changed to 100 IFPEEK(H1+M+1)<>32THEN16.

• "Space Duel" (December) runs as listed, but gives an unfair advantage to player 2, who gets ten points per hit. Player 1 receives only nine points. To temporarily fix the program, POKE 49664,11 after you LOAD Space Duel, but before you SYS 49152. Use MLX's Load, New Address, and Save commands to make a permanent correction. The correction in MLX format is: 49662 003, 105, 011, 141, 078, 003, 083.

• Due to the mechanics of the INTeger function, the final program in "Printing Tables" (page

127, March) will add -1 to negative numbers. Thanks to readers Evie Matheus and H. Flaschka for discovering our mistake. INT rounds a number down to the nearest integer. INT(-77.22) gives a result of -78. To fix this, use ABS and SGN in lines 22-24.

```
22 I$=LEFT$(STR$(SGN(I)),1)+STR$(INT(ABS(I)))+"."+RIGHT$(STR$(I*100),2)
24 J$=LEFT$(STR$(SGN(I)),1)+STR$(INT(ABS(I)))+"."+RIGHT$(STR$(I*100),2)
```

A simpler method is to add 1 to all negative numbers, to offset the 1 that is subtracted.

```
21 IF I<0 THEN I=I+1
23 IF J<0 THEN J=J+1
```

The disadvantage to this simpler method is that you cannot use the negative numbers in later calculations because they will be off by 1.

• The 64 version of "Speed Reader" (February) contains a typographical error. Line 640 should be DATA "TO CORRECTING OUR BAD HABITS, BUT THE".

• As noted in February Bugswatter, before entering "64 BASIC Aid" (January), you must POKE52,154: POKE56,154: CLR before you load or run MLX. In addition, MLX will not allow an ending address higher than 40960; use 40959 instead of 40961.

Finally, because it is a machine language program, it must be LOADED to the section of memory it was written for. To do this, use a secondary address of 1 (LOAD "BASIC AID",8,1 for disk, LOAD "BASIC AID",1,1 for tape). After the program is loaded, type NEW (to reset the BASIC pointers) and SYS39852 to begin the program.

• Some readers have reported ?SYNTAX ERRORS in lines 770 and 860 of MLX (January, February, and March). These lines are fine with the spaces properly inserted (ST AND191). Without the spaces, STAND is interpreted as the TANGent function. ☹



**TUSSEY MOUNTAIN SOFTWARE SAYS:**  
**COMPARE OUR PRICES ON C64 SOFTWARE!**



Wordpro 3+/64 w/Spellright(d) .....	\$64.	Practicalc 64 by MSI(d) .....	\$37.
Script 64(d) .....	\$64.	Practicalc 64 by MSI(cass) .....	\$35.
Calc Result Adv. (cart,d) .....	\$99.	Cardco Printer Int. w/Graphics .....	\$67.
Calc Result Easy (cart) .....	\$57.	Cardco Printer Int./A. ....	\$59.
Hesware Multiplan(d) .....	\$67.	Cardco Numeric Keypad .....	\$29.
Hesware Omnicalc(d) .....	\$34.	Cardco Lightpen .....	\$29.
HesModem I (64 or 20) .....	\$54.	The Last One(d) .....	\$69.
P.S. (Progr. Spreadsheet) (d) .....	\$55.	Delphi's Oracle(d) .....	\$106.

**TO ORDER:** For MasterCard, Visa, COD orders call 814-234-2236. Hours are 10-7 mon-sat, 12-5 sun EST. For credit card orders by mail send Card numb., exp. date. For Personal checks, certified checks, money orders write: **Tussey Mountain Software, Box 1006, State College, PA, 16804. MOST ORDERS SHIPPED IN 48 HOURS!** Credit Card orders add \$3.00 shipping. COD add \$4.50. Others shipped free. Allow 2 weeks for personal checks to clear. Orders shipped UPS unless specified otherwise. Manufacturer's warranty honored with our invoice and original packaging. PA residents add 6% sales tax.



# 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...":FOR I=886 TO
1018:READ A:CK=CK+A:POKE I,A:NEXT
110 IF CK<>17539 THEN PRINT "{DOWN}YOU MADE
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
```



# 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.

## Braces And Special Characters

The exception to this typing rule is when you see the braces, such as "{DOWN}". Anything within a set of braces 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 braces 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 words within braces 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, [x], you should hold down the Commodore key while pressing the key inside the special brackets. (The Commodore key is the key in the lower left corner of the keyboard.) Again, if the key is preceded by a number, you should press the key as many times as necessary.

Rarely, you'll see a solitary letter of the alphabet enclosed in braces. These characters can be entered on the Commodore 64 by holding down

the CTRL key while typing the letter in the braces. For example, {A} would indicate that you should press CTRL-A. You should never have to enter such a character on the VIC-20, but if you do, you would have to leave the quote mode (press RETURN and cursor back up to the position where the control character should go), press CTRL-9 (RVS ON), the letter in braces, and then CTRL-0 (RVS OFF).

About the *quote mode*: You know that you can move the cursor around the screen with the CRSR keys. Sometimes a programmer will want to move the cursor under program control. That's why you see all the {LEFT}'s, {HOME}'s, and {BLU}'s in our programs. The only way the computer can tell the difference between direct and programmed cursor control is the quote mode.

Once you press the quote (the double quote, SHIFT-2), you are in the quote mode. If you type something and then try to change it by moving the cursor left, you'll only get a bunch of reverse-video lines. These are the symbols for cursor left. The only editing key that isn't programmable is the DEL key; you can still use DEL to back up and edit the line. Once you type another quote, you are out of quote mode.

You also go into quote mode when you INSERT spaces into a line. In any case, the easiest way to get out of quote mode is to just press RETURN. You'll then be out of quote mode and you can cursor up to the mistyped line and fix it.

Use the following table when entering cursor and color control keys:


When You Read:	Press:	See:	When You Read:	Press:	See:	When You Read:	Press:	See:
{CLR}	SHIFT CLR/HOME		{CYN}	CTRL 4		{7}	CTRL 7	
{HOME}	CLR/HOME		{PUR}	CTRL 5		{8}	CTRL 8	
{UP}	SHIFT ↑ CRSR		{GRN}	CTRL 6		{F1}	F1	
{DOWN}	↓ CRSR		{BLU}	CTRL 7		{F2}	SHIFT F2	
{LEFT}	SHIFT ← CRSR		{YEL}	CTRL 8		{F3}	F3	
{RIGHT}	→ CRSR		{1}	CTRL 1		{F4}	SHIFT F4	
{RVS}	CTRL 9		{2}	CTRL 2		{F5}	F5	
{OFF}	CTRL 0		{3}	CTRL 3		{F6}	SHIFT F6	
{BLK}	CTRL 1		{4}	CTRL 4		{F7}	F7	
{WHT}	CTRL 2		{5}	CTRL 5		{F8}	SHIFT F8	
{RED}	CTRL 3		{6}	CTRL 6				



You've been searching  and searching  for our

new 

number to find **EDUCATIONAL SOFTWARE** for your

**COMMODORE 64™** and  **VIC 20™**

You've been told our **TOUCH TYPING TUTOR** was the #1 VIC 20 Educational best seller in 1983. (We thank our customers.) —*Compute's Gazette!*

You've read our **TOUCH TYPING TUTOR** reviews — "For the serious student, this is the best, most pedagogically sound program available." —*Midnite Software Gazette*

Now your search has ended.

## TOUCH TYPING TUTOR

TTT64D .....diskette US \$24.95

TTT64 .....cassette US \$19.95

TTT5K VIC 20 .....cassette US \$19.95

## TAYLORMADE SOFTWARE



P.O. Box 5574  
Lincoln, NE 68505  
(402) 488-0196



US/Canada shipping \$1.50

Foreign US\$ 4.00

Dealer inquiries welcome

Commodore 64 & VIC 20 are 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**

- 5. The Reset Switch**... Here it is, a Reset Switch for Vic or Commodore. Get back into control of a "Hung-Up" program. Resets all pointers. Easy two solder connection installation. Every computer should have one.

**\$9.95**

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.

### 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	\$ _____
_____	The Reset Switch @ \$9.95	\$ _____
_____	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 \_\_\_\_\_

**Bytes & Pieces, Inc.**

Dealer  
Inquiries  
Invited.

550 N. 68th Street,  
Wauwatosa, WI 53213 414/257-3562



# SpeedScript Revisited

(Article on page 38.)

## BEFORE TYPING...

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

## File Converter

```
100 PRINT"[CLR]{RVS}{N}{2 SPACES}SPEEDSCR
IPT FILE CONVERSION PROGRAM{3 SPACES}
" :rem 25
110 GOSUB410 :rem 167
120 INPUT"[DOWN]INPUT FILE NAME";I$ :rem 113
130 IFI$=""THEN120 :rem 211
140 INPUT"[DOWN]OUTPUT FILE NAME";O$ :rem 218
150 PRINT"[DOWN]{RVS}D{OFF}ISK, {RVS}S
{OFF}CREEN, {RVS}P{OFF}RINTER, {RVS}O
{OFF}THER" :rem 29
160 GETA$:IFA$=""THEN160 :rem 81
170 DV=-(A$="T")-3*(A$="S")-4*(A$="P")-8*
(A$="D"):SA=7 :rem 153
180 IFDV=0THENINPUT"DEVICE NUMBER";DV:INP
UT"SECONDARY ADDRESS";SA :rem 11
190 PRINT"[2 DOWN]WHICH CONVERSION:" :rem 192
200 PRINT"[DOWN]1) SPEEDSCRIPT TO COMMODO
RE ASCII" :rem 197
210 PRINT"[DOWN]2) SPEEDSCRIPT TO TRUE AS
CII" :rem 98
220 PRINT"[DOWN]3) COMMODORE ASCII TO SPE
EDSCRIPT" :rem 201
230 GETP$:IFP$<"1"ORP$>"3"THEN230:rem 101
240 ADR=828+VAL(P$)*3-3 :rem 220
250 OPEN15,8,15,"I0":REM REMOVE,"I0" IF
{SPACE}YOU'VE CHANGED THE DRIVE'S SPE
ED :rem 97
260 OPEN1,8,3,I$:INPUT#15,EN,EM$:F$=I$:IF
EN=0THEN290 :rem 44
270 PRINT"[DOWN]DISK ERROR FOR ";F$:PRINT
EM$ :rem 185
280 PRINT"[3 DOWN]RUN{3 UP}":CLOSE1:CLOSE
2:CLOSE15:END :rem 48
290 IFDV<>8THENOPEN2,DV,SA,O$:GOTO380
:rem 60
300 EX$="S,W":IFP$="3"THENEX$="P,W" :rem 56
310 OPEN2,DV,SA,"0:"+O$+EX$:INPUT#15,EN,E
M$:F$=O$ :rem 42
320 IFEN=0THEN380 :rem 238
330 IFEN<>63THEN270 :rem 99
340 IFEN=63THENPRINT"[DOWN]";O$;" EXISTS.
.. REPLACE? {RVS}Y{OFF}/{RVS}N{OFF}:" :rem 26
350 GETA$:IFA$<>"Y"ANDAS$<>"N"THEN350 :rem 45
360 IFA$="N"THEN270 :rem 36
370 PRINT#15,"S0:"+O$:CLOSE2:GOTO310 :rem 100
380 SYS(ADR):IF(PEEK(144)AND64)THENPRINT"
{DOWN}DONE.":GOTO280 :rem 26
390 PRINT"I/O ERROR DURING CONVERSION.":I
NPUT#15,EN,EM$:IFEN<>0THEN270:rem 253
400 GOTO280 :rem 103
```

```
410 FORI=828TO1001:READA:POKEI,A:CK=CK+A:
NEXT:IFCK=21584THENRETURN :rem 222
420 PRINT"[RVS]ERROR IN DATA STATEMENTS." :rem 251
:END
430 DATA 076,069,003,076,122,003 :rem 33
440 DATA 076,174,003,032,225,255 :rem 36
450 DATA 240,018,032,216,003,032 :rem 20
460 DATA 095,003,032,183,255,072 :rem 39
470 DATA 032,224,003,104,041,064 :rem 21
480 DATA 240,233,076,204,255,133 :rem 38
490 DATA 251,041,064,010,005,251 :rem 24
500 DATA 041,191,133,251,041,032 :rem 20
510 DATA 073,032,010,005,251,201 :rem 12
520 DATA 095,208,002,169,013,133 :rem 34
530 DATA 251,096,032,225,255,240 :rem 37
540 DATA 221,032,216,003,032,095 :rem 24
550 DATA 003,041,127,201,065,144 :rem 25
560 DATA 018,201,091,176,014,170 :rem 34
570 DATA 165,251,041,128,073,128 :rem 43
580 DATA 074,074,133,251,138,005 :rem 41
590 DATA 251,133,251,032,183,255 :rem 40
600 DATA 072,032,224,003,104,041 :rem 15
610 DATA 064,240,207,076,204,255 :rem 37
620 DATA 032,225,255,240,169,032 :rem 35
630 DATA 216,003,201,013,208,002 :rem 14
640 DATA 169,031,072,041,128,074 :rem 40
650 DATA 133,251,104,041,063,005 :rem 24
660 DATA 251,133,251,032,183,255 :rem 38
670 DATA 072,032,224,003,104,041 :rem 22
680 DATA 064,240,217,076,204,255 :rem 45
690 DATA 162,001,032,198,255,076 :rem 47
700 DATA 207,255,162,002,032,201 :rem 21
710 DATA 255,165,251,076,210,255 :rem 42
```

## Props

(Article on page 50.)

```
1 GOSUB1000:GOSUB80:GOSUB50:SYS49152 :rem 241
2 POKEW1,17:SYS49408:C=PEEK(CX):IFCTHEND=
PEEK(SP+A(C)):GOSUB10 :rem 122
3 IFPEEK(HM)THENGOSUB20 :rem 222
4 FORJ=RTO20-DL:NEXT:BZ=BZ+R:IFBZ=MXTHENB
Z=.:GOSUB50 :rem 247
5 IFSK>NTHENGOSUB7 :rem 212
6 POKEH1,(PEEK(251)/5+9):POKEW1,16:GOTO2 :rem 90
7 KL=KL+R:IFKL>NTHENKL=.:POKEFAST,INT(RND
(R)*M):POKE845,(.) :rem 238
8 IFRND(R)>PTHENPOKEFAST,V:POKE845,(.) :rem 159
9 RETURN :rem 25
10 POKEBD,2:FORJ=RTO40:POKESP+A(C),D:POKE
W1,129:SYS49608:POKEW1,128:NEXT:rem 31
11 SC=SC-SK:IFSC<RTHENSCL=.:rem 179
12 POKEBD,.:POKECX,.:RETURN :rem 234
13 REM LINES 7-9 = 'VARY SPEED' RTN :rem 146
14 REM LINES 10-12 = 'HIT PROP' RTN :rem 90
20 REM*** 'HOME' :rem 59
21 SC=SC+SK+3:IFSC>199THENGOSUB70:GOSUB80 :rem 107
22 POKESP+21,0:PRINTSC$[6]SCORE:"SC :rem 202
23 FORJ=100TO1STEP-2:SYS49608:POKEW1,21:P
OKEH1,J:POKEW1,20:NEXT :rem 231
24 PRINTSC$[13 SPACES]:POKEHM,0:POKECX,
0 :rem 171
50 REM*** 'NEW COOP' :rem 48
```



```

51 POKESP+21,0:POKEW3,21:POKEH3,PEEK(251)
   :PRINTSC$"{YEL}NEW COOP":PRINT"{HOME}
   [6]";
   :rem 50
52 FORJ=1TO24:PRINT"{RIGHT}Z{36 RIGHT}Z":
   NEXT
   :rem 120
53 PRINT"{RIGHT}Z{36 RIGHT}Z{HOME}":POKEW
   1,16:POKEW3,20
   :rem 54
54 IFHF=0THENHF=1:GOTO58
   :rem 254
55 HF=0:J=1062:FS=40*(INT(RND(1)*25))
   :rem 17
56 HI=INT((FS+J)/256):LO=(FS+J)-(HI*256):
   POKE843,LO:POKE844,HI
   :rem 124
57 SYS49615:POKEJ+FS+L1,10:PRINTSC$
   "{10 SPACES}":POKESP+21,255:GOSUB100:RE
   TURN
   :rem 172
58 J=1025:FS=40*(INT(RND(1)*25))
   :rem 222
59 HI=INT((FS+J)/256):LO=(FS+J)-(HI*256):
   POKE843,LO:POKE844,HI
   :rem 127
60 SYS49615:POKEJ+FS+L1,10:PRINTSC$
   "{10 SPACES}":POKESP+21,255:GOSUB100:RE
   TURN
   :rem 166
70 REM** NEXT LEVEL
   :rem 86
71 PRINTCHR$(142):FORJ=1TO500:NEXT:PRINTC
   HR$(14):POKESP+21,0:POKEW2,21
   :rem 93
72 POKEW1,20:POKEFV,3 :FORK=5TO115STEP2:P
   OKEW3,21:POKEBD,K:POKE646,K
   :rem 0
73 POKEH1,K*1.4:PRINTSC$"NEXT LEVEL?":POK
   EH3,K/4:SYS49608:POKEW3,20:NEXT
   :rem 253
74 POKEW1,20:POKEFV,66:FORJ=1TO3500:NEXT:
   POKEH3,100:POKEBD,0:POKEW2,129:RETURN
   :rem 202
80 REM** PICK SKILL LEVEL
   :rem 232
81 POKESP+21,0:POKEFNA(0),32:KZ=0:rem 193
82 PRINTSC$"{YEL}{3 LEFT}PICK SKILL LEVEL
   "
   :rem 136
83 PRINTSC$"[8]{2 DOWN}{RIGHT}(1 - 6)"
   :rem 181
84 GOSUB200
   :rem 126
85 IFKZ<>1THENSYS49608:GOTO84
   :rem 100
86 PRINTSC$"{3 LEFT}{16 SPACES}"
   :rem 208
87 PRINTSC$"{2 DOWN}{11 SPACES}"
   :rem 28
88 SC=0:RETURN
   :rem 141
100 REM** PAUSE ROUTINE
   :rem 151
101 IFFNB(.)THENRETURN
   :rem 133
102 POKEBD,13:PRINTSC$"{RVS}{YEL} PAUSING
   {OFF}"SC$"[8]{2 DOWN}SCORE:"SC
   :rem 253
103 PRINTSC$"{4 DOWN}FIRE=PLAY"SC$
   "{6 DOWN}ZERO=QUIT"
   :rem 32
104 FORJ=1TO1500:NEXT:POKEBD,0
   :rem 97
105 SYS49608:S=S+1:IFINT(S/43)=S/43THENPO
   KEW3,20:POKEH3,PEEK(SP+1):POKEW3,21
   :rem 170
106 IFFNB(.).THEN110
   :rem 167
107 GOSUB200
   :rem 170
108 IFSK>5THENGOSUB7
   :rem 31
109 GOTO105
   :rem 105
110 PRINTSC$"{12 SPACES}"SC$"{2 DOWN}
   {10 SPACES}"SC$"{4 DOWN}{9 SPACES}"
   :rem 127
111 PRINTSC$"{6 DOWN}{9 SPACES}"
   :rem 132
112 POKEW3,20:POKEBD,2:POKEBD,5:POKESP+31
   ,0:POKEBD,7:POKEBD,3:POKEBD,0:RETURN
   :rem 116
200 REM** SKILL
   :rem 115
201 G=PEEK(197)
   :rem 46
203 IFG=35THEN300
   :rem 210
204 IFG=56THENSK=1:DL=1:POKEFAST,1:POKERO
   T,4:POKE829,0:POKE845,0:KZ=1:RETURN
   :rem 110
205 IFG=59THENSK=2:DL=5:POKEFAST,2:POKERO
   T,3:POKE829,0:POKE845,0:KZ=1:RETURN
   :rem 119
206 IFG=8THENSK=3:DL=10:POKEFAST,3:POKERO
   T,2:POKE829,0:POKE845,0:KZ=1:RETURN
   :rem 111
207 IFG=11THENSK=4:DL=15:POKEFAST,4:POKERO
   T,1:POKE829,0:POKE845,0:KZ=1:RETURN
   :rem 160
208 IFG=16THENSK=5:DL=19:POKEFAST,5:POKERO
   T,0:POKE829,0:POKE845,0:KZ=1:RETURN
   :rem 171
209 IFG=19THENSK=6:DL=19:POKEFAST,6:POKERO
   T,0:POKE829,0:POKE845,0:KZ=1:RETURN
   :rem 177
210 RETURN
   :rem 115
300 REM** QUIT
   :rem 56
301 POKE49221,2:POKE49228,0:POKE829,0:POK
   E845,0:POKE646,7
   :rem 43
302 FORJ=SP+1TOSP+15STEP2:POKEJ,123:NEXT:
   POKE214,24:PRINT:FORJ=1TO24
   :rem 112
303 PRINT:FORK=1TO20:NEXTK,J:POKEW3,20
   :rem 189
304 POKEW3,21:PRINTSC$"BYE BYE . . .":FOR
   J=1TO2400:NEXT:POKESP+21,0
   :rem 159
305 FORJ=L1TOVM+3:POKEJ,0:NEXT:SYS64738
   :rem 244
1000 REM** INITIALIZE
   :rem 21
1001 REM**ML RTNS: SPRITEMOVE=49152:BIRDM
   OVE=49408:FLAPWING=49608:PUTMATE=496
   15
   :rem 182
1002 POKE54296,47:POKE54295,66 :POKE53272
   ,21:POKE53281,0:BD=53280:POKEBD,0
   :rem 170
1004 GOSUB10000:GOSUB12000:FAST=49221:ROT
   =49228:R=1:N=5:M=7:P=(.98):V=28:MX=2
   00
   :rem 138
1006 REM** SET UP SPRITES
   :rem 228
1008 POKE53275,255:SP=53248:CX=SP+31:POKE
   CX,0
   :rem 138
1012 FORJ=2040TO2047:POKEJ,221:NEXT
   :rem 85
1014 B=80:FORJ=SP+1TOSP+15STEP2:POKEJ,B:B
   =B+20:NEXT
   :rem 187
1016 POKESP+23,0:POKESP+29,24
   :rem 186
1018 POKESP,40:POKESP+2,70:POKESP+4,100:P
   OKESP+6,130:POKESP+8,188:POKESP+10,2
   45
   :rem 24
1020 POKESP+12,20:POKESP+14,48:POKESP+16,
   192
   :rem 75
1022 POKESP+39,7:POKESP+40,3:POKESP+41,4:
   POKESP+42,13
   :rem 180
1024 POKESP+43,7:POKESP+44,3:POKESP+45,13
   :POKESP+46,4
   :rem 189
1025 REM ML VARBLs, ETC IN CASET BUFR
   :rem 77
1026 POKE828,221:REM START SPRITE PNTRS
   :rem 117
1028 POKE829,1:REM SPRITE ROTATE COUNTR
   :rem 153
1030 POKE830,40:POKE831,0:REM LINE VAL
   :rem 168
1032 POKE832,0:REM R/L JYSTK FLAG:rem 136
1034 POKE833,0:REM U/D JYSTK FLAG:rem 134
1036 POKE834,91:REM BIRD CHAR VAL:rem 110
1038 POKE835,1:POKE836,4:REM UP LIMIT
   :rem 164
1040 POKE837,230:POKE838,6:REM DN LIMIT
   :rem 244
1042 POKE841,1:REM SPRITE MOVE COUNTR
   :rem 247

```



```

1044 HM=842:POKEHM,0:REM 'HOME' FLAG
                                :rem 66
1046 POKE843,95:POKE844,5:REM MATE'S FIRS
T LOCATION                      :rem 61
1048 POKE251,144:POKE252,4:REM FIRST BIRD
LOCATION - ZERO PAGE             :rem 101
1049 POKE845,0:REM FILTER CUT COUNT
                                :rem 165
1050 DIMA(129):A(1)=1:A(2)=3:A(4)=5:A(8)=
7:A(16)=9:A(32)=11:A(64)=13:A(128)=1
5                                :rem 235
1052 HF=0:BZ=0                  :rem 6
1054 DEFFNA(X)=((PEEK(252)*256)+PEEK(251)
)                                :rem 5
1056 DEFFNB(X)=(PEEK(56320)AND16) :rem 88
1058 SC$="{HOME}{15 RIGHT}{10 DOWN}"
                                :rem 121
1060 REM*** SOUND VRBLS         :rem 59
1064 L1=54272:H1=L1+1:W1=L1+4:W2=L1+11
                                :rem 55
1068 L3=L1+14:H3=L1+15:W3=L1+18 :rem 170
1070 FH=L1+22:FV=L1+23:VM=L1+24 :rem 231
1074 REM*** SETUP SOUNDS       :rem 155
1076 POKEL1+5,64:POKEL1+12,15:POKEL1+19,1
2                                :rem 128
1078 POKEL1+7,255:POKEL1+8,255 :rem 183
1082 POKEL1+6,0:POKEL1+13,240:POKEL1+20,1
2                                :rem 109
1084 POKEFH,90                  :rem 31
1100 GOSUB11000                 :rem 52
3000 REM*** DRAW COOPS         :rem 215
3004 FORJ=1984TO1024STEP-40:POKEJ+L1,10:P
OKEJ,90:POKEJ+L1+1,10:POKEJ+1,90
                                :rem 90
3006 POKEJ+L1+38,10:POKEJ+38,90:POKEJ+L1+
39,10:POKEJ+39,90:NEXT        :rem 194
3008 FORJ=1024TO1984STEP40:POKEJ+L1,13:PO
KEJ+L1+1,13:POKEJ+L1+38,13 :rem 204
3010 POKEJ+L1+39,13:NEXT:RETURN :rem 159
10000 REM*** TITLE             :rem 213
10001 PRINT"{CLR}{63}CHR$(142); :rem 19
10002 PRINT"{18 RIGHT}{RVS}{£ {*}"
                                :rem 104
10003 PRINT"{17 RIGHT}{RVS}{£{3 SPACES}
{*}"                            :rem 76
10004 PRINT"{17 RIGHT}{RVS}{2 SPACES}P
{2 SPACES}"                    :rem 21
10005 PRINT"{17 RIGHT}{RVS}{2 SPACES}R
{2 SPACES}"                    :rem 24
10006 PRINT"{17 RIGHT}{RVS}{2 SPACES}O
{2 SPACES}"                    :rem 22
10007 PRINT"{17 RIGHT}{RVS}{2 SPACES}P
{2 SPACES}"                    :rem 24
10008 PRINT"{17 RIGHT}{*}{RVS} S {OFF}
£"                            :rem 54
10009 PRINT"{18 RIGHT}{RVS}{3 SPACES}
{OFF}"                        :rem 121
10010 PRINT"{18 RIGHT}{*}{RVS} {OFF}{£"
                                :rem 249
10011 PRINT"{19 RIGHT}{RVS} {OFF}"
                                :rem 143
10012 PRINT"{17 RIGHT}{RVS}{£{3 SPACES}
{*}"                            :rem 76
10013 PRINT"{6 RIGHT}{RVS}{£{8 SPACES}
{*}{RIGHT} UCI {RIGHT}{£
{8 SPACES}{*}"                :rem 49
10014 PRINT"{6 RIGHT}{RVS}{2 SPACES}PROPS
{5 SPACES}- {5 SPACES}PROPS
{2 SPACES}"                    :rem 105
10015 PRINT"{6 RIGHT}{*}{RVS}{8 SPACES}
{OFF}{£{RIGHT}{RVS} JCK {RIGHT}
{OFF}{*}{RVS}{8 SPACES}{OFF}{£"
                                :rem 4
10016 PRINT"{17 RIGHT}{*}{RVS}
{3 SPACES}{OFF}{£"            :rem 226
10017 PRINT"{19 RIGHT}{RVS} {OFF}"
                                :rem 149
10018 PRINT"{18 RIGHT}{RVS}{£ {*}"
                                :rem 111
10019 PRINT"{18 RIGHT}{RVS}{3 SPACES}
{OFF}"                        :rem 122
10020 PRINT"{17 RIGHT}{RVS}{£ P {*}"
                                :rem 155
10021 PRINT"{17 RIGHT}{RVS}{2 SPACES}R
{2 SPACES}"                    :rem 22
10022 PRINT"{17 RIGHT}{RVS}{2 SPACES}O
{2 SPACES}"                    :rem 20
10023 PRINT"{17 RIGHT}{RVS}{2 SPACES}P
{2 SPACES}"                    :rem 22
10024 PRINT"{17 RIGHT}{RVS}{2 SPACES}S
{2 SPACES}"                    :rem 26
10025 PRINT"{17 RIGHT}{*}{RVS}
{3 SPACES}{OFF}{£"            :rem 226
10026 PRINT"{18 RIGHT}{*}{RVS} {OFF}{£
{HOME}{CYN}READING{2 DOWN}{3 LEFT}D
ATA..."                      :rem 73
10027 Q$="{28 RIGHT}":POKE214,19:PRINT
                                :rem 121
10028 PRINTQ$"JOYSTICK":PRINTQ$"{2 DOWN}
{2 RIGHT}IN PORT 2"           :rem 155
10999 RETURN                   :rem 236
11000 REM*** INSTRUCTIONS      :rem 9
11002 PRINT"{CLR}":POKE53272,31 :rem 84
11004 R$="{4 RIGHT}"           :rem 97
11006 PRINT"{63}{4 RIGHT}ZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZ"            :rem 136
11007 PRINT"{4 RIGHT}ZZ{25 RIGHT}ZZ"
                                :rem 123
11008 PRINTR$"ZZ{YEL} YOU ARE A PIGEON
{WHT}X{YEL} LOST {63}ZZ" :rem 245
11010 PRINTR$"ZZ{YEL} IN A DANGEROUS SKY
{SPACE}FULL {63}ZZ"          :rem 131
11012 PRINTR$"ZZ{YEL} OF WHIRLING PROPELL
ERS. {63}ZZ"                  :rem 10
11013 PRINT"{4 RIGHT}ZZ{25 RIGHT}ZZ"
                                :rem 120
11014 PRINTR$"ZZ{CYN} BACK AT THE COOP, Y
OUR{2 SPACES}{63}ZZ"          :rem 17
11016 PRINTR$"ZZ{CYN} MATE WAITS FAITHFUL
LY-- {33}X{63}Z"              :rem 245
11017 PRINT"{4 RIGHT}ZZ{25 RIGHT}ZZ"
                                :rem 124
11018 PRINTR$"ZZ{33} WATCH OUT FOR THE P
ROPS {63}ZZ"                  :rem 164
11020 PRINTR$"ZZ{33} WHILE YOU FLY FOR H
OME. {63}ZZ"                  :rem 113
11021 PRINT"{4 RIGHT}ZZ{25 RIGHT}ZZ"
                                :rem 119
11022 PRINTR$"ZZ{YEL} HIT FIRE BUTTON DUR
ING{2 SPACES}{63}ZZ"          :rem 140
11024 PRINTR$"ZZ{YEL} SCORE DISPLAY TO PA
USE, {63}ZZ"                  :rem 189
11026 PRINTR$"ZZ{YEL} OR CHANGE SKILL LEV
EL.{2 SPACES}{63}ZZ"          :rem 76
11027 PRINT"{4 RIGHT}ZZ{25 RIGHT}ZZ"
                                :rem 125
11028 PRINTR$"ZZ{CYN} HIT ZERO KEY DURING
ANY {63}ZZ"                   :rem 162
11030 PRINTR$"ZZ{CYN} PAUSE TO{2 SPACES}Q
UIT PLAYING. {63}ZZ"          :rem 130
11032 PRINT"{4 RIGHT}ZZ{25 RIGHT}ZZ"
                                :rem 121

```



```

11038 PRINT$"ZZ{YEL} PRESS FIRE BUTTON N      49176 DATA 20, 3, 169, 234, 141, 21
OW{3 SPACES}{6}ZZ" :rem 102 :rem 200
11040 PRINT$"ZZ{YEL} TO CHOOSE SKILL          49182 DATA 3, 88, 96, 169, 0, 141:rem 118
{2 SPACES}LEVEL. {6}ZZ" :rem 101 49188 DATA 73, 3, 238, 1, 208, 206
11042 PRINT"{4 RIGHT}ZZ{25 RIGHT}ZZ" :rem 161
:rem 122 49194 DATA 3, 208, 238, 5, 208, 206
11044 PRINT"{4 RIGHT}ZZZZZZZZZZZZZZZZZZZZZZ 49200 DATA 7, 208, 238, 9, 208, 206
ZZZZZZZZZZ" :rem 241 :rem 210
11050 POKEW3,16:POKEH3,1:POKEL3,90:POKEW2    49206 DATA 11, 208, 238, 13, 208, 206
,129 :rem 187 :rem 206
11052 SYS49608:POKEFH,PEEK(VM+3)/2:IFFNB(    49212 DATA 15, 208, 238, 73, 3, 173
0)THEN11052 :rem 70 :rem 42
11054 PRINT"{HOME}"CHR$(142):FORJ=1TO200:    49218 DATA 73, 3, 201, 2, 208, 222
NEXT:PRINTCHR$(14) :rem 205 :rem 205
11056 PRINT"{CLR}":POKEW3,20:SYS49920:REM    49224 DATA 173, 61, 3, 201, 5, 240
WHITE COL MEMORY :rem 121 :rem 142
11058 POKE843,254:POKE844,3:POKE251,220:P    49230 DATA 6, 238, 61, 3, 76, 49 :rem 63
OKEFH,2:RETURN :rem 44 49236 DATA 234, 173, 60, 3, 201, 224
12000 REM** SPRITE SHAPE DATA :rem 141 :rem 246
12001 IFPEEK(14081)=8ANDPEEK(15065)=195TH    49242 DATA 208, 5, 169, 220, 141, 60
ENRETURN :rem 161 :rem 250
12287 CK=0:FORJ=14080TO14142:READQ:CK=CK+    49248 DATA 3, 141, 248, 7, 141, 249
Q:POKEJ,Q:NEXT :rem 162 :rem 212
12288 DATA 0,8,0,0,24,0,0,56,0,0,56,0,0,5 49254 DATA 7, 141, 250, 7, 141, 251
6,0,0,24,0,0,16,0,0,16,0,0,0,0 :rem 199
:rem 58 49260 DATA 7, 141, 252, 7, 141, 253
12289 DATA 0,24,0,0,60,0,0,24,0,0,0,0,0,8 49266 DATA 7, 141, 254, 7, 141, 255
,0,0,8,0,0,24,0,0,28,0,0,28,0,0,28 :rem 210
:rem 246 49272 DATA 7, 238, 60, 3, 169, 0 :rem 59
12290 DATA 0,0,24,0,0,16,0 :rem 221 49278 DATA 141, 61, 3, 173, 69, 192
12351 FORJ=14144TO14206:READQ:CK=CK+Q:POK    :rem 218
EJ,Q:NEXT :rem 103 49284 DATA 10, 10, 10, 10, 10, 141
12352 DATA 0,0,0,16,0,0,28,0,0,14,0,0,15, 49290 DATA 78, 3, 24, 173, 77, 3 :rem 66
0,0,7,128,0,3,128,0,1,192,0,0,128 :rem 119
:rem 215 49296 DATA 109, 78, 3, 141, 77, 3:rem 119
12353 DATA 0,0,24,0,0,60,0,0,24,0,0,2,0,0 49302 DATA 141, 22, 212, 206, 8, 212
,3,128,0,1,192,0,1,224,0,0,240,0,0 :rem 236
:rem 238 49308 DATA 76, 49, 234 :rem 237
12354 DATA 112,0,0,56,0,0,8,0,0,0 :rem 44 49350 IFCK<>22679THENPRINT"{3 DOWN}ERROR
12415 FORJ=14208TO14270:READQ:CK=CK+Q:POK    {SPACE}IN DATA LINES 49000-49308":E
EJ,Q:NEXT :rem 106 :rem 210
12416 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 49400 REM*** POKE BIRDMOVE RTN :rem 222
0,0,0,0,0,0,0,0,0,0 :rem 232 49407 CK=0:FORJ=49408TO49643:READQ:CK=CK+
12417 DATA 0, 0,124,24, 62,255,189,255,12    Q:POKEJ,Q:NEXT :rem 192
4,24, 62, 0,0, 0,0,0,0,0,0,0,0,0,0 :rem 150
:rem 37 49408 DATA 160, 0, 169, 32, 145, 251
12418 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 49414 DATA 165, 251, 133, 253, 165, 252
:rem 82 :rem 150
12479 FORJ=14272TO14334:READQ:CK=CK+Q:POK    49420 DATA 133, 254, 173, 0, 220, 162
EJ,Q:NEXT :rem 118 :rem 35
12480 DATA 0,0,0,0,0,8,0,0,56,0,0,112,0,0 49426 DATA 0, 74, 176, 1, 136, 74:rem 110
,240,0,1,224,0,1,192,0,3,128,0,2 :rem 151
:rem 157 49432 DATA 176, 1, 200, 74, 176, 1
12481 DATA 0,0,24,0,0,60,0,0,24,0,0,64,0, 49438 DATA 202, 74, 176, 1, 232, 74
1,192,0,3,128,0,7,128,0,15 :rem 210
12482 DATA 0,0,14,0,0,28,0,0,16,0,0,0,0,0 49444 DATA 142, 64, 3, 140, 65, 3:rem 102
:rem 157 49450 DATA 173, 64, 3, 201, 1, 240
12500 IFCK<>5053THENPRINT"{3 DOWN}ERROR I    :rem 142
N DATA LINES 12000-12482":END :rem 54
49000 REM*** POKE PROPMOVE RTN :rem 250 49468 DATA 198, 254, 198, 253, 76, 73
49151 CJ=0:FORJ=49152TO49310:READQ:CK=CK+    :rem 85
Q:POKEJ,Q:NEXT :rem 174 49474 DATA 193, 230, 253, 208, 2, 230
49152 DATA 120, 173, 21, 3, 201, 234 :rem 47
:rem 235 49480 DATA 254, 160, 0, 177, 253, 201
49158 DATA 208, 13, 169, 33, 141, 20 :rem 45
:rem 253
49164 DATA 3, 169, 192, 141, 21, 3
49170 DATA 76, 31, 192, 169, 49, 141
:rem 11

```



```

49486 DATA 32, 208, 3, 76, 91, 193
                                     :rem 171
49492 DATA 201, 90, 208, 14, 76, 108
                                     :rem 3
49498 DATA 193, 165, 253, 133, 251, 165
                                     :rem 166
49504 DATA 254, 133, 252, 76, 116, 193
                                     :rem 105
49510 DATA 169, 1, 141, 74, 3, 96:rem 109
49516 DATA 165, 251, 133, 253, 165, 252
                                     :rem 153
49522 DATA 133, 254, 173, 65, 3, 201
                                     :rem 250
49528 DATA 1, 240, 39, 201, 255, 240
                                     :rem 250
49534 DATA 3, 76, 200, 193, 162, 0
                                     :rem 151
49540 DATA 165, 253, 208, 2, 198, 254
                                     :rem 59
49546 DATA 198, 253, 232, 224, 40, 208
                                     :rem 107
49552 DATA 243, 24, 165, 253, 205, 67
                                     :rem 56
49558 DATA 3, 165, 254, 237, 68, 3
                                     :rem 172
49564 DATA 176, 34, 240, 32, 144, 38
                                     :rem 6
49570 DATA 162, 0, 230, 253, 208, 2
                                     :rem 195
49576 DATA 230, 254, 232, 224, 40, 208
                                     :rem 98
49582 DATA 245, 24, 165, 253, 205, 69
                                     :rem 63
49588 DATA 3, 165, 254, 237, 70, 3
                                     :rem 168
49594 DATA 144, 4, 240, 2, 176, 8:rem 112
49600 DATA 165, 253, 133, 251, 165, 254
                                     :rem 149
49606 DATA 133, 252, 160, 0, 173, 66
                                     :rem 253
49612 DATA 3, 145, 251, 173, 75, 3
                                     :rem 155
49618 DATA 133, 253, 173, 76, 3, 133
                                     :rem 5
49624 DATA 254, 173, 66, 3, 145, 253
                                     :rem 8
49630 DATA 201, 96, 240, 4, 238, 66
                                     :rem 209
49636 DATA 3, 96, 169, 91, 141, 66
                                     :rem 176
49642 DATA 3, 96
                                     :rem 241
49650 IFCK<>33160THENPRINT"{3 DOWN}ERROR
{SPACE}IN DATA LINES 49400-49642":E
ND
                                     :rem 205
49662 REM*** POKE COPYCHAR RTN
                                     :rem 233
49663 CK=0:FORJ=49664TO49704:READQ:CK=CK+
Q:POKEJ,Q:NEXT
                                     :rem 198
49664 DATA 169,0,133,4,169,208,133,5
                                     :rem 252
49672 DATA 169,0,133,6,169,56,133,7
                                     :rem 208
49680 DATA 162,0,160,0,177,4,145,6
                                     :rem 140
49688 DATA 200,192,255,208,247,230,5,230
                                     :rem 193
49696 DATA 7,232,224,16,208,236,160,0,96
                                     :rem 196
49700 IFCK<>4894THENPRINT"{3 DOWN}ERROR I
N DATA LINES 49662-49696":END
                                     :rem 184
49918 REM*** POKE WHITEMEM RTN
                                     :rem 244

49919 CK=0:FORJ=49920TO49939:READQ:CK=CK+
Q:POKEJ,Q:NEXT
                                     :rem 207
49920 DATA 162, 0, 169, 1, 157, 0:rem 101
49926 DATA 216, 157, 0, 217, 157, 0
                                     :rem 209
49932 DATA 218, 157, 0, 219, 232, 208
                                     :rem 54
49938 DATA 241, 96
                                     :rem 93
49950 IFCK<>2607THENPRINT"{3 DOWN}ERROR I
N DATA LINES 49918-49938":END
                                     :rem 184
62000 REM* COPY CHAR SET TO 14336:rem 239
62002 POKE56334,PEEK(56334)AND254:POKE1,P
EEK(1)AND251
                                     :rem 27
62004 SYS49664
                                     :rem 8
62006 POKE1,PEEK(1)OR4:POKE56334,PEEK(563
34)OR1
                                     :rem 237
62400 REM** POKE NEW CHARACTER DATA **
                                     :rem 72
62500 CK=0:FORJ=15064TO15071:READQ:CK=CK+
Q:POKEJ,Q:NEXT
                                     :rem 160
62501 DATA 0,195,102,60,24,0,0,0
                                     :rem 10
62502 FORJ=15072TO15079:READQ:CK=CK+Q:POK
EJ,Q:NEXT
                                     :rem 116
62503 DATA 0,0,195,126,24,0,0,0
                                     :rem 220
62504 FORJ=15080TO15087:READQ:CK=CK+Q:POK
EJ,Q:NEXT
                                     :rem 116
62505 DATA 0,0,66,255,153,0,0,0
                                     :rem 225
62506 FORJ=15088TO15095:READQ:CK=CK+Q:POK
EJ,Q:NEXT
                                     :rem 125
62507 DATA 0,0,0,90,255,129,0,0
                                     :rem 227
62508 FORJ=15096TO15103:READQ:CK=CK+Q:POK
EJ,Q:NEXT
                                     :rem 116
62509 DATA 0,0,0,24,126,195,0,0
                                     :rem 226
62510 FORJ=15104TO15111:READQ:CK=CK+Q:POK
EJ,Q:NEXT
                                     :rem 98
62511 DATA 0,0,0,24,60,102,195,129
                                     :rem 119
62512 FORJ=15056TO15063:READQ:CK=CK+Q:POK
EJ,Q:NEXT
                                     :rem 112
62513 DATA 255, 60, 24, 24, 24, 24, 60,25
5
                                     :rem 235
62600 IFCK<>3255THENPRINT"{3 DOWN}ERROR I
N DATA LINES 62000-62513":END
                                     :rem 132
63000 RETURN
                                     :rem 217

```

## SuperSprite

(Article on page 64.)

### 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 REMA:
20 :
21 REM{9 SPACES}LIST OF VARIABLES:rem 177
22 REM SP SPRITE PAGE
23 REM V{2 SPACES}BASE OF VIDEO CHIP
24 REM C{2 SPACES}BASE OF COLOR RAM
25 REM SD BASE OF SOUND CHIP
26 REM S{2 SPACES}BASE OF SCREEN

```



```

27 REM HS BYTE TO SAVE HIGH SCORE:rem 138
28 REM CH KERNAL PRINT ROUTINE (CHROUT)
:rem 111
29 REM AR IMAGE OF ACCUMULATOR REGISTER
:rem 127
30 REM BS BEST SCORE YET AT RUNTIME
:rem 49
31 REM ZS OLD SCREEN BASE AND CHAR SET
:rem 140
32 REM ZC OLD BACKGROUND COLOR :rem 36
33 REM ZE OLD BORDER COLOR :rem 5
34 REM ZR OLD VALUE OF REPEAT FLAG
:rem 195
35 REM ZL OLD PRINT COLOR :rem 221
36 REM GL TURN NUMBER IN GAME LOOP
:rem 221
37 REM I AND J{2 SPACES}FOR-NEXT INDICES
:rem 6
38 REM U, U2 AND U$ WORKHORSES :rem 186
39 REM T1-T9{2 SPACES}CONSTANTS IN GAME L
OOP :rem 55
40 REM B$ GAME SCREEN BUILDING BLOCK
:rem 65
41 REM T$ 16 SPACE INDENT :rem 86
42 REM W{2 SPACES}EITHER V OR V+1 (SPRITE
POS.) :rem 83
43 REM PS PART-SCORE (ONE TURN) :rem 74
44 REM TS TOTAL SCORE SO FAR :rem 110
45 REM FQ QUIT FLAG, SET BY PRESSING *
:rem 137
46 : :rem 164
47 REM{9 SPACES}SUBROUTINES :rem 178
48 REM{2 SPACES}1000 ONE TURN OF THE GAME
:rem 204
49 REM{2 SPACES}2000 PRINT GAME SCREEN
:rem 122
50 REM{2 SPACES}3000 EXIT (OR PLAY AGAIN)
:rem 206
51 REM{2 SPACES}4000 BUILD SPRITE :rem 85
52 REM{2 SPACES}5000 INSTRUCTIONS DATA ST
RINGS :rem 9
53 REM{2 SPACES}6000 PRINT INSTRUCTIONS
:rem 84
54 REM{2 SPACES}7000 FLY SPRITE AHEAD OF
{SPACE}INSTR. :rem 124
55 REM{2 SPACES}8000 WAIT TO RESUME GAME
:rem 217
56 : :rem 165
100 REM SEED RANDOM NUMBER GENERATOR, DIS
ABLE CHARACTER SET SHIFTS :rem 49
110 U=RND(-TI):POKE657,128 :rem 245
120 REM SAVE OLD ENVIRONMENT, BUILD NEW O
NE, BUILD SPRITE, PRINT INSTRUCTIONS
:rem 35
130 SP=13:V=53248 :rem 171
140 POKE2041,PEEK(V+24):POKEV+24,22
:rem 244
150 POKE2042,PEEK(V+32):POKEV+32,7
:rem 199
160 POKE2043,PEEK(V+33):POKEV+33,7
:rem 203
170 POKE2044,PEEK(646):POKE646,0 :rem 56
180 POKE2045,PEEK(650):POKE650,128
:rem 155
190 PRINT"{CLR}":GOSUB4000 :rem 124
200 GOSUB6000:PRINT"{CLR}":POKEV+24,20
:rem 84
210 DEFFNSC(U)=INT(U+U*GL/10) :rem 205
220 FORI=1TO16:T$=T$+"{RIGHT}":NEXTI
:rem 241
230 B$="{3 DOWN}"+CHR$(13)+"{RVS}":FORI=0
TO39:B$=B$+" ":NEXTI :rem 107
240 V=53248:C=55296:S=1024:SD=54272
:rem 151
250 HS=PEEK(43)+PEEK(44)*256+5 :rem 124
260 T1=135:T2=255:T3=230:T4=133 :rem 181
270 T5=5:T6=9:T7=.992:T8=42:T9=32:rem 116
280 BS=PEEK(HS):CH=65508:AR=780 :rem 48
290 REM INITIALIZE CURSORS, PRINT GAME SC
REEN :rem 53
300 E(1)=1269:E(2)=1475:E(3)=1682:E(4)=18
88 :rem 164
310 POKEV+21,0:GOSUB2000 :rem 127
320 REM MAKE SPACE FOR SCORE ON SCREEN, S
ET SOUND CHIP, CLEAR KEYBOARD BUFFER
:rem 38
330 FORI=1080TO1100:POKEI,32:NEXT:rem 237
340 POKESD+4,16:POKESD+11,16:POKESD+18,16
:rem 150
350 POKESD+6,240:POKESD+13,240:POKESD+20,
240 :rem 33
360 POKESD+24,15 :rem 130
370 GETA$:IFA$<>" "THEN370 :rem 148
380 : :rem 213
390 REM BEGIN 20-TURN GAME LOOP,SET PART
{SPACE}SCORE TO ZERO :rem 191
400 FORGL=1TO20:PS=0 :rem 203
410 REM POSITION SPRITE, UNEXPAND IT, ZER
O THE COLLISION DETECT REGISTER:rem 8
420 POKEV,23:POKEV+1,49 :rem 96
430 POKEV+23,0:POKEV+31,0 :rem 178
440 REM PRINT SCORE INFO, TURN ON SOUND
:rem 76
450 PRINT"{HOME}{DOWN}"T$"TURN:"GL"
{2 SPACES}SCORE:"TS :rem 1
460 PRINT"{DOWN}"T$"{5 RIGHT}BEST YET:"PE
EK(HS)-1 :rem 146
470 POKESD+4,17:POKESD+11,17:POKESD+18,17
:rem 157
480 REM TURN ON SPRITE, TAKE 1 TURN, GOTO
570 IF GAME ABORTED (FQ SET):rem 135
490 POKEV+21,1:GOSUB1000:IFFQ=1THEN550
:rem 31
500 REM SCROLL SPRITE OFF SCREEN IF NO CO
LLISION, OTHERWISE SCREECH TO A HALT
:rem 130
510 IFPEEK(V+1)>T3THENFORI=PEEK(V+1)TO250
:POKEV+1,I:NEXTI:GOTO550 :rem 111
520 POKESD+4,129:POKESD+11,33:POKESD+18,3
3 :rem 201
530 FOR I=72TO20STEP-2:POKESD+1,I:POKESD+
8,I-3:POKESD+15,I-2:NEXTI :rem 62
540 REM TURN OFF SOUND. JUMP TO EXIT IF F
Q SET :rem 131
550 POKESD+4,16:POKESD+11,16:POKESD+18,16
:rem 153
560 POKESD+1,0:POKESD+8,0:POKESD+15,0
:rem 197
570 IFFQ=1THENFQ=0:GL=20:NEXT:POKEV+21,0:
GOTO670 :rem 1
580 REM CALCULATE PART SCORE FROM SPRITE
{SPACE}Y POSITION, MODIFY, ADD TO TOT
AL SCORE :rem 39
590 PS=FNSC(INT((PEEK(V+1)-71)/40))
:rem 214
600 TS=TS+INT((PS↑1.4+PS)/2) :rem 156
610 POKEV+21,0:POKEV+16,0 :rem 179
620 NEXTGL :rem 106
630 REM UPDATE HIGH SCORE RECORD, ZERO SO
UND CHIP, GO TO EXIT (3000) :rem 103
640 IFPEEK(HS)<TS+1THENPOKEHS,TS+1
:rem 161

```



```

650 FORI=0TO24:POKE SD+I,0:NEXTI :rem 211
660 FORI=1TO500:NEXTI :rem 52
670 GOSUB3000:CLR:GOTO210 :rem 1
680 : :rem 216
970 REM SUBROUTINE--ONE TURN OF THE GAME :rem 169
: :rem 219
980 : :rem 219
990 REM DECIDE WHICH SPRITE POSITION REGI
STER TO ALTER (W) AND BY HOW MUCH (U)
:rem 58
1000 U2=RND(1):W=V+INT(U2*2):U=GL/7+2 :rem 123
:rem 123
1010 REM Y-EXPAND SPRITE IF EXPRESSION TR
UE :rem 162
1020 IFU>T7ANDPEEK(V+1)<T1THENPOKEV+23,1 :rem 255
:rem 255
1030 REM SET MSB OF SPRITE X-POSITION IF
{SPACE}NECESSARY :rem 24
1040 IFPEEK(W)+U>T2THENPOKEV+16,1:POKEV,P
EEK(V)+U-T2:GOTO1070 :rem 45
1050 REM UPDATE SPRITE POSITION.
{2 SPACES}EXIT IF COLLISION DETECTED
OR 4TH GAP CROSSED :rem 250
1060 POKEW,PEEK(W)+U :rem 198
1070 IF(PEEK(V+31)AND1)ORPEEK(V+1)>T3THEN
RETURN :rem 164
1080 REM CHANGE FREQUENCIES ACCORDING TO
{SPACE}SPRITE Y POSITION AND RANDOM
{SPACE}NUMBER U2 :rem 67
1090 U=232-PEEK(V+1):POKESD+1,U*2/3:POKES
D+8,U+3*U2:POKESD+15,U+3 :rem 70
1100 REM POLL KEYBOARD. EXIT ON *, PAUSE
{SPACE}ON ' '; GOTO1000 IF NOT A FUN
CTION KEY :rem 70
1110 SYSCH:U=PEEK(AR):IFU=T8THENT8S=0:FQ=1
:RETURN{17 SPACES}:REM EXIT :rem 253
1120 IFU=T9THENGOSUB8000:REM TAKE 5 :rem 97
:rem 97
1130 IFU<T4THEN1000:REM INPUT<F1 :rem 190
1140 REM MOVE LINE RIGHT IF KEY NOT SHIFT
ED (1120) LEFT IF SHIFTED (1140) :rem 16
:rem 16
1150 U=U-132:IFU<T5THEN1180:REM INPUT F1-
F4 :rem 25
:rem 25
1160 IFU<T6THEN1200:REM INPUT F5-F8 :rem 56
:rem 56
1170 GOTO1000:REM INPUT>F8 :rem 45
1180 E(U)=E(U)+(E(U)=1100+200*U-(U>2)):PO
KEE(U),223:POKEE(U)-1,160 :rem 78
1190 POKE E(U)+4+(U>2),32:POKEE(U)+5+(U>2
),95:E(U)=E(U)+1:GOTO1000 :rem 130
1200 U=U-4:E(U)=E(U)-(E(U)=1064+200*(U)):
POKEE(U)+3+(U>2),95 :rem 227
1210 POKEE(U)+4+(U>2),160:POKEE(U)-1,32:P
OKEE(U)-2,223:E(U)=E(U)-1:GOTO1000 :rem 217
:rem 217
1970 REM SUBROUTINE--PRINT GAME SCREEN :rem 134
:rem 134
1980 : :rem 12
1990 REM LINE COLORS SET IN LINE 2000. LA
ST COLOR IS FOR PRINTED MESSAGES :rem 240
:rem 240
2000 PRINT"{CLR}{2 DOWN}{PUR}"B$"{GRN}"B$
"£7}"B$"{£3}"B$"{HOME}{BLK}" :rem 125
:rem 125
2010 FORI=1TO4:POKEE(I)-1,223:POKEE(I)+3-
(I<3),95:FORJ=E(I)TOE(I)+2-(I<3) :rem 179
:rem 179
2020 POKEJ,32:NEXTJ,I:RETURN{7 SPACES} :rem 32
:rem 32
2030 : :rem 255
2980 REM SUBROUTINE--EXIT OR RE-ENTRY :rem 82
:rem 82
2990 : :rem 14
3000 PRINT"{CLR}{DOWN}"T$"{2 RIGHT}YOUR S
CORE:"TS :rem 218
3010 PRINT"{DOWN}"T$"{4 RIGHT}BEST YET:";
PEEK(HS)-1 :rem 218
3020 PRINT"{3 DOWN}{RIGHT}PLAY AGAIN (Y/N
)? :rem 192
3030 GETA$:IFA$=""THEN3030 :rem 175
3040 IFA$="Y"THENRETURN :rem 164
3050 IFA$="N"THEN3070 :rem 132
3060 GOTO3030 :rem 200
3070 IF NOT(BS<PEEK(HS))THEN3100 :rem 112
3080 PRINT:PRINT" CONGRATULATIONS, YOU BR
OKE THE RECORD. :rem 40
3090 PRINT:PRINT" BE SURE TO SAVE THE PRO
GRAM. :rem 1
3100 PRINT:PRINT" SEE YOU AROUND. :rem 11
3110 PRINT"{4 DOWN}" :rem 218
3120 REM RESTORE PREVIOUS ENVIRONMENT :rem 160
:rem 160
3130 POKEV+24,PEEK(2041):POKEV+32,PEEK(20
42):POKEV+33,PEEK(2043) :rem 186
3140 POKE646,PEEK(2044):POKE650,PEEK(2045
):POKE657,0 :rem 219
3150 END :rem 160
3160 : :rem 4
3980 REM SUBROUTINE--BUILD SPRITE AT PAGE
13 (LOCATIONS 832-895 IN TAPE BUFFE
R) :rem 103
3990 : :rem 15
4000 FOR I=0TO41:READU:POKE(64*SP)+I,U:NE
XT :rem 115
4010 FORI=42TO63:POKE(64*SP)+I,0:NEXT :rem 222
:rem 222
4020 POKE2040,SP:POKEV+23,1:POKEV+29,1:PO
KEV+39,0:RETURN :rem 180
4030 RETURN :rem 167
4040 DATA 248,0,0,62,0,0,7,128,0,1,224 :rem 43
:rem 43
4050 DATA 0,0,120,0,0,62,0,0,15,112 :rem 132
:rem 132
4060 DATA 0,7,248,0,7,248,0,6,120,0 :rem 163
:rem 163
4070 DATA 6,12,0,6,6,0,3,6,0,1,128 :rem 103
:rem 103
4080 : :rem 6
4980 REM DATA FOR INSTRUCTIONS PAGE :rem 140
:rem 140
4990 : :rem 16
5000 DATA"YOUR OBJECTIVE IS TO MANEUVER T
HE :rem 238
5010 DATA"HORIZONTAL LINES SO THAT SUPERS
PRITE :rem 192
5020 DATA"MAY FLY SAFELY THROUGH THE GAPS
. :rem 244
5030 DATA"THE FOUR LINES ARE CONTROLLED B
Y THE :rem 102
5040 DATA"FOUR FUNCTION KEYS ON YOUR RIGH
T. :rem 123
5050 DATA :rem 6
5060 DATA"PRESSING ONE OF THESE KEYS WILL
CAUSE :rem 199
5070 DATA"THE CORRESPONDING LINE TO SLIDE
RIGHT. :rem 174
5080 DATA"THE SAME KEY SHIFTED WILL CAUSE
ITS :rem 25
5090 DATA"LINE TO SLIDE LEFT.{2 SPACES}PR
ESS THE SPACE :rem 249

```



```

5100 DATA"BAR TO PAUSE, '*' TO ABORT.
                                     :rem 229
5110 DATA"                                     :rem 3
5120 DATA"A GAME CONSISTS OF 20 TURNS. PO
      INTS ARE                                     :rem 75
5130 DATA"AWARDED FOR EVERY GAP SAFELY TR
      AVERSED.                                     :rem 233
5140 DATA"THE PAYOFFS INCREASE WITH THE N
      UMBER OF                                     :rem 68
5150 DATA"GAPS TRAVERSED ON A TURN, AND W
      ITH THE                                     :rem 37
5160 DATA"NUMBER OF TURNS TAKEN.
      {2 SPACES}SUPERSPRITE'S                     :rem 131
5170 DATA"FLIGHT SPEED AND NATURAL WAYWAR
      DNESS                                         :rem 122
5180 DATA"ALSO INCREASE AS THE GAME PROGR
      ESSES."                                       :rem 111
5190 DATA"                                     :rem 11
5200 DATA"THE MAXIMUM SCORE IS 253. GOOD
      {SPACE}LUCK.                                 :rem 104
5210 DATA"                                     :rem 4
5220 DATA"{4 SPACES}PRESS{SHIFT-SPACE}SPA
      CE{SHIFT-SPACE}BAR{SHIFT-SPACE}TO ST
      ART{SHIFT-SPACE}GAME.                         :rem 204
5230 :                                     :rem 4
5980 REM SUBROUTINE--PRINT INSTRUCTIONS
                                     :rem 102
5990 :                                     :rem 17
6000 POKEV,23:POKEV+1,49:POKEV+21,1
                                     :rem 58
6010 GOSUB7000:GOSUB7000                     :rem 143
6020 FORW=1TO23:READU$:PRINTU$:GOSUB7000:
      NEXTW                                         :rem 220
6030 POKEV+21,0:POKEV,23:POKEV+1,49:POKEV
      +23,0:POKEV+29,0                             :rem 152
6040 GETA$:IFA$<>CHR$(32)THEN 6040
                                     :rem 103
6050 RETURN                                     :rem 171
6060 :                                     :rem 6
6980 REM SUBROUTINE--FLY SPRITE DOWN 1 PR
      INT LINE (AHEAD OF INSTRUCTIONS)
                                     :rem 243
6990 :                                     :rem 18
7000 FORI=1TO8:POKEV,PEEK(V)+1                 :rem 83
7010 POKEV+1,PEEK(V)+1:NEXT                     :rem 210
7020 POKEV,PEEK(V)+1:RETURN                     :rem 188
7030 :                                     :rem 4
7980 REM SUBROUTINE--ANSWER THE PHONE
                                     :rem 81
7990 :                                     :rem 19
8000 POKE SD+24,0                               :rem 123
8010 SYSCH:IFPEEK(AR)<>32THEN8010:rem 252
8020 POKE SD+24,15                             :rem 179
8030 RETURN                                     :rem 171

```

```

10 POKE204,0:GOSUB800:POKE650,128:SP=1
                                     :rem 234
20 GETA$:IFA$=""THEN20                         :rem 231
22 IFA$=CHR$(34)THEN20                         :rem 223
25 IFA$="{CLR}"THENGOSUB800:A$=""GOTO20
                                     :rem 171
30 IFASC(A$)<141ANDASC(A$)>132THENGOSUB10
      0                                             :rem 197
32 IFA$=""THEN20                               :rem 107
35 IFASC(A$)=13AND(PEEK(210)=7)AND(PEEK(2
      09)>151)THEN20                             :rem 97
40 POKE205,3:WAIT207,1:PRINTA$;               :rem 85
50 IFPEEK(210)=7AND(PEEK(209)+PEEK(211)>1
      91)THENPOKE205,3:WAIT207,1:PRINT"UP"
      ;                                             :rem 68
60 GOTO20                                       :rem 1
100 X=0                                         :rem 86
110 IFA$="{F1}"THENX=5                         :rem 133
120 IFA$="{F3}"THENX=10                        :rem 179
130 IFA$="{F5}"THENX=15                        :rem 186
140 IFA$="{F7}"THENX=20                        :rem 184
150 IFA$="{F2}"THENX=25                        :rem 191
160 IFX>0THENA$="":FORQ=1TOX:A$=A$+"
      {RIGHT}":NEXTQ:RETURN                       :rem 141
170 IFA$="{F4}"THENIFSP=1THENSP=2:POKE198
      4,178:POKE56256,1:RETURN                   :rem 154
175 IFA$="{F4}"THENSP=1:POKE1984,177:POKE
      56256,1:RETURN                             :rem 206
180 IFA$="{F6}"THENG1=984:TF=14:RW=23:CL=
      40:REM EXPANDED                             :rem 238
190 IFA$="{F8}"THENG1=944:TF=15:RW=11:CL=
      80:REM NORMAL                               :rem 110
200 A$=""                                       :rem 120
210 GOSUB60000                                :rem 9
220 RETURN                                     :rem 116
500 POKE53281,0:POKE53280,2                   :rem 236
510 PRINT"{CLR}{WHT}{2 DOWN}{13 RIGHT}MEM
      O WRITER"                                     :rem 165
530 PRINT"{5 DOWN}{6 RIGHT}THIS IS A SCRE
      EN-ORIENTED"                               :rem 200
540 PRINT"{6 RIGHT}WORD PROCESSING PROGRA
      M"                                           :rem 105
550 PRINT"{6 RIGHT}USING THE COMMODORE-64
      'S"                                         :rem 54
560 PRINT"{8 RIGHT}OWN BUILT-IN EDITING"
                                     :rem 144
570 PRINT"{12 RIGHT}CAPABILITIES."             :rem 97
590 PRINT"{7 DOWN}{2 RIGHT}HIT ANY KEY FO
      R LIST OF CONTROL KEYS"                   :rem 235
600 POKE198,0                                  :rem 195
610 GETA$:IFA$=""THEN610                       :rem 81
620 PRINT"{CLR}{2 DOWN}{2 RIGHT}F1 F3 F5
      {SPACE}F7 F2{5 SPACES}TAB FROM CURREN
      T"                                           :rem 230
630 PRINT"{21 RIGHT}POSITION IN INCRE-"
                                     :rem 117
640 PRINT"{21 RIGHT}MENTS OF FIVE."
                                     :rem 64
645 PRINT"{DOWN}{3 RIGHT}F4{8 SPACES}SET
      {SPACE}SINGLE OR DOUBLE SPACE"
                                     :rem 200
650 PRINT"{DOWN}{3 RIGHT}F6{8 SPACES}PRIN
      T EXPANDED"                                 :rem 38
660 PRINT"{13 RIGHT}CHARACTERS"                 :rem 198
670 PRINT"{DOWN}{3 RIGHT}F8{8 SPACES}PRIN
      T IN NORMAL SIZE"                         :rem 124
680 PRINT"{13 RIGHT}CHARACTERS."               :rem 246
690 PRINT"{2 DOWN}{RIGHT}ALL EDITING KEYS
      WORK AS NORMAL."                           :rem 150
700 PRINT"{RIGHT}TEXT CANNOT SCROLL PAST
      {SPACE}END OF SCREEN."                   :rem 238

```

## Memo Writer

(Article on page 72.)

### 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: Memo Writer—64 Version

9 GOSUB500 :rem 78



```

710 PRINT"{RIGHT}YOU MAY TYPE IN UPPERCAS 100 X=0 :rem 86
E/GRAPHICS OR " :rem 12 110 IFA$="{F1}"THENX=5 :rem 133
720 PRINT"{RIGHT}UPPERCASE/LOWERCASE MODE 120 IFA$="{F3}"THENX=10 :rem 179
; THE PRINT" :rem 209 130 IFA$="{F5}"THENX=15 :rem 186
730 PRINT"{RIGHT}ROUTINE WILL AUTOMATICAL 140 IFA$="{F7}"THENX=20 :rem 184
LY" :rem 191 150 IFA$="{F2}"THENX=25 :rem 191
740 PRINT"{RIGHT}SET THE PRINT MODE CORRE 160 IFX>0THENA$="":FORQ=1TOX:A$=A$+"
CTLY." :rem 237 {RIGHT}":NEXTQ:RETURN :rem 141
750 PRINT"{2 DOWN}{9 RIGHT}HIT ANY KEY TO 170 IFA$="{F4}"THENIFSP=1THENSP=2:POKE816
BEGIN" :rem 82 4,178:RETURN :rem 201
760 POKE198,0 :rem 202 175 IFA$="{F4}"THENSP=1:POKE8164,177:RETU
770 GETA$:IFA$=""THEN770 :rem 95 RN :rem 253
780 RETURN :rem 127 180 IFA$="{F6}"THENG1=7640:TF=14:RW=11:CL
800 POKE53281,1:PRINT"{CLR}>";:POKE53281, =40:REM EXPANDED :rem 23
0:FORX=1TO11:PRINT"{2 DOWN}{LEFT}>";: 190 IFA$="{F8}"THENG1=7600:TF=15:RW=5:CL=
NEXTX :rem 6 80:REM NORMAL :rem 109
810 PRINT"{2 DOWN}{LEFT}{13 Y}{RVS}MEM 200 A$="" :rem 120
O WRITER{OFF}{14 Y}{HOME}";:rem 115 210 GOSUB60000 :rem 9
820 RETURN :rem 122 220 RETURN :rem 116
60000 Q=PEEK(53272):IFQ=21THENG1$=CHR$(14 500 POKE 36879,9 :rem 58
5):GOTO60010 :rem 21 510 PRINT"{CLR}{RVS}{WHT}{4 DOWN}
60005 G1$=CHR$(17) :rem 142 {5 RIGHT}MEMO WRITER" :rem 241
60010 G1$=G1$+CHR$(TF) :rem 131 530 PRINT"{2 DOWN}{2 RIGHT}THIS IS A SCRE
60020 OPEN4,4:WAIT207,1:POKE204,255 :rem 234 EN-" :rem 199
:rem 234 540 PRINT"{4 RIGHT}ORIENTED WORD":rem 116
60030 FORG0=0TORW:G0$=G1$:G1=G1+CL 545 PRINT"{2 RIGHT}PROCESSING PROGRAM"
:rem 223 :rem 190
60040 FORG2=G1TOG1+(CL-1):G3=PEEK(G2) 550 PRINT"{2 RIGHT}USING THE VIC-20'S"
:rem 115 :rem 247
60050 IFG3>128THENG3=G3-128:G4=1:G0$=G0$+ 560 PRINT"{RIGHT}OWN BUILT-IN EDITING"
CHR$(18) :rem 187 :rem 197
60060 IF(G3>0)*(G3<32)THENG3=G3+64:GOTO60 570 PRINT"{5 RIGHT}CAPABILITIES.":rem 150
100 :rem 185 590 PRINT"{4 DOWN}{5 RIGHT}{RVS}HIT ANY K
60070 IF(G3>31)*(G3<64)THENG3=G3+128:GOTO 600 POKE198,0 :rem 195
60080 IF(G3>63)*(G3<96)THENG3=G3+128:GOTO 610 GETA$:IFA$=""THEN610 :rem 81
60090 IF(G3>95)*(G3<128)THENG3=G3+64:GOTO 620 PRINT"{CLR}{4 DOWN}{2 RIGHT}F1,F3,F5,
60100 :rem 47 F7,F2-TAB" :rem 142
60100 G0$=G0$+CHR$(G3) :rem 97 630 PRINT"IN INCREMENTS OF FIVE":rem 184
60110 IFG4=1THENG0$=G0$+CHR$(146):G4=0 645 PRINT"{DOWN}{4 RIGHT}F4-SET SINGLE"
:rem 76 :rem 74
60120 NEXTG2:PRINT#4,G0$:IFSP=2THENPRINT# 647 PRINT"{2 RIGHT}OR DOUBLE SPACING"
4 :rem 13 :rem 13
60130 NEXTG0:CLOSE4:POKE204,0 :rem 239 650 PRINT"{DOWN}{2 RIGHT}F6-PRINT EXPANDE
60140 RETURN :rem 219 D" :rem 54
660 PRINT"{5 RIGHT}CHARACTERS" :rem 222
670 PRINT"{DOWN}{3 RIGHT}F8-PRINTS NORMAL
" :rem 42

```

## Program 2:

### Memo Writer—VIC Version

```

9 GOSUB500 :rem 78
10 GOSUB800:POKE650,128:SP=1:POKE204,0 :rem 234
:rem 132
20 GETA$:IFA$=""THEN20LIST10 :rem 132
21 IFA$=CHR$(20)ANDPEEK(210)=31ANDPEEK(20 9)>205THENA$=CHR$(157)+CHR$(32)+CHR$(1
57) :rem 58
22 IFA$=CHR$(34)THEN20 :rem 223
23 IFA$=CHR$(13)ANDPEEK(210)=31ANDPEEK(20 9)>205THEN20 :rem 19
25 IFA$="{CLR}"THENPOKE204,1:GOSUB800:POK 6204,0:A$="":GOTO20 :rem 98
30 IFASC(A$)<141ANDASC(A$)>132THENGOSUBL0 0 :rem 197
32 IFA$=""THEN20 :rem 107
35 IFASC(A$)=13AND(PEEK(210)=31)AND(PEEK( 209)>226)THEN20 :rem 145
40 POKE205,3:WAIT207,1:PRINTA$; :rem 85
50 IFPEEK(210)=31AND(PEEK(209)+PEEK(211)> 227)THENPOKE205,3:WAIT207,1:PRINT"UP}
"; :rem 113
60 GOTO20 :rem 1
682 PRINT"{4 DOWN}{5 RIGHT}{RVS}HIT ANY K 683 POKE198,0 :rem 14
EY" :rem 206
685 GETA$:IFA$=""THEN685 :rem 105
690 PRINT"{CLR}{3 DOWN}{3 RIGHT}ALL EDITI 695 PRINT"{4 RIGHT}WORK AS NORMAL."
NG KEYS" :rem 166 :rem 183
700 PRINT"{DOWN}{2 RIGHT}TEXT CANNOT SCRO 705 PRINT"{3 SPACES}PAST THE END OF"
LL" :rem 138 :rem 242
707 PRINT"{5 SPACES}THE SCREEN." :rem 62
710 PRINT"{DOWN}YOU MAY USE UPPERCASE/GRA 715 PRINT"{DOWN}{2 SPACES}THE PRINT ROUTI
PHICS OR UPPERCASE/LOWERCASE MODE." :rem 227
NE" :rem 19
730 PRINT"{2 RIGHT}WILL AUTOMATICALLY 745 PRINT"{2 RIGHT}CORRECTLY." :rem 144
{4 SPACES}SET THE PRINT MODE" :rem 53

```



```

750 PRINT"{2 DOWN} {RVS}HIT ANY KEY TO BE
    GIN{OFF}" :rem 241
770 GETA$:IFA$=" "THEN770 :rem 95
780 POKE198,0:RETURN :rem 230
800 PRINT"{CLR}{RVS}>{OFF}";:FORI=1TO5:PR
    INTSPC(79)">";:NEXTI :rem 182
805 PRINT:PRINT:PRINT:PRINT"{18 SPACES}EN
    D "; :rem 213
810 PRINT" [4 Y]{RVS}MEMO WRITER{OFF}
    [5 Y]{HOME}"; :rem 214
820 RETURN :rem 122
60000 REM :rem 218
60004 G1$=CHR$(145) :rem 191
60010 G1$=G1$+CHR$(TF) :rem 131
60020 OPEN4,4:WAIT207,1:POKE204,255
    :rem 234
60030 FORG0=0TORW:G0$=G1$:G1=G1+CL
    :rem 223
60040 FORG2=G1TOG1+(CL-1):G3=PEEK(G2)
    :rem 115
60050 IFG3>128THENG3=G3-128:G4=1:G0$=G0$+
    CHR$(18) :rem 187
60060 IF(G3>0)*(G3<32)THENG3=G3+64:GOTO60
    100 :rem 185
60070 IF(G3>31)*(G3<64)THEN60100 :rem 186
60080 IF(G3>63)*(G3<96)THENG3=G3+128:GOTO
    60100 :rem 47
60090 IF(G3>95)*(G3<128)THENG3=G3+64:GOTO
    60100 :rem 48
60100 G0$=G0$+CHR$(G3) :rem 97
60110 IFG4=1THENG0$=G0$+CHR$(146):G4=0
    :rem 76
60120 NEXTG2:PRINT#4,G0$:IFSP=2THENPRINT#
    4 :rem 132
60130 NEXTG0:CLOSE4:POKE204,0 :rem 239
60140 RETURN :rem 219

```

```

120 PRINT"DO YOU GO UP, GO DOWN":PRINT"OR
    STAY THE SAME":PRINT"TO PLAY THE SEC
    OND?" :rem 221
130 PRINT"{DOWN}PRESS{2 SPACES}F1 FOR UP"
    :PRINTTAB(7)"F3 FOR SAME":PRINTTAB(7)
    "F5 FOR DOWN" :rem 250
140 L$="*****" :rem 6
150 FOR I=0TO8:READF(I):NEXT :rem 189
160 DATA 232,231,228,225,223,219,215,209,
    207 :rem 114
170 B(1)=135:B(2)=134:B(3)=133 :rem 218
175 POKE 36878,15:S=36876 :rem 70
180 GOSUB20 :rem 123
190 SC=0:FOR T=1 TO 10 :rem 132
200 PRINT"{CLR}{4 DOWN}{BLK}":FOR I=1TO5:
    PRINTL$:NEXT :rem 42
210 N1=INT(9*RND(0)):P1=7796+N1*22:rem 96
220 POKE P1,81:POKE P1+30720,2 :rem 72
230 N2=INT(9*RND(0)):P2=7802+N2*22:rem 89
240 POKE P2,81:POKE P2+30720,2 :rem 76
250 A=SGN(N1-N2)+2:FL=0 :rem 16
260 POKE S,237:GOSUB 30 :rem 255
270 PRINT"{3 DOWN}{BLU}F1{2 SPACES}UP":PR
    INT"F3{2 SPACES}SAME":PRINT"F5
    {2 SPACES}DOWN" :rem 64
280 GET A$:IF A$=" "THEN 280 :rem 87
290 IF ASC(A$)<133 OR ASC(A$)>135 THEN 28
    0 :rem 88
300 IF ASC(A$)=B(A) THEN 350 :rem 135
310 FL=1:POKE S,159:GOSUB 30 :rem 56
320 POKE S,135:GOSUB 30:GOTO 280 :rem 6
350 POKE S,F(N1):GOSUB 40 :rem 122
360 POKE S,F(N2):GOSUB 40 :rem 124
370 IF FL=0 THEN SC=SC+1 :rem 28
380 NEXT T :rem 46
390 PRINT "{2 DOWN}SCORE = ";SC;"OUT OF 1
    0" :rem 134
400 PRINT "{DOWN}{GRN}TRY AGAIN? (Y/N)"
    :rem 89
410 GET A$:IF A$="Y" THEN 190 :rem 171
420 IF A$<>"N" THEN 410 :rem 90
430 PRINT"{CLR}{BLU}":END :rem 43

```

## The Beginner's Corner

(Article on page 74.)

### 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: Stepping Up Or Down (VIC Version)

```

5 REM STEPPING UP OR DOWN :rem 1
10 GOTO 100 :rem 43
20 PRINT"{DOWN}{GRN}PRESS RETURN{BLU}"
    :rem 238
22 GETA$:IFA$=" "THEN22 :rem 235
24 IF ASC(A$)<>13THEN22 :rem 243
26 RETURN :rem 72
30 FOR D=1 TO 80:NEXT D:POKE S,0:RETURN
    :rem 246
40 FOR D=1 TO 500:NEXT D:POKE S,0:RETURN
    :rem 36
100 PRINT"{CLR}{BLU}":PRINT"STEPPING UP O
    R DOWN" :rem 7
110 PRINT"{2 DOWN}TWO NOTES ARE SHOWN.":P
    RINT"FROM THE FIRST ONE," :rem 83

```

### Program 2: Stepping Up Or Down (64 Version)

```

5 REM STEPPING UP OR DOWN :rem 1
10 POKE53281,1:GOTO 100 :rem 244
20 PRINT"{DOWN}{GRN}PRESS RETURN{BLU}"
    :rem 238
22 GETA$:IFA$=" "THEN22 :rem 235
24 IF ASC(A$)<>13THEN22 :rem 243
26 RETURN :rem 72
30 POKE W,17:FOR D=1 TO 80:NEXT D:POKE W,
    0:RETURN :rem 78
40 POKE W,17:FOR D=1 TO 500:NEXT D:POKE W
    ,0:RETURN :rem 124
100 PRINT"{CLR}{BLU}":PRINTTAB(10)"STEPP
    ING UP OR DOWN" :rem 144
110 PRINT"{2 DOWN}YOU WILL SEE TWO NOTES.
    ":PRINT"FROM THE FIRST ONE, DO YOU MO
    VE UP," :rem 150
120 PRINT"MOVE DOWN, OR STAY THE SAME TO
    {SPACE}PLAY THE SECOND NOTE?":rem 211
130 PRINT"{DOWN}PRESS{2 SPACES}F1 FOR UP"
    :PRINTTAB(7)"F3 FOR SAME":PRINTTAB(7)
    "F5 FOR DOWN" :rem 250
140 L$="*****" :rem 134
150 FOR I=0TO8:READHF(I),LF(I):NEXT
    :rem 93

```



```

160 DATA 44,193,42,62,37,162,33,135,31,16
    5,28,49,25,30,22,96,21,31 :rem 173
170 B(1)=135:B(2)=134:B(3)=133 :rem 218
175 POKE 54296,15:V1=54273:V2=54272:W=542
    76:POKE 54277,64:POKE 54278,128 :rem 134
180 GOSUB20 :rem 123
190 SC=0:FOR T=1 TO 10 :rem 132
200 PRINT"[CLR]{5 DOWN}{BLK}":FOR I=1TO5:
    PRINTL$:NEXT :rem 59
210 N1=INT(9*RND(0)):P1=1280+N1*40:rem 78
220 POKE P1,81:POKE P1+54272,2 :rem 80
230 N2=INT(9*RND(0)):P2=1287+N2*40:rem 90
240 POKE P2,81:POKE P2+54272,2 :rem 84
250 A=SGN(N1-N2)+2:FL=0 :rem 16
260 POKE V1,112:POKE V2,199:GOSUB 30 :rem 235
270 PRINT"[3 DOWN]{BLU}F1{2 SPACES}UP":PR
    INT"F3{2 SPACES}SAME":PRINT"F5
    {2 SPACES}DOWN" :rem 64
280 GET A$:IF A$=""THEN 280 :rem 87
290 IF ASC(A$)<133 OR ASC(A$)>135 THEN 28
    0 :rem 88
300 IF ASC(A$)=B(A) THEN 350 :rem 135
310 FL=1:POKE V1,10:POKE V2,143:GOSUB 30
    :rem 227
320 POKE V1,8:POKE V2,97:GOSUB 30:GOTO 28
    0 :rem 102
350 POKE V1,HF(N1):POKE V2,LF(N1):GOSUB 4
    0 :rem 117
360 POKE V1,HF(N2):POKE V2,LF(N2):GOSUB 4
    0 :rem 120
370 IF FL=0 THEN SC=SC+1 :rem 28
380 NEXT T :rem 46
390 PRINT "{2 DOWN}SCORE = ";SC;"OUT OF 1
    0" :rem 134
400 PRINT "{DOWN}{GRN}TRY AGAIN? (Y/N)"
    :rem 89
410 GET A$:IF A$="Y" THEN 190 :rem 171
420 IF A$<>"N" THEN 410 :rem 90
430 PRINT"[CLR]{BLU}":END :rem 43

```

```

SUB500:NEXT:FORI=1TO1000:NEXT:XT%=30:G
    OTO65 :rem 179
55 POKEV,15:FORI=0TO5:POKES1,P%(I):FORJ=1
    TO50:NEXTJ:IFI=3THENFORJ=1TO75:NEXTJ
    :rem 30
60 NEXTI:FORI=1TO50:NEXTI:POKEV,0:rem 202
65 PB%=PB%+30-XT%:PA%=PA%+1:IFPA%<10THEN3
    0 :rem 143
70 PR%=PR%*10:S$=STR$(PR%):X1%=3-LEN(S$)*
    2:PRINT"[CLR]{BLK}":Y%=3:H%=0 :rem 210
75 FORM=2TOLEN(S$):X%=X1%+M*4:Z%=VAL(MID$
    (S$,M,1)):GOSUB500:NEXT:PB%=PB%*L%
    :rem 223
80 X%=X%+4:Z%=12:GOSUB500 :rem 166
85 PRINTTAB(204)"TIMED SCORE":S$=STR$(PB%
    ):X1%=5-LEN(S$)*2:Y%=12:H%=2 :rem 246
90 FORM=2TOLEN(S$):X%=X1%+M*4:Z%=VAL(MID$
    (S$,M,1)):GOSUB500:NEXT :rem 92
92 PRINTTAB(182)"HIT ANY KEY" :rem 181
94 GETA$:IFA$=""THEN94 :rem 253
96 GOTO25 :rem 15
100 FORI=0TO2:A%(I)=0:B%(I)=0:C%(I)=0:R%(
    I)=0:NEXT:R%(3)=0:C%(3)=0 :rem 97
110 FORI=0TOL%-1:A%(I)=INT(10*RND(1)):IFA
    %(I)<>0THENE%=I :rem 215
120 B%(I)=INT(10*RND(1)):IFB%(I)<>0THENF%
    =I :rem 146
130 W%=A%(I)+B%(I)+R%(I):IFW%<10THENC%(I)
    =W%:G%=I:NEXT:RETURN :rem 64
140 C%(I)=W%-10:R%(I+1)=1:NEXT:C%(I)=1:G%
    =I:RETURN :rem 92
200 POKE36879,24:PRINT"[CLR]":H%=0:Y%=2:F
    ORM=0TOE%:X%=Q%-M*4:Z%=A%(M):GOSUB500
    :NEXT :rem 41
210 POKE198,0:Y%=7:FORM=0TOF%:X%=Q%-M*4:Z
    %=B%(M):GOSUB500:NEXT :rem 218
220 X%=Q%-(F%+1)*4:Z%=10:GOSUB500:POKEV,1
    5:FORM=282-Q%TO269+Q% :rem 45
230 POKES%+M,160:POKES%+M,H%:POKES1,M-30:
    NEXT:POKEV,0 :rem 17
240 H%=6:Y%=14:Z%=11:FORM=0TOG%:X%=Q%-M*4
    :GOSUB500:NEXT:RETURN :rem 54
300 U%=0:H%=2:Y%=14:TI$="000000" :rem 134
310 PRINT"[HOME]{BLU}"TAB(220)TAB(220)"PR
    OBLEM{2 SPACES}TIME{3 SPACES}RIGHT":P
    RINTTAB(2):PA%+1:TAB(17):PR% :rem 76
320 FORM=0TOG% :rem 69
330 XT%=TI/60:IFXT%>30THEN RETURN :rem 32
340 POKES%+472,ASC(MID$(TI$,5,1))+128:POK
    ECS+472,0 :rem 17
350 POKES%+473,ASC(MID$(TI$,6,1))+128:POK
    ECS+473,0 :rem 21
360 GETA$:IFA$=""ORA$<"0"ORA$>"9"THEN330
    :rem 201
370 X%=Q%-M*4:Z%=VAL(A$):GOSUB500:U%=U%+Z
    %*10↑M:NEXT:RETURN :rem 94
400 FORI=0TO12:FORJ=0TO3:FORK=0TO3:READN%
    (I,J,K):NEXTK,J,I :rem 147
410 FORI=0TO5:READP%(I):NEXTI:RETURN
    :rem 75
500 POKEV,15:FORI=0TO3:FORJ=0TO3:K=T%*Y%+
    X%+T%*J+I:POKES%+K,N%(Z%,J,I):POKES+
    K,H% :rem 221
510 POKES1,255-X%*J-Y%*I:NEXTJ,I:POKEV,0:
    POKES1,0:RETURN :rem 3
1100 DATA108,226,226,123,225,32,32,97,225
    ,32,32,97,32,226,226,32 :rem 119
1101 DATA32,108,97,32,32,126,97,32,32,
    97,32,32,226,226,126 :rem 232
1102 DATA108,226,226,123,32,32,98,126,108
    ,226,32,32,124,226,226,126 :rem 10

```

## Fast Add

(Article on page 84.)

### Program 1: Fast Add—VIC Version

```

1 DIMA%(2),B%(2),C%(3),R%(3),P%(5),N%(12,
    3,3):T%=22:V=36878:S1=36876:I=RND(-TI)
    :rem 122
5 S%=4*(PEEK(36866)AND128)+64*(PEEK(36869)
    )AND112:CS=37888+4*(PEEK(36866)AND128)
    :rem 113
10 PRINT"[CLR]"TAB(74)"FASTADD":PRINTTAB(
    68)"1, 2, OR 3 DIGITS?":GOSUB400
    :rem 20
15 GETA$:IFA$=""ORA$<"1"ORA$>"3"THEN15
    :rem 97
20 L%=VAL(A$):Q%=9+L%*2 :rem 199
25 PB%=0:PR%=0:PA%=0 :rem 86
30 GOSUB100:W%=0:FORI=0TOG%:W%=W%+C%(I)*1
    0↑I:NEXT:GOSUB200:GOSUB300:FORI=1TO100
    0:NEXT :rem 18
35 IFU%=W%THENPR%=PR%+1:GOTO55 :rem 93
40 POKE36879,8:PRINT"[CLR]{WHT}"TAB(138);
    "W R O N G":PRINTTAB(49)"THE CORRECT"
    :rem 14
45 PRINTTAB(50)"ANSWER IS":FORI=1TO2000:N
    EXT:GOSUB200 :rem 53
50 Y%=14:FORM=0TOG%:X%=Q%-M*4:Z%=C%(M):GO

```



```

1103 DATA108,226,226,123,32,108,98,126,10
      8,32,32,97,32,226,226,32 :rem 175
1104 DATA32,32,254,32,32,255,225,32,124,2
      26,251,126,32,32,124,32 :rem 101
1105 DATA225,226,226,126,124,226,127,32,1
      08,32,108,126,32,226,126,32 :rem 56
1106 DATA32,255,226,32,225,98,98,32,225,3
      2,32,97,32,226,226,32 :rem 39
1107 DATA225,226,226,97,32,32,255,32,32,2
      55,32,32,32,126,32,32 :rem 18
1108 DATA108,226,226,123,124,98,98,126,22
      5,32,32,97,32,226,226,32 :rem 190
1109 DATA108,226,226,123,124,98,98,97,32,
      32,108,126,32,226,126,32 :rem 190
1110 DATA32,32,123,32,32,98,252,123,32,32
      ,97,32,32,32,32,32 :rem 110
1111 DATA160,160,160,97,160,160,160,97,16
      0,160,160,97,160,160,160,97 :rem 70
1112 DATA108,123,32,123,124,126,255,32,32
      ,255,108,123,124,32,124,126 :rem 42
1113 DATA215,225,231,235,231,235 :rem 77

```

## Program 2: Fast Add—64 Version

```

1 DIMA%(2),B%(2),C%(3),R%(3),P%(5),N%(12,
  3,3):T%=40:V=54296:S1=54272:I=RND(-TI)
      :rem 106
5 CS=55296:S%=1024 :rem 201
7 FORL=S1TOS1+24:POKES1,0:NEXTL:POKEV,15:
  POKES1+5,17:POKES1+6,241 :rem 52
10 PRINT"[CLR]"TAB(56)"FASTADD":PRINTTAB(
  91)"1, 2, OR 3 DIGITS?":GOSUB400
      :rem 16
15 GETA$:IFA$=""ORA$<"1"ORA$>"3"THEN15
      :rem 97
20 L%=VAL(A$):Q%=9+L%*2 :rem 199
25 PB%=0:PR%=0:PA%=0 :rem 86
30 GOSUB100:W%=0:FORI=0TOG%:W%=W%+C%(I)*1
  0↑I:NEXT:GOSUB200:GOSUB300 :rem 93
32 FORI=1TO1000:NEXT :rem 224
35 IFU%=W%THENPR%=PR%+1:GOTO55 :rem 93
40 POKE53281,0:PRINT"[CLR]WHT]"TAB(136);
  "W R O N G":PRINTTAB(55)"THE CORRECT"
      :rem 243
45 PRINTTAB(56)"ANSWER IS":FORI=1TO2000:N
  EXT:GOSUB200 :rem 59
50 Y%=14:FORM=0TOG%:X%=Q%-M*4+8:Z%=C%(M):
  GOSUB500:NEXT:FORI=1TO1000 :rem 20
53 NEXT:XT%=30:GOTO65 :rem 48
55 FORI=1TO6:POKES1+4,33:POKES1+1,PI(I):F
  ORA=1TODU(I):NEXT:POKES1+4,32 :rem 99
60 FORJ=1TO50:NEXT:NEXT :rem 255
65 PB%=PB%+30-XT%:PA%=PA%+1:IFPA%<10THEN3
  0 :rem 143
70 PR%=PR%*10:S$=STR$(PR%):X1%=3-LEN(S$)*
  2:PRINT"[CLR][BLK]":Y%=3:H%=0 :rem 210
75 FORM=2TOLLEN(S$):X%=X1%+M*4+8:Z%=VAL(MI
  D$(S$,M,1)):GOSUB500:NEXT:PB%=PB%*L%
      :rem 66
80 X%=X%+4:Z%=12:GOSUB500 :rem 166
85 PRINTTAB(13)"{9 DOWN} TIMED SCORE":S$=
  STR$(PB%):X1%=5-LEN(S$)*2:Y%=12:H%=2
      :rem 93
90 FORM=2TOLLEN(S$):X%=X1%+M*4+8:Z%=VAL(MI
  D$(S$,M,1)):GOSUB500:NEXT :rem 191
92 PRINTTAB(13)"{8 DOWN} HIT ANY KEY"
      :rem 6
94 GETA$:IFA$=""THEN94 :rem 253
96 GOTO25 :rem 15
100 FORI=0TO2:A%(I)=0:B%(I)=0:C%(I)=0:R%(
  I)=0:NEXT:R%(3)=0:C%(3)=0 :rem 97

```

```

110 FORI=0TOL%-1:A%(I)=INT(10*RND(1)):IFA
  %(I)<>0THENB%=I :rem 215
120 B%(I)=INT(10*RND(1)):IFB%(I)<>0THENF%
  =I :rem 146
130 W%=A%(I)+B%(I)+R%(I):IFW%<10THENC%(I)
  =W%:G%=I:NEXT:RETURN :rem 64
140 C%(I)=W%-10:R%(I+1)=1:NEXT:C%(I)=1:G%
  =I:RETURN :rem 92
200 POKE53281,1:POKE53280,0:PRINT"[CLR]":
  H%=0:Y%=2:FORM=0TOE%:X%=Q%-M*4+8:Z%=A
  %(M) :rem 72
205 GOSUB 500:NEXT :rem 37
210 POKE198,0:Y%=7:FORM=0TOF%:X%=Q%-M*4+8
  :Z%=B%(M):GOSUB500:NEXT :rem 61
220 X%=Q%-(F%+1)*4+8:Z%=10:GOSUB500:FORM=
  466-Q%TO453+Q%-1 :rem 156
230 POKES%+M,160:POKECS+M,H%:FORA=1TO5:PO
  KES1+4,33:POKES1+1,M-430 :rem 147
235 NEXT:NEXT:POKES1+4,32 :rem 47
240 H%=6:Y%=14:Z%=11:FORM=0TOG%:X%=Q%-M*4
  +8:GOSUB500:NEXT:RETURN :rem 153
300 U%=0:H%=2:Y%=14:TI$=""000000 :rem 134
310 PRINT"[HOME][BLU]{19 DOWN} PROBLEM";T
  AB(18)"TIME";TAB(30)"RIGHT" :rem 178
315 PRINTTAB(3):PA%+1:TAB(31)PR% :rem 92
320 FORM=0TOG% :rem 69
330 XT%=TI/60:IFXT%>30THEN RETURN :rem 32
340 POKES%+819,ASC(MID$(TI$,5,1))+128:POK
  ECS+819,0 :rem 27
350 POKES%+820,ASC(MID$(TI$,6,1))+128:POK
  ECS+820,0 :rem 13
360 GETA$:IFA$=""ORA$<"0"ORA$>"9"THEN330
      :rem 201
370 X%=Q%-M*4+8:Z%=VAL(A$):GOSUB500:U%=U%
  +Z%*10↑M:NEXT:RETURN :rem 193
400 FORI=0TO12:FORJ=0TO3:FORK=0TO3:READN%
  (I,J,K):NEXTK,J,I :rem 147
410 FORI=0TO5:READP%(I):NEXTI:FORQ=1TO6:R
  EADPI(Q),DU(Q):NEXTQ:RETURN :rem 198
500 POKES1+4,33:FORI=0TO3:FORJ=0TO3:K=T%*
  Y%+X%+T%*J+I :rem 171
505 POKES%+K,N%(Z%,J,I):POKECS+K,H%:POKES
  1+1,125-X%*J-Y%*I:POKES1,30 :rem 228
510 NEXTJ,I:POKES1+4,32:RETURN :rem 139
1100 DATA108,226,226,123,225,32,32,97,225
      ,32,32,97,32,226,226,32 :rem 119
1101 DATA32,108,97,32,32,126,97,32,32,32,
      97,32,32,226,226,126 :rem 232
1102 DATA108,226,226,123,32,32,98,126,108
      ,226,32,32,124,226,226,126 :rem 10
1103 DATA108,226,226,123,32,108,98,126,10
      8,32,32,97,32,226,226,32 :rem 175
1104 DATA32,32,254,32,32,255,225,32,124,2
      26,251,126,32,32,124,32 :rem 101
1105 DATA225,226,226,126,124,226,127,32,1
      08,32,108,126,32,226,126,32 :rem 56
1106 DATA32,255,226,32,225,98,98,32,225,3
      2,32,97,32,226,226,32 :rem 39
1107 DATA225,226,226,97,32,32,255,32,32,2
      55,32,32,32,126,32,32 :rem 18
1108 DATA108,226,226,123,124,98,98,126,22
      5,32,32,97,32,226,226,32 :rem 190
1109 DATA108,226,226,123,124,98,98,97,32,
      32,108,126,32,226,126,32 :rem 190
1110 DATA32,32,123,32,32,98,252,123,32,32
      ,97,32,32,32,32,32 :rem 110
1111 DATA160,160,160,97,160,160,160,97,16
      0,160,160,97,160,160,160,97 :rem 70
1112 DATA108,123,32,123,124,126,255,32,32
      ,255,108,123,124,32,124,126 :rem 42
1113 DATA215,225,231,235,231,235 :rem 77

```



```

1114 DATA 25,50,33,50,42,50,50,250,42,50,
      50,300 :rem 230
19999 S1=54272 :rem 218
20000 V=S1+24:FORL=S1TOS1+24:POKES1,0:NEX
      TL:POKEV,15:POKES1+5,17:POKES1+6,24
      1 :rem 209
20010 POKES1+4,33:POKES1+1,25:FORI=1TO50:
      NEXTI:POKES1+4,32:FORI=1TO50:NEXT
      :rem 75
20020 POKES1+1,33:POKES1+4,33:FORA=1TO50:
      NEXT:POKES1+4,32:FORI=1TO50 :NEXT
      :rem 250
20025 POKES1+4,33:POKES1+1,42:FORI=1TO50:
      NEXT:POKES1+4,32:FORI=1TO50:NEXT
      :rem 7
20028 POKES1+4,33:POKES1+1,50:FORI=1TO250
      :NEXT:POKES1+4,32:FORI=1TO50:NEXT
      :rem 59
20030 POKES1+4,33:POKES1+1,42:FORI=1TO50:
      NEXTI:POKES1+4,32:FORI=1TO50:NEXTI
      :rem 149
20040 POKES1+4,33:POKES1+1,50:FORI=1TO300
      :NEXT:POKES1+4,32 :rem 216
20050 POKES1+4,33:POKES1+1,66:FORI=1TO300
      :NEXT:POKES1+4,32 :rem 224

```

## Power BASIC

(Article on page 95.)

### Program 1: VIC Step Lister

```

10 TM = PEEK(56) * 256 + PEEK(55):rem 242
15 TM = TM - 242 :rem 170
20 LSB = (TM/256-INT(TM/256)) * 256 :rem 219
25 MSB = INT(TM/256) :rem 47
30 POKE 55,LSB: POKE 56,MSB :rem 235
35 POKE 253,LSB+13: POKE 254,MSB :rem 223
40 FOR I = TM TO TM + 241 :rem 120
50 READ A: POKE I,A: CHK = CHK + A: NEXT
   {SPACE}I :rem 50
60 X = 828: FOR I = X TO X + 23 :rem 39
70 READ A: POKE I,A: CHK = CHK + A: NEXT
   {SPACE}I :rem 52
80 IF CHK <> 33283 THEN PRINT "DATA ERROR
   ": END :rem 255
90 PRINT "SYS" TM "TO INITIATE": NEW
   :rem 60
100 DATA162,0,189,60,3,149,115,232,224,23
   ,208,246,0,201,64,240,22,234 :rem 90
120 DATA234,234,234,201,58,176,10,201,32,
   240,7,56,233,48,56,233,208,96:rem 163
140 DATA76,115,0,160,0,185,0,2,201,64,208
   ,243,200,185,0,2,201,0,240,9 :rem 66
160 DATA201,45,208,244,169,171,153,0,2,16
   9,1,133,122,160,1,24,185,0,2 :rem 88
180 DATA32,107,201,32,19,198,160,0,32,121
   ,0,32,107,201,165,20,5,21,208:rem 124
200 DATA6,169,255,133,20,133,21,160,1,132
   ,198,160,1,132,15,177,95,240 :rem 100
210 DATA85,32,44,200,32,215,202,134,1,132
   ,2,173,198,0,240,251,169,0 :rem 242
240 DATA141,198,0,166,1,164,2,200,177,95,
   170,200,177,95,197,21,208,4,228,20
   :rem 157
250 DATA240,2,176,44,132,73,32,205,221,16
   9,32,164,73,41,127,32,71,203 :rem 103
270 DATA201,34,208,6,165,15,73,255,133,15
   ,200,240,17,177,95,208,16 :rem 224
290 DATA168,177,95,170,200,177,95,134,95,

```

```

133,96,208,163,76,116,196,16 :rem 164
310 DATA218,201,255,240,214,36,15,48,210,
   56,233,127,170,132,73,160,255,202
   :rem 91
320 DATA240,8,200,185,158,192,16,250,48,2
   45,200,185,158,192,48,181,32 :rem 129
350 DATA71,203,208,245,0,230,122,208,2,23
   0,123,173,0,2,201,58,240,10 :rem 21
360 DATA201,32,240,239,108,253,0 :rem 74
370 DATA234,234,234,96 :rem 114

```

### Program 2: 64 Step Lister

```

10 TM=49152 :rem 68
20 FOR I = TM TO TM + 241 :rem 118
30 READ A: POKE I,A: CHK = CHK + A: NEXT
   {SPACE}I :rem 48
40 X = 828: FOR I = X TO X + 23 :rem 37
50 READ A: POKE I,A: CHK = CHK + A: NEXT
   {SPACE}I :rem 50
60 IF CHK <> 32456 THEN PRINT "DATA ERROR
   ": END :rem 254
70 SYS49152 :rem 107
100 DATA162,0,189,60,3,149,115,232,224,23
   ,208,246,0,201,64,240,22,201 :rem 84
120 DATA58,176,10,201,32,240,11,56,233,48
   ,56,233,208,96,76,116,164,234:rem 169
140 DATA76,115,0,160,0,185,0,2,201,64,208
   ,243,200,185,0,2,201,0,240,9,201
   :rem 1
160 DATA45,208,244,169,171,153,0,2,169,1,
   133,122,160,1,24,185,0,2,32,107
   :rem 238
180 DATA169,32,19,166,160,0,32,121,0,32,1
   07,169,165,20,5,21,208,6,169 :rem 106
200 DATA255,133,20,133,21,160,1,132,198,1
   60,1,132,15,177,95,240,175,32:rem 144
220 DATA44,168,32,215,170,134,25,132,26,1
   73,198,0,240,251,169,0,141,198
   :rem 214
230 DATA0,166,25,164,26,200,177,95,170,20
   0,177,95,197,21,208,4,228,20 :rem 120
250 DATA240,2,176,44,132,73,32,205,189,16
   9,32,164,73,41,127,32,71,171 :rem 120
270 DATA201,34,208,6,165,15,73,255,133,15
   ,200,240,17,177,95,208,16,168:rem 171
290 DATA177,95,170,200,177,95,134,95,133,
   96,208,163,108,6,3,16,218,201:rem 188
310 DATA255,240,214,36,15,48,210,56,233,1
   27,170,132,73,160,255,202,240,8
   :rem 251
320 DATA200,185,158,160,16,250,48,245,200
   ,185,158,160,48,181,32,71,171,208
   :rem 112
350 DATA245,0,230,122,208,2,230,123,173,0
   ,2,201,58,240,10,201,32,240,239
   :rem 214
370 DATA76,13,192,234,234,234,96 :rem 99

```

## Cassette Beeper

(Article on page 102.)

### Program 1: Cassette Beeper—VIC Version

```

5 PRINT CHR$(147)"{3 DOWN}{3 SPACES}CASSE
   TTE BEEPER":PRINT :rem 253
10 EM=PEEK(56)*256+PEEK(55)-60 :rem 118
15 PT=EM:GOSUB100 :rem 36

```



```

20 POKE51, LB: POKE55, LB :rem 62
25 POKE52, HB: POKE56, HB :rem 61
30 PRINT: PRINT "SYS "EM: PRINT "(TO RESET PO
INTERS)" :rem 130
35 PT=PT+21 :rem 123
40 FORX=1 TO 7 STEP 2: GOSUB 100: B(X)=LB: B(X+1)
=HB :rem 220
45 PT=PT+3: NEXT X :rem 29
50 FORX=EMTOEM+57 :rem 63
55 READN: IFN<0 THEN N=B(N*-1) :rem 122
60 POKE X, N: NEXT X: PRINT: PRINT :rem 198
65 SYS EM: SYSB(3)+B(4)*256: NEW :rem 216
75 DATA 169, -1141, 48, 3, 169, -2, 141, 49, 3, 1
69, -5, 141, 50, 3, 169, -6 :rem 225
80 DATA 141, 51, 3, 96, 32, 73, 245, 76, -7, -8, 32
, 133, 246, 8, 72, 9, 15 :rem 191
85 DATA 141, 14, 144, 169, 240, 141, 12, 144, 165
, 162, 105, 250, 197, 162 :rem 187
90 DATA 48, 252, 169, 0, 141, 12, 144, 104, 40, 96
:rem 222
100 HB=INT(PT/256): LB=PT - HB*256: RETURN
:rem 69

```

## Program 2: Cassette Beeper—64 Version

```

10 I=49152 :rem 236
20 READ A: IF A=256 THEN END :rem 169
30 POKE I, A: I=I+1: GOTO 20 :rem 130
49152 DATA 169, 21, 141, 48, 3, 169, 192
:rem 155
49160 DATA 141, 49, 3, 169, 27, 141, 50 :rem 96
49168 DATA 3, 169, 192, 141, 51, 3, 96 :rem 59
49176 DATA 32, 165, 244, 76, 30, 192, 32
:rem 153
49184 DATA 237, 245, 8, 72, 169, 0, 160 :rem 107
49192 DATA 24, 153, 0, 212, 136, 208, 250
:rem 186
49200 DATA 169, 15, 141, 24, 212, 169, 85
:rem 197
49208 DATA 141, 5, 212, 169, 170, 141, 6
:rem 141
49216 DATA 212, 169, 100, 141, 0, 212, 141
:rem 223
49224 DATA 1, 212, 169, 17, 141, 4, 212 :rem 84
49232 DATA 169, 16, 141, 4, 212, 104, 40
:rem 133
49240 DATA 96, 256 :rem 85

```

# Machine Language For Beginners

(Article on page 104.)

### 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 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<>38268 THEN PRINT "ERROR IN DATA
STATEMENTS": STOP :rem 202
12288 DATA 160, 0, 169, 8, 153, 0 :rem 99
12294 DATA 148, 153, 0, 149, 200, 208 :rem 40
12300 DATA 247, 160, 0, 169, 224, 153 :rem 33
12306 DATA 0, 16, 153, 228, 17, 200 :rem 184
12312 DATA 192, 22, 208, 245, 169, 21 :rem 39
12318 DATA 133, 71, 169, 16, 133, 72 :rem 251
12324 DATA 162, 24, 160, 0, 169, 224 :rem 240
12330 DATA 145, 71, 200, 145, 71, 202 :rem 25
12336 DATA 240, 16, 24, 165, 71, 105 :rem 242
12342 DATA 22, 133, 71, 165, 72, 105 :rem 240
12348 DATA 0, 133, 72, 76, 38, 48 :rem 108
12354 DATA 169, 20, 133, 204, 32, 155 :rem 36
12360 DATA 224, 164, 98, 185, 149, 15 :rem 56
12366 DATA 201, 224, 240, 244, 169, 90 :rem 92
12372 DATA 153, 149, 15, 198, 204, 208 :rem 100
12378 DATA 235, 169, 214, 133, 251, 169
:rem 155
12384 DATA 17, 133, 252, 32, 193, 48 :rem 254
12390 DATA 162, 10, 32, 203, 48, 165 :rem 241
12396 DATA 197, 201, 0, 240, 14, 201 :rem 237
12402 DATA 1, 240, 25, 201, 53, 240 :rem 174
12408 DATA 91, 201, 60, 240, 86, 208 :rem 246
12414 DATA 231, 198, 251, 160, 0, 177 :rem 40
12420 DATA 251, 201, 32, 240, 16, 230 :rem 17
12426 DATA 251, 76, 102, 48, 160, 2 :rem 194
12432 DATA 177, 251, 201, 32, 240, 25 :rem 30
12438 DATA 76, 102, 48, 230, 251, 160 :rem 40
12444 DATA 1, 169, 32, 145, 251, 165 :rem 249
12450 DATA 251, 208, 2, 198, 252, 198 :rem 51
12456 DATA 251, 32, 193, 48, 76, 102 :rem 254
12462 DATA 48, 160, 1, 169, 32, 145 :rem 200
12468 DATA 251, 230, 251, 208, 2, 230 :rem 33
12474 DATA 252, 32, 193, 48, 76, 102 :rem 255
12480 DATA 48, 160, 1, 169, 120, 145 :rem 246
12486 DATA 251, 136, 208, 251, 96, 160 :rem 101
12492 DATA 0, 136, 208, 253, 202, 208 :rem 36
12498 DATA 248, 96, 165, 251, 133, 253 :rem 112
12504 DATA 165, 252, 133, 254, 32, 9 :rem 247
12510 DATA 49, 160, 0, 165, 193, 145 :rem 246
12516 DATA 253, 162, 10, 32, 203, 48 :rem 239
12522 DATA 169, 32, 145, 253, 32, 9 :rem 201
12528 DATA 49, 177, 253, 201, 32, 208 :rem 48
12534 DATA 7, 169, 193, 145, 253, 76 :rem 10
12540 DATA 229, 48, 201, 224, 240, 4 :rem 240
12546 DATA 201, 90, 240, 223, 76, 102 :rem 33
12552 DATA 48, 56, 165, 253, 233, 22 :rem 254
12558 DATA 133, 253, 165, 254, 233, 0 :rem 43
12564 DATA 133, 254, 96, 256 :rem 222

```

## Program 3: 64 Version

```

10 I=49152 :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<>41231 THEN PRINT "ERROR IN DATA S
TATEMENTS": STOP :rem 186
49152 DATA 160, 0, 169, 8, 153, 0 :rem 99
49158 DATA 216, 153, 0, 217, 153, 0 :rem 198
49164 DATA 218, 153, 0, 219, 200, 208 :rem 42
49170 DATA 241, 160, 0, 169, 224, 153 :rem 42
49176 DATA 0, 4, 153, 192, 7, 200 :rem 99
49182 DATA 192, 40, 208, 245, 169, 39 :rem 63
49188 DATA 133, 71, 169, 4, 133, 72 :rem 215
49194 DATA 162, 24, 160, 0, 169, 224 :rem 255
49200 DATA 145, 71, 200, 145, 71, 202 :rem 31
49206 DATA 240, 16, 24, 165, 71, 105 :rem 248
49212 DATA 40, 133, 71, 165, 72, 105 :rem 246
49218 DATA 0, 133, 72, 76, 44, 192 :rem 159

```



```

49224 DATA 169,20,133,204,32,158 :rem 45
49230 DATA 224,164,98,185,168,3 :rem 12
49236 DATA 201,224,240,244,169,90 :rem 98
49242 DATA 153,168,3,198,204,208 :rem 56
49248 DATA 235,169,169,133,251,169 :rem 170
49254 DATA 7,133,252,32,199,192 :rem 9
49260 DATA 162,5,32,209,192,165 :rem 1
49266 DATA 197,201,56,240,14,201 :rem 46
49272 DATA 8,240,25,201,46,240 :rem 198
49278 DATA 91,201,35,240,86,208 :rem 7
49284 DATA 231,198,251,160,0,177 :rem 55
49290 DATA 251,201,32,240,16,230 :rem 32
49296 DATA 251,76,108,192,160,2 :rem 7
49302 DATA 177,251,201,32,240,25 :rem 36
49308 DATA 76,108,192,230,251,160 :rem 100
49314 DATA 1,169,32,145,251,165 :rem 255
49320 DATA 251,208,2,198,252,198 :rem 57
49326 DATA 251,32,199,192,76,108 :rem 64
49332 DATA 192,160,1,169,32,145 :rem 254
49338 DATA 251,230,251,208,2,230 :rem 39
49344 DATA 252,32,199,192,76,108 :rem 65
49350 DATA 192,160,1,169,120,145 :rem 44
49356 DATA 251,136,208,251,96,160 :rem 107
49362 DATA 0,136,208,253,202,208 :rem 42
49368 DATA 248,96,165,251,133,253 :rem 118
49374 DATA 165,252,133,254,32,15 :rem 51
49380 DATA 193,160,0,165,193,145 :rem 53
49386 DATA 253,162,10,32,209,192 :rem 52
49392 DATA 169,32,145,253,32,15 :rem 5
49398 DATA 193,177,253,201,32,208 :rem 111
49404 DATA 7,169,193,145,253,76 :rem 16
49410 DATA 235,192,201,224,240,4 :rem 35
49416 DATA 201,90,240,223,76,108 :rem 45
49422 DATA 192,56,165,253,233,40 :rem 52
49428 DATA 133,253,165,254,233,0 :rem 49
49434 DATA 133,254,96,256 :rem 228

```

## Sound Story

(Article on page 116.)

```

100 POKE 36879,108:V=36878 :rem 115
110 PRINT"[CLR]":FORI=1TO7:PRINT:NEXT:PRI
NTTAB(6)"[WHT]{2 DOWN}SOUND STORY"
:rem 162
120 FORL=1TO3000:NEXT :rem 19
130 PRINT"[CLR]":POKE36879,8:FORI=1TO9:PR
INT:NEXT :rem 202
140 PRINTTAB(2)"[CYN]IT WAS A DARK AND
{5 SPACES}":PRINTTAB(4)"STORMY NIGHT.
.." :rem 118
150 POKEV,15:S=36874:FORL=1TO2000:NEXT
:rem 60
160 POKES,195:FORL=1TO600:NEXT:POKES,201:
FORL=1TO200:NEXT :rem 117
170 POKES,203:FORL=1TO300:NEXT:POKES,0:FO
RL=1TO100:NEXT :rem 5
180 POKES,195:FORL=1TO200:NEXT:POKES,0:FO
RL=1TO200:NEXT :rem 16
190 FORT=1TO12:POKES,211:FORL=1TO50:NEXT:
POKES,213:FORL=1TO50:NEXT:NEXT:POKES,
0 :rem 140
200 PRINT"[CLR]":FORL=1TO1000:NEXT:POKE36
879,25:S=36877 :rem 148
210 FORL=250TO2000STEP-1:POKES,L:NEXT:POKE
S,0:POKE36879,8:FORL=1TO100:NEXT
:rem 148
220 POKE36879,25:FORL=250TO2000STEP-1:POKE
S,L:NEXT:POKES,0:POKE36879,8 :rem 26

```

```

230 FORL=15TO4STEP-.04:POKEV,L:POKES,130:
NEXT :rem 105
240 POKE36878,15:POKE36879,25:FORL=250TO1
80STEP-1:POKES,L:NEXT:POKES,0:POKE368
79,8 :rem 46
250 FORL=15TO1STEP-.03:POKEV,L:POKES,128:
NEXT :rem 110
260 FORI=1TO6:PRINT:NEXT:PRINT"{3 SPACES}
SUDDENLY, OFF IN":PRINT:PRINT" THE DI
STANCE, A TRAIN" :rem 73
270 PRINTTAB(7)"WENT BY..." :rem 44
280 FORL=1TO4000:NEXT :rem 27
290 PRINT"[CLR]":L=1 :rem 243
300 FORD=200TO15STEP-5:L=L+.3:POKEV,L
:rem 6
310 POKES,200:FORT=1TO60:NEXTT:POKES,0:FO
RT=1TOD:NEXTT:NEXTD :rem 249
320 FORL=15TO0STEP-.3:POKEV,L:POKES,200:F
ORT=1TO40:NEXTT:POKES,0:FORT=1TO40:NE
XTT:NEXTL :rem 4
330 FORI=1TO8:PRINT:NEXT:PRINTTAB(4)"THEN
THE TRAIN":PRINT:PRINTTAB(3)"ENGINEE
R HEARD " :rem 34
340 PRINTTAB(18)"[UP]A":PRINT:PRINTTAB(4)
"FLYING SAUCER!" :rem 166
350 FORL=1TO4000:NEXT :rem 25
360 PRINT"[CLR]":C=36879:POKEC,138:POKEV,
15:S=36874:GOSUB430 :rem 180
370 POKEC,110:GOSUB440 :rem 28
380 POKEC,25:GOSUB430 :rem 241
390 POKEC,93:GOSUB440 :rem 248
400 POKEC,127:GOSUB430 :rem 29
410 POKEC,76:GOSUB440 :rem 242
420 POKEC,8:POKES,0:GOTO450 :rem 144
430 FORL=180TO254:POKES,L:FORM=1TO5:NEXTM
:POKES,0:NEXTL:RETURN :rem 131
440 FORL=254TO180STEP-1:POKES,L:FORM=1TO5
:NEXTM:POKES,0:NEXTL:RETURN :rem 30
450 PRINT"[CLR]":FORI=1TO8:PRINT:NEXT:PRI
NT"{3 SPACES}A SECRET MESSAGE" :rem 7
460 PRINT:PRINTTAB(3)"WAS SENT..." :rem 60
470 FORM=1TO30:POKEV,15:POKE36876,200:FOR
L=1TO(INT(RND(1)*80)):NEXT :rem 69
480 POKE 36876,0:FORL=1TO(INT(RND(1)*80))
:NEXT:NEXT :rem 42
490 PRINT"[CLR]":FORI=1TO9:PRINT:NEXT:PRI
NTTAB(5)"A PING PONG" :rem 0
500 PRINT:PRINTTAB(5)"BALL DROPPED!"
:rem 212
510 POKEV,15:S=36874 :rem 131
520 FORD=200TO0STEP-5:POKES,250:FORT=1TO2
:NEXTT:POKES,0:FORT=1TOD:NEXTT:NEXTD
:rem 114
530 PRINT"[CLR]":FORI=1TO7:PRINT:NEXT:PRI
NTTAB(3)"THE STORM ENDED," :rem 86
540 PRINT:PRINTTAB(2)"THE STARS CAME OUT,
" :rem 51
550 PRINT:PRINTTAB(2)" AND THE CRICKETS":
PRINT:PRINTTAB(2)"STARTED CHIRPING..."
:rem 185
560 FORL=1TO5000:NEXT :rem 29
570 PRINT"[CLR]":POKE36879,104:FORL=1TO20
:rem 27
580 X=INT(RND(1)*506) :rem 153
590 POKE7680+X,46:POKE38400+X,7:NEXT
:rem 130
600 S=36876:FORL=15TO1STEP-.5:POKEV,L
:rem 27
610 FORT=1TO5:POKES,243:FORT=1TO3:NEXTT:P
OKES,0:FORT=1TO8:NEXTT :rem 188
620 NEXTN:FORT=1TO100:NEXTT:NEXTL:rem 206

```



```

630 FOR I=1 TO 10:PRINT:PRINT TAB(8)"
    {YEL}THE END{BLU}" :rem 91
640 FOR L=1 TO 3000:NEXT:PRINT "{CLR}":POKE 36
    879,27:END :rem 216

```

# Joystick Control For The 64

(Article on page 118.)

## 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.

## Chase And Tag

```

15 REM{2 SPACES}*{8 SPACES}CHASE AND TAG
    {8 SPACES}* :rem 177
40 POKE 53280,4 :POKE 53281,14 :PRINT "
    {WHT}" :CLR:CO=54272:GOTO 700 :rem 52
50 REM *** INDIRECT ADDRESS REGISTERS ***
    :rem 85
55 POKE 52254,1 :POKE 52246,2 :POKE 52247
    ,42 :POKE 52245,82 :POKE 52253,81
    :rem 193
60 POKE 52249,80 :POKE 52251,40 :POKE 522
    50,0 :POKE 52255,41 :rem 231
65 POKE 52126,1 :POKE 52118,2 :POKE 52119
    ,42 :POKE 52117,82 :POKE 52125,81
    :rem 184
70 POKE 52121,80 :POKE 52123,40 :POKE 521
    22,0 :POKE 52127,41 :rem 215
100 REM *** PLAYER MOVEMENT WITH JOYSTICK
    *** :rem 85
110 R = 1422 :L = 1385 :P = 0 :rem 117
120 POKE R,81:POKER+CO,0:POKE L,102:POKEL
    +CO,0 :rem 191
125 P = P+1 :rem 209
130 IF P = 100 THEN GOTO 310 :rem 61
140 X = PEEK(56321)+52000 :rem 195
145 IF X < 52245 THEN GOTO 140 :rem 188
150 M = PEEK(X)-41 :rem 128
160 Y = PEEK(56320)+52000 :rem 197
165 IF Y < 52117 THEN GOTO 160 :rem 191
170 N = PEEK(Y)-41 :rem 132
180 C = (R+M - 1063) / 40 :IF C - INT(C)
    {SPACE}= 0 THEN M = 0 :GOTO 210
    :rem 210
190 G = (R+M - 1064) / 40 :IF G - INT(G)
    {SPACE}= 0 THEN M = 0 :GOTO 210
    :rem 224
200 IF R+M < 1063 OR R+M > 1822 THEN M =
    {SPACE}0 :rem 80
210 POKE R,32:POKER+CO,0:R = R+M :rem 12
220 POKE R,81:POKER+CO,0 :rem 126
225 IF T = 1 AND R = L THEN GOTO 310
    :rem 148
230 H = (L+N - 1063) / 40 :IF H - INT(H)
    {SPACE}= 0 THEN N = 0 :GOTO 260
    :rem 222
240 B = (L+N - 1064) / 40 :IF B - INT(B)
    {SPACE}= 0 THEN N = 0 :GOTO 260
    :rem 206
250 IF L+N < 1063 OR L+N > 1822 THEN N =

```

```

    {SPACE}0 :rem 76
260 POKE L,32 :POKEL+CO,0:L = L+N:rem 250
270 POKE L,102:POKEL+CO,0 :rem 161
275 IF T = 0 AND R = L THEN GOTO 310
    :rem 152
280 IF P = 80 THEN POKE 53281,12 :rem 14
290 GOTO 125 :rem 108
300 REM *** THE EXPLOSION *** :rem 21
310 FOR B = 1 TO 6 :rem 4
320 FOR A = 0 TO 15 :POKE 53281,A :NEXT A
    :rem 198
330 NEXT B :rem 23
340 POKE L,32:POKEL+CO,0:POKE R,32:POKER+
    CO,0 :rem 145
400 REM *** RESETTING THE PLAYING FIELD A
    ND KEEPING SCORE *** :rem 212
410 POKE 53281,1 :POKE 1422,81:POKE 1422+
    CO,0:POKE 1385,102:POKEL385+CO,0
    :rem 130
420 IF T = 0 AND P = 100 THEN GOTO 610
    :rem 214
430 IF T = 1 AND P < 100 THEN GOTO 610
    :rem 215
500 REM *** UPDATING THE LEFT PLAYERS SCO
    RE *** :rem 121
510 E = E+1 :rem 185
520 IF E = 58 AND F = 49 THEN GOTO 1300
    :rem 4
530 IF E = 58 THEN F = 49 :POKE 1945,F:PO
    KEL945+CO,0 :rem 56
540 IF E = 58 THEN E = 48 :rem 52
550 POKE 1976,9:POKEL976+CO,0:POKE 1977,2
    0:POKEL977+CO,0 :rem 135
560 POKE 1946,E:POKEL946+CO,0:POKE 1949,3
    2:POKEL949+CO,0 :rem 143
565 POKE 1950,32:POKEL950+CO,0:T = 1 :GOT
    O 110 :rem 129
600 REM *** UPDATING THE RIGHT PLAYERS SC
    ORE *** :rem 205
610 D = D+1 :rem 184
620 IF D = 58 AND QG = 49 THEN GOTO 1300
    :rem 86
630 IF D = 58 THEN QG = 49 :POKE 1980,QG:
    POKEL980+CO,0 :rem 218
640 IF D = 58 THEN D = 48 :rem 51
650 POKE 1949,9:POKEL949+CO,0:POKE 1950,2
    0:POKEL950+CO,0 :rem 118
660 POKE 1981,D:POKEL981+CO,0:POKE 1976,3
    2 :rem 52
665 POKE 1976,32 :POKEL976+CO,0:POKE 1977
    ,32 :POKEL977+CO,0:T=0:GOTO 110
    :rem 189
700 REM *** ENTERING THE PLAYERS INITIALS
    *** :rem 49
710 POKE 53281,0:PRINT "{CLR}":POKE 53281,1
    :rem 141
720 PRINT "{BLK}{3 RIGHT}ENTER THE LEFT P
    LAYERS 3 INITIALS{5 DOWN}" :rem 224
725 PRINT "{17 RIGHT}"; :rem 151
730 FOR KL = 1 TO 3 :rem 92
735 GETK$:IFK$=" "THEN 735 :rem 117
737 IFK$>"Z"OR K$<"A"THEN 735 :rem 26
739 PRINTK$; :rem 218
740 IM$(KL)=K$:NEXT KL :rem 249
750 POKE 53281,0:PRINT "{CLR}":POKE 53281,1
    :rem 145
760 PRINT "{3 RIGHT}ENTER THE RIGHT PLAYE
    RS 3 INITIALS{5 DOWN}" :rem 167
765 PRINT "{17 RIGHT}"; :rem 155
770 FOR KR = 1 TO 3 :rem 102

```



```

775 GETK$:IFK$=""THEN775 :rem 125
777 IFK$>"Z"OR K$<"A"THEN775 :rem 34
779 PRINTK$; :rem 222
780 IN$(KR)=K$:NEXT KR :rem 10
800 POKE53281,0:PRINT "{CLR}":POKE53281,1 :rem 141
810 E = 48 :D = 48 :T = 0 :S = 1024 :V = :rem 121
      {SPACE}1824 :Z = 1863 :rem 121
900 REM *** TITLE ON SCREEN *** :rem 88
910 DATA 34,32,3,8,1,19,5,32,1,14,4,32,20 :rem 165
      ,1,7,32,34 :rem 165
920 FOR TT = 1 TO 17 :READ TD :POKE 1874+ :rem 134
      TT,TD :NEXT TT :rem 134
1000 REM *** DRAWING THE PLAYING FIELD ** :rem 6
      * :rem 6
1010 POKE S,108:POKE S+CO,0:FOR B = 1 TO 3 :rem 163
      8 :POKE S+B,121:POKE S+B+CO,0:NEXT B :rem 163
1020 POKE S+B,123 :FOR B = 79 TO 799 STEP :rem 132
      40 :POKE S+B,117:POKE S+B+CO,0:NEXT :rem 132
      {SPACE}B :rem 91
1030 POKE Z,126 :FOR B = 38 TO 1 STEP -1 :rem 118
      {SPACE}:POKE V+B,120:POKE V+B+CO,0:NE :rem 118
      XT B :rem 118
1040 POKE V,124:POKE V+CO,0:FOR B = 760 TO :rem 160
      40 STEP -40 :POKE S+B,118 :NEXT B :rem 160
1050 : :rem 0
1100 REM *** INITIALIZING LEFT PLAYERS SC :rem 244
      ORE *** :rem 244
1110 LE = 1905 :rem 96
1120 FOR KL = 1 TO 3 :X = ASC(INS$(KL)) :P :rem 57
      OKE LE,X-64 :LE = LE+1 :NEXT KL :rem 57
1130 POKE 1949,9:POKE1949+CO,0:POKE 1950, :rem 86
      20:POKE1949+CO,0:POKE 1946,E :rem 86
1135 POKE1946+CO,0 :rem 230
1200 REM *** INITIALIZING RIGHT PLAYERS S :rem 72
      CORE *** :rem 72
1210 RE = 1940 :rem 102
1220 FOR KR = 1 TO 3 :X = ASC(INS$(KR)) :P :rem 157
      OKE RE,X-64 :RE = RE+1 :NEXT KR :rem 157
1230 POKE 1981,D:POKE1981+CO,0:GOTO 55 :rem 106
1300 POKE53281,0:PRINT "{CLR}":POKE53281, :rem 185
      1 :rem 185
1310 PRINT " IF YOU WANT TO PLAY ANOTHER :rem 96
      {SPACE}GAME" :rem 96
1320 PRINT " PRESS {RVS}'Y'{OFF} OR {RVS} :rem 229
      'N'{OFF}." :POKE198,0 :rem 229
1330 GET AN$:IF AN$="" THEN 1330 :rem 77
1340 IF AN$="N" THEN GOTO 1400 :rem 6
1350 IF AN$="Y" THEN 40 :rem 120
1360 GOTO1330 :rem 202
1400 POKE53281,0:PRINT "{CLR}":POKE53281 :rem 165
      ,1:PRINT " BYE" :rem 165
1410 END :rem 157

```

## Print Sound For The VIC-20

(Article on page 121.)

```

10 PRINTCHR$(147)"VIC PRINT-SOUND":rem 69
20 GOSUB500 :rem 119
30 PRINT"SOUND IS NOW DEVICE 2" :rem 47
40 CLOSE1:OPEN1,2:PRINT#1,"AEHMZLMZZZZZZ" :rem 187

```

```

50 IFPEEK(983)>0THEN50 :rem 16
60 PRINT"A TUNE BY J.S. BACH" :rem 86
70 PRINT#1,"FHJMKKOMMRQRMJFHJMKMOMKJHJFEFH :rem 52
      " :rem 52
80 PRINT#1,"AEHKJHJFHJMKKOMMRQRMJFHJMKMOMKJH :rem 123
      FAFEF" :rem 123
90 END :rem 64
500 DATA76,194,3,76,132,3,141,219,3,152,7 :rem 236
      2,165,154,201,2 :rem 236
510 DATA208,47,173,219,3,201,65,48,33,201 :rem 41
      ,91,16,29,56,233 :rem 41
520 DATA65,168,185,220,3,172,215,3,192,25 :rem 193
      5,240,249,172,217,3 :rem 193
530 DATA145,249,200,152,41,255,141,217,3, :rem 24
      238,215,3,104,168,173 :rem 24
540 DATA219,3,24,96,104,168,173,219,3,76, :rem 97
      122,242,72,152,72 :rem 97
550 DATA173,218,3,240,15,206,218,3,240,6, :rem 132
      104,168,104,76,191 :rem 132
560 DATA234,169,0,240,27,173,215,3,240,24 :rem 188
      1,172,216,3,177,249 :rem 188
570 DATA72,200,152,41,255,141,216,3,206,2 :rem 169
      15,3,169,10,141,218 :rem 169
580 DATA3,104,141,10,144,141,11,144,141,1 :rem 197
      2,144,24,144,207,120 :rem 197
590 DATA169,3,141,21,3,141,39,3,169,63,14 :rem 148
      1,20,3,169,66 :rem 148
600 DATA141,38,3,88,96,0,0,0,0,0,183,187, :rem 52
      191,195,199 :rem 52
610 DATA201,203,207,209,212,215,217,219,2 :rem 69
      21,223,225,227,228,229 :rem 69
620 DATA231,232,233,235,236,237,0:rem 128
630 FORI=828TO1013:READJ:POKEI,J:NEXTI :rem 154
640 POKE36878,15:SYS828:RETURN :rem 96

```

## Ski Physics

(Article on page 125.)

### 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: Ski Physics—VIC Loader

```

5 PRINT"{CLR}{4 RIGHT}{11 DOWN}JUST A SEC :rem 11
  OND" :rem 11
15 POKE51,0:POKE52,28:POKE55,0:POKE56,28: :rem 183
  CLR:B=7168:C=25600 :rem 183
20 FORJ=0TO512:POKEB+J,PEEK(B+J+C):NEXTJ :rem 229
25 FORI=216TO255:READA$:POKEB+I,A$:NEXTI: :rem 140
  FORI=280TO327:READB$:POKEB+I,B$:NEXTI :rem 140
30 PRINT"{CLR}":POKE36869,240:SC=36879:PO :rem 226
  KESC,232 :rem 226
35 PRINT"{BLK}{5 DOWN}{5 RIGHT}SKI PHYSIC :rem 185
  S" :rem 185
50 V=36878:F2=36875:POKEV,15 :rem 85
55 READA$,D$:IFA$=-1THEN65 :rem 229
60 POKEF2,A$:FORT=1TOD$:NEXTT:GOTO55 :rem 56
65 POKEV,0:POKEF2,0:D=1 :rem 117
75 PRINTSPC(69)"{BLK}PRESS THE F1 KEY" :rem 65

```



```

80 GETA$:IFA$=""THEN80 :rem 243
85 IFA$=CHR$(133)THENONGOTO100,140,185,2 :rem 139
25 :rem 177
90 GOTO80 :rem 10
100 PRINT"{CLR}":POKESC,200:PRINT:PRINTTA :rem 4
B(37)"{BLK}[2 P]"
105 PRINT"{BLK}[4 SPACES]PO[M][G] :rem 23
[N]M[2 SPACES]N[H][N][2 P]":P
RINT"{BLK}[4 SPACES][M][G][M]
[G][N] MN [H][N][2 P]"
110 PRINTSPC(47)"{RED}TIME IS DURATION":P :rem 67
RINT:PRINT"{BLK}[4 SPACES]IT IS MEASU
RED"
111 PRINT:PRINT"{6 SPACES}IN SECONDS" :rem 209
112 PRINT"{6 SPACES}MINUTES":PRINT"
{6 SPACES}AND HOURS":D=2:GOTO75
140 PRINT"{CLR}":POKESC,248:PRINT:PRINT:P :rem 197
RINT"{BLK} M[4 SPACES][P]" :rem 103
145 PRINT"{BLK}[2 SPACES]M[2 SPACES]NL :rem 222
[H][N][Y][H] O[N] POMN"
150 PRINT"{BLK}[3 SPACES]MN L L[N][P] :rem 69
[H] L[N] [M][G][M][G]"
155 PRINTSPC(47)"{RED}VELOCITY IS SPEED": :rem 94
PRINT:PRINT:PRINT"{BLK}[4 SPACES]IT I
S MEASURED"
160 PRINT"{BLK}[4 SPACES]IN FEET/SECOND": :rem 142
PRINT"{BLK}[4 SPACES]OR MILES/HOUR":D
=3:GOTO75
185 PRINT"{CLR}":POKESC,216:PRINT:PRINT:P :rem 209
RINT"{BLK}[3 SPACES][P][3 SPACES]
[P]"TAB(18)"[P]"
190 PRINT"{3 SPACES}[H]M[M][G]L PON :rem 4
M[N]M[H]O L"
195 PRINT"{3 SPACES}[H]N[M][G]@ :rem 155
[M][G]OP[N] [H]L L":PRINT"
{3 SPACES}[Y]"
200 PRINTSPC(46)"{RED}DISTANCE IS LENGTH": :rem 129
PRINT:PRINT:PRINT"{BLK}[3 SPACES]IT
{SPACE}IS MEASURED"
205 PRINT"{3 SPACES}IN FEET OR MILES":D=4 :rem 12
:GOTO75
225 PRINT"{CLR}":POKESC,168:PRINTSPC(92)" :rem 125
*INSTRUCTIONS*"
227 PRINT:PRINT:PRINT"{2 SPACES}YOU WILL :rem 47
{SPACE}BE GIVEN"
230 PRINT"{2 SPACES}PROBLEMS TO SOLVE":PR :rem 94
INT"{2 SPACES}WITH TIME,DISTANCE":PRI
NT"{2 SPACES}AND VELOCITY."
235 PRINT"{2 SPACES}ANSWER WITH THE":PRIN :rem 108
T"{2 SPACES}CORRECT NUMBER."
236 PRINT"{2 SPACES}IGNORE REMAINDERS." :rem 72
237 PRINT"{2 SPACES}DO NOT GIVE UNITS." :rem 218
D=5:PRINT"{3 DOWN}[7 RIGHT]PRESS F1"
238 IFPEEK(197)<>39THEN238 :rem 245
240 PRINT"{CLR}TAPE OR DISK":INPUTT$ :rem 50
245 IFLEFT$(T$,1)<>"T"ANDLEFT$(T$,1)<>"D" :rem 192
THEN240
250 IFLEFT$(T$,1)="T"THENPOKE631,131:POKE :rem 162
198,1:END
260 PRINT"{CLR}FILENAME":INPUTFI$ :rem 186
263 FI$="LOAD"+CHR$(34)+FI$+CHR$(34)+",8" :rem 104
265 PRINT"{CLR}[3 DOWN]";FI$+"{HOME}"

```

```

{DOWN}"; :rem 240
266 POKE631,13:POKE632,13 :rem 139
268 POKE198,2 :rem 207
500 DATA96,96,60,31,44,72,142,3,0,0,0,128 :rem 176
,16,32,64,128,1,2,4,12,0,0,0,0,0,0
505 DATA24,24,8,30,45,76,152,16,16,24,8,5 :rem 52
6,255,0,0,0,0,0,0
507 DATA16,15,128,64,32,0,0,0,0,240 :rem 207
510 DATA56,56,124,126,255,255,255,255,74, :rem 158
32,136,2,212,21,162,205,255,255,255,2
55
515 DATA255,255,255,255,1,2,4,8,16,32,64, :rem 169
128
520 DATA231,200,225,150,231,350,0,30,225, :rem 59
150,229,175,225,220,229,150,225,175,2
29,200
525 DATA228,350,231,200,225,200,231,150,2 :rem 217
32,250,231,275,228,175,225,400,-1,-1

```

## Program 2:

### Ski Physics—VIC Main Program

```

240 SC=36879:C=25600 :rem 49
250 PRINT"{CLR}":POKESC,152:POKE36869,255 :rem 255
255 A=INT(RND(1)*3+1):B=INT(RND(1)*7+1):X :rem 174
=INT(RND(1)*26+5):Y=INT(RND(1)*101+50
)
260 IFA=1THEND$="FEET":TT$="SECONDS":V$=" :rem 165
F/S"
265 IFA=2THEND$="MILES":TT$="HOURS":V$="M :rem 160
PH"
270 IFA=3THEND$="FEET":TT$="MINUTES":V$=" :rem 184
F/M"
280 ONBGOSUB440,470,440,455,470,440,470 :rem 56
285 PRINT:PRINT:PRINT:PRINTTAB(19)"{BLK}[ :rem 254
f":PRINTTAB(19)"{BLU}]{BLU}{"
290 FORN=19TO15STEP-1:PRINTTAB(N)"{BLU}{" :rem 169
:NEXTN
295 PRINTTAB(12)"{BLU}'':PRINTTAB(12) :rem 86
"BLU}'':PRINT"
300 PRINT"{PUR}'':{YEL}[2 SPACES] :rem 28
{BLU}'':PRINT"
{2 SPACES}"
305 PRINT"{16 UP}" :rem 121
310 Z=7680+(9*22)+20:C=Z+30720:V=36878:F2 :rem 228
=36875
315 INPUT"{BLK}[5 SPACES]";S$:S=VAL(S$):I :rem 144
FS=KTHEN360
330 POKEV,15:PRINT"{RED}[2 SPACES]UH UH! :rem 206
{SPACE}SORRY[2 SPACES]"
335 FORM=230TO190STEP-1:POKEF2,M:FORM=1TO :rem 32
5:NEXTT:NEXTM
340 Q=5:GOSUB480:POKEV,0:FORN=1TO4:POKEZ, :rem 124
32:POKEZ+21,32:POKEZ+22,32:POKEZ+23,3
2
345 POKEZ+22,30:POKEC+22,0:POKEZ+43,35:PO :rem 222
KEC+43,0:POKEZ+44,31:POKEC+44,0
350 POKEZ+45,36:POKEC+45,0:Z=Z+22:C=C+22: :rem 243
NEXTN
354 FORN=0TO4:POKEZ+20+N,38:POKEC+20+N,7: :rem 81
NEXTN
355 POKE36877,210:FORL=15TO0STEP-1:POKEV, :rem 5
L:FORM=1TO50:NEXTM:NEXTL:PP=1

```



```

356 POKE36877,0:POKEV,0:GOTO400 :rem 90
360 POKEV,15:PRINT"{RED} YEP! THATS RIGHT
      ":FORM=190TO230:POKEF2,M:FORT=1TO5:N
      EXT:NEXTM :rem 13
362 Q=10:GOSUB480:POKEV,0:FORN=1TO3:POKEZ
      ,32:POKEZ+21,32:POKEZ+22,32:POKEZ+23,
      32 :rem 171
364 POKEZ+21,30:POKEC+21,0:POKEZ+42,35:PO
      KEC+42,0:POKEZ+43,31:POKEC+43,0
      :rem 217
366 POKEZ+44,36:POKEC+44,0:Z=Z+21:C=C+21:
      NEXTN :rem 246
368 POKEV,15:FORN=148TO220STEP.7:POKEF2,L
      :NEXTL:POKEV,0:FORT=1TO500:NEXTT:PP=2
      :rem 65
400 PRINT"{CLR}":POKESC,184 :rem 192
405 IFPP=1THENPRINTSPC(134)"YOU LOST THIS
      TIME" :rem 171
407 IFPP=1THENPRINTSPC(49)"BETTER LUCK
      {12 SPACES}NEXT TIME" :rem 63
410 IFPP=2THENPRINTSPC(158)"YOU SOLVED TH
      E{8 SPACES}GIVEN{2 SPACES}PROBLEM"
      :rem 61
412 FORT=1TO5000:NEXT :rem 33
415 D=5:GOTO250 :rem 90
440 K=INT(Y/X) :rem 60
441 PRINTSPC(24)"{BLK}A SKIER WENT {RED}"
      ;Y:PRINT"{BLU}{2 SPACES}";D$ :rem 17
442 PRINT"{BLK}{2 SPACES}IN {RED}";X;"
      {BLU}";TT$ :rem 230
445 PRINT"{BLK}{2 SPACES}WHAT WAS HIS":PR
      INT"{BLK}{2 SPACES}VELOCITY IN {BLU}"
      ;V$:RETURN :rem 144
455 K=X*Y:PRINTSPC(24)"{BLK}A SKIER WENT
      {SPACE}{RED}";X:PRINT"{BLU}{2 SPACES}
      ";V$:PRINT"{BLK}{2 SPACES}FOR {RED}";
      Y"{BLU}";TT$ :rem 96
460 PRINT"{BLK}{2 SPACES}WHAT WAS HIS":PR
      INT"{BLK}{2 SPACES}DISTANCE IN {RED}"
      ;"{BLU}";D$:RETURN :rem 242
470 K=INT(Y/X) :rem 63
471 PRINTSPC(24)"{BLK}A SKIER WENT {RED}"
      ;Y:PRINT"{BLU}{2 SPACES}";D$ :rem 20
472 PRINT"{BLK}{2 SPACES}AT {RED}";X;"
      {BLU}";V$ :rem 149
475 PRINT"{BLK}{2 SPACES}HOW MUCH TIME DI
      D":PRINT"{BLK}{2 SPACES}HE SKI IN
      {BLU}";TT$:RETURN :rem 2
480 FORT=1TO5:POKEZ,32:POKEZ+1,32:POKEZ+2
      2,32:POKEZ+21,27:POKEC+21,0 :rem 122
482 POKEZ+22,28:POKEC+22,0:POKEZ+43,29:PO
      KEC+43,0:Z=Z+21:C=C+21:NEXTN :rem 116
484 POKEV,0:Z=Z-21:POKEZ,32:POKEZ+1,32:PO
      KEZ+22,32 :rem 35
486 Z=7680+(14*22)+14:C=Z+30720:POKEZ+1,3
      2 :rem 248
488 FORT=1TOQ:POKEZ,32:POKEZ+21,32:POKEZ+
      22,32:POKEZ+23,32 :rem 56
490 POKEZ-1,30:POKEC-1,0:POKEZ+20,35:POKE
      C+20,0:POKEZ+21,31:POKEC+21,0:rem 105
492 POKEZ+22,36:POKEC+22,0:Z=Z-1:C=C-1:NE
      XTN:RETURN :rem 168

```

## Program 3:

### Ski Physics—64 Version

```

5 PRINT"{CLR}" :rem 153
15 POKE52,48:POKE56,48:CLR:K=12288:PC=563
      34:CP=53272 :rem 254
16 POKEPC,PEEK(PC)AND254:POKE1,PEEK(1)AND

```

```

251 :rem 164
20 FORJ=0TO512:POKEK+J,PEEK(53248+J):NEXT
      J :rem 68
25 FORI=216TO255:READA$:POKEK+I,A$:NEXTI:
      FORI=280TO327:READB$:POKEK+I,B$:NEXTI
      :rem 158
26 POKE1,PEEK(1)OR4:POKEPC,PEEK(PC)OR1
      :rem 115
30 PRINT"{CLR}":POKECP,21:BC=53280:BK=532
      81:POKEBC,5:POKEBK,1 :rem 249
35 PRINT"{BLK}{8 DOWN}{13 RIGHT}SKI PHYSI
      CS" :rem 212
50 V=54272:FORN=VTOV+24:POKE1,0:NEXT
      :rem 16
52 POKEV+5,88:POKEV+4,33:POKEV+24,143:POK
      EV+6,195 :rem 151
55 READLF,HF,DR:IFLF=-1THEN65 :rem 36
60 POKEV,LF:POKEV+1,HF:FORT=1TODR:NEXTT:G
      OTO55 :rem 68
65 POKEV,0:POKEV+1,0:POKEV+24,0:D=1
      :rem 91
75 PRINTSPC(131)"{BLK}PRESS THE{2 SPACES}
      F1 KEY" :rem 103
80 GETA$:IFA$=""THEN80 :rem 243
85 IFA$=CHR$(133)THENONGOTO100,140,185,2
      25,250 :rem 116
90 GOTO80 :rem 10
100 PRINT"{CLR}":POKEBC,6:POKEBK,1:PRINT:
      PRINTSPC(104)"{BLK}{2 P$}" :rem 36
105 PRINTSPC(13)"{BLK}POKE{G$}{N$}M
      {2 SPACES}{N$}{H$}{N$}{2 P$}":PRINTSP
      C(13)"{BLK}{N$}{H$}{M$}{G$}{N$}
      {SPACE}MN {H$}{N$}{2 P$}" :rem 95
110 PRINTSPC(92)"{RED}TIME IS DURATION"
      :rem 82
111 PRINTSPC(93)"{BLK}IT IS MEASURED"
      :rem 38
112 PRINTSPC(55)"IN MINUTES":PRINTSPC(15)
      "AND HOURS":D=2:GOTO75 :rem 154
140 PRINT"{CLR}":POKEBC,4:POKEBK,1:PRINTS
      PC(130)"{BLK} M{4 SPACES}{P$}"
      :rem 124
145 PRINTSPC(10)"{BLK}{2 SPACES}M
      {2 SPACES}{NL}{H$}{N$}{Y$}{H$} O
      {N$} POMN" :rem 118
150 PRINTSPC(10)"{BLK}{3 SPACES}MN L L
      {N$}{P$}{H$} L{N$} {M$}{G$}
      {M$}{G$}" :rem 221
155 PRINTSPC(92)"{RED}VELOCITY IS SPEED":
      PRINTSPC(93)"{BLK}IT IS MEASURED"
      :rem 115
160 PRINTSPC(13)"IN FEET/SECOND":PRINTSPC
      (13)"OR MILES/HOUR":D=3:GOTO75
      :rem 164
185 PRINT"{CLR}":POKEBC,7:POKEBK,1
      :rem 166
189 PRINTSPC(129)"{3 SPACES}{P$}
      {3 SPACES}{P$}{10 SPACES}{P$}"
      :rem 83
190 PRINTSPC(9)"{3 SPACES}{H$}M{M$}
      {G$}L PONM{N$}M{H$}O L" :rem 116
195 PRINTSPC(9)"{3 SPACES}{H$}{N$}{M$}
      {G$}@ {M$}{G$}OP{N$} {H$}L L"
      :rem 73
196 PRINTSPC(9)"{3 SPACES}{T$}" :rem 132
200 PRINTSPC(91)"{RED}DISTANCE IS LENGTH"
      :PRINTSPC(92)"{BLK}IT IS MEASURED"
      :rem 149
205 PRINTSPC(52)"IN FEET OR MILES":D=4:G
      OTO75 :rem 170

```



```

225 PRINT "{CLR}":POKEBC,13:POKEBK,1:PRINT
   SPC(132)*"INSTRUCTIONS*" :rem 175
226 PRINT:PRINT:PRINTTAB(11)"YOU WILL BE
   {SPACE}GIVEN" :rem 184
230 PRINTSPC(11)"PROBLEMS TO SOLVE":PRINT
   SPC(11)"WITH TIME,DISTANCE" :rem 21
231 PRINTSPC(11)"AND VELOCITY." :rem 112
235 PRINTSPC(11)"ANSWER WITH THE":PRINTSP
   C(11)"CORRECT NUMBER." :rem 158
236 PRINTSPC(11)"IGNORE REMAINDERS."
   :rem 225
237 PRINTSPC(11)"DO NOT GIVE UNITS.":D=5:
   GOTO75 :rem 69
250 PRINT "{CLR}":POKEBC,10:POKEBK,1:POKEC
   P,(PEEK(CP)AND240)+12 :rem 66
255 A=INT(RND(0)*3+1):B=INT(RND(0)*7+1):X
   =INT(RND(0)*21+5):Y=INT(RND(0)*101+50
   ) :rem 165
260 IFA=1THEND$="FEET":TT$="SECONDS":V$="
   F/S" :rem 165
265 IFA=2THEND$="MILES":TT$="HOURS":V$="M
   PH" :rem 160
270 IFA=3THEND$="FEET":TT$="MINUTES":V$="
   F/M" :rem 184
280 ONBGOSUB440,470,440,455,470,440,470
   :rem 56
285 PRINT:PRINT:PRINT:PRINTTAB(29)"{BLK}[
   £":PRINTTAB(29)"{BLU}]{BLU}(" :rem 0
290 FORN=29TO25STEP-1:PRINTTAB(N)"{BLU}("
   :NEXTN :rem 171
295 PRINTTAB(22)"{BLU}' '' ''':PRINTTAB(22)
   "{BLU}' '' '''" :rem 88
300 PRINTTAB(5)"{PUR}' '' '' '' '' '' '' ''
   {YEL}
   {2 SPACES}%%%%{BLU}' '' '' '' '' '' ''
   :rem 223
301 PRINTTAB(3)"{PUR}' '' '' '' '' '' '' ''
   :rem 130
305 PRINT "{15 UP}" :rem 232
310 Z=1024+(9*40)+30:V=54272:C=Z+V:rem 41
315 INPUT "{BLK}{5 SPACES}":S$:S=VAL(S$):I
   FS=KTHEN360 :rem 144
330 POKEV+24,143:PRINT "{RED}{2 SPACES}UH
   {SPACE}UH! SORRY{2 SPACES}" :rem 145
335 FORM=60TO20STEP-5:POKEV,INT(M/3):POKE
   V+1,M:FORT=1TO15:NEXTT:NEXTM :rem 254
340 Q=5:GOSUB480:POKEV+1,0:POKEV,0:FORN=1
   TO4:POKEZ,32:POKEZ+39,32 :rem 53
341 POKEZ+40,32:POKEZ+41,32 :rem 37
345 POKEZ+40,30:POKEC+40,0:POKEZ+79,35:PO
   KEC+79,0:POKEZ+80,31:POKEC+80,0
   :rem 240
350 POKEZ+81,36:POKEC+81,0:Z=Z+40:C=C+40:
   NEXTN :rem 243
354 FORN=0TO4:POKEZ+38+N,38:POKEC+38+N,7:
   NEXTN:POKEV,195:POKEV+1,16 :rem 155
355 FORL=15TO5STEP-1:POKEV+24,L:FORT=1TO5
   0:NEXTT:NEXTL:PP=1:POKEV,0:POKEV+1,0
   :rem 4
356 GOTO400 :rem 107
360 PRINT "{RED} YEP! THATS RIGHT ":FORM=0
   TO60STEP2:POKEV,M/2 :rem 130
361 POKEV+1,INT(M/2):NEXTM:POKEV,0:POKEV+
   1,0 :rem 233
362 Q=10:GOSUB480:FORN=1TO3:POKEZ,32:POKE
   Z+39,32 :rem 210
363 POKEZ+40,32:POKEZ+41,32 :rem 41
364 POKEZ+39,30:POKEC+39,0:POKEZ+78,35:PO
   KEC+78,0:POKEZ+79,31:POKEC+79,0
   :rem 15
366 POKEZ+80,36:POKEC+80,0:Z=Z+39:C=C+39:
   NEXTN :rem 8
367 POKEV+24,143 :rem 122
368 FORM=12TO60STEP4:POKEV,M/2:POKEV+1,M:
   FORT=1TO25:NEXTT:NEXTM:POKEV,0
   :rem 182
369 POKEV+1,0:PP=2 :rem 40
400 PRINT "{CLR}":POKEBC,14:POKEBK,1
   :rem 202
404 IFPP=2THEN410 :rem 250
405 PRINT:PRINT:PRINTSPC(171)"YOU LOST
   {2 SPACES}THIS TIME" :rem 110
406 PRINTSPC(90)"BETTER LUCK NEXT TIME"
   :rem 110
407 GOTO415 :rem 110
410 PRINT:PRINT:PRINTSPC(206)"YOU SOLVED
   {SPACE}THE GIVEN PROBLEM" :rem 248
415 D=5:GOTO75 :rem 47
440 K=INT(Y/X) :rem 60
441 PRINTSPC(42)"{BLK}A SKIER WENT {RED}"
   ;Y:PRINT "{BLU}{2 SPACES}";D$ :rem 17
442 PRINT "{BLK}{2 SPACES}IN {RED}";X;"
   {BLU}";TT$ :rem 230
445 PRINT "{BLK}{2 SPACES}WHAT WAS HIS":PR
   INT "{BLK}{2 SPACES}VELOCITY IN {BLU}"
   ;V$:RETURN :rem 144
455 K=X*Y:PRINTSPC(42)"{BLK}A SKIER WENT
   {SPACE}{RED}";X :rem 229
456 PRINT "{BLU}{2 SPACES}";V$:PRINT "{BLK}
   {2 SPACES}FOR {RED}";Y"{BLU}";TT$
   :rem 224
460 PRINT "{BLK}{2 SPACES}WHAT WAS HIS":PR
   INT "{BLK}{2 SPACES}DISTANCE IN {RED}"
   ;"{BLU}";D$:RETURN :rem 242
470 K=INT(Y/X) :rem 63
471 PRINTSPC(42)"{BLK}A SKIER WENT {RED}"
   ;Y:PRINT "{BLU}{2 SPACES}";D$ :rem 20
472 PRINT "{BLK}{2 SPACES}AT {RED}";X;"
   {BLU}";V$ :rem 149
475 PRINT "{BLK}{2 SPACES}HOW MUCH TIME DI
   D":PRINT "{BLK}{2 SPACES}HE SKI IN
   {BLU}";TT$:RETURN :rem 2
480 FORN=1TO5:POKEZ,32:POKEZ+1,32:POKEZ+4
   0,32:POKEZ+39,27:POKEC+39,0 :rem 140
482 POKEZ+40,28:POKEC+40,0:POKEZ+79,29:PO
   KEC+79,0:Z=Z+39:C=C+39:NEXTN :rem 152
484 POKEV+24,0:Z=Z-39:POKEZ,32:POKEZ+1,32
   :POKEZ+40,32 :rem 189
486 Z=1024+(14*40)+24:C=Z+V:POKEZ+1,32
   :rem 69
488 FORN=1TOQ:POKEZ,32:POKEZ+39,32:POKEZ+
   40,32:POKEZ+41,32 :rem 65
490 POKEZ-1,30:POKEC-1,0:POKEZ+38,35:POKE
   C+38,0:POKEZ+39,31:POKEC+39,0:rem 141
492 POKEZ+40,36:POKEC+40,0:Z=Z-1:C=C-1:NE
   XTN:RETURN :rem 168
500 DATA96,96,60,31,44,72,142,3,0,0,0,128
   ,16,32,64,128,1,2,4,12,0,0,0,0,0,0
   :rem 176
505 DATA24,24,8,30,45,76,152,16,16,24,8,5
   6,255,0,0,0,0,0,0 :rem 52
506 DATA16,15,128,64,32,0,0,0,0,240
   :rem 206
510 DATA56,56,124,126,255,255,255,255,74,
   32,136,2 :rem 166
512 DATA212,21,162,205,255,255,255,255
   :rem 126
515 DATA255,255,255,255,1,2,4,8,16,32,64,
   128 :rem 169
520 DATA31,21,175,195,16,125,31,21,275,19
   5,16,125,239,19,150,195,16,100
   :rem 215

```



```

525 DATA239,19,150,195,16,100,239,19,125,
    209,18,300,31,21,175,195,16,125
                                     :rem 16
530 DATA31,21,170,96,22,200,31,21,230,209
    ,18,115,195,16,400,-1,-1,-1
                                     :rem 13

```

# Sound Sculptor For The 64

(Article on page 46.)

## 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.

Tape users: Program 1 automatically loads Program 2. It is recommended that you SAVE them on the same tape.

Disk users: SAVE Program 2 as "2". Program 1 must be LOADED and RUN before using Program 2.

## Program 1: Sound Sculptor—ML Loader

```

80 PRINT"{CLR}":PRINT:PRINT"{6 SPACES}PLE
    ASE WAIT ONE MOMENT..."
                                     :rem 201
90 REM SPRITE CREATOR
                                     :rem 52
100 POKE2040,11:FORL=0TO24:READSP:POKE704
    +L,SP:NEXTL:POKE53287,0
                                     :rem 205
110 FORL=25TO63:POKE704+L,0:NEXTL:rem 17
200 REM ML PROGRAM POKER
                                     :rem 168
210 FORL=49152 TO 50702
                                     :rem 169
220 READ DA:POKE L,DA:CK=CK+DA:NEXT
                                     :rem 87
230 IF CK<>211739 THEN PRINT "ERROR IN DA
    TA STATEMENTS":STOP
                                     :rem 38
240 PRINT"{CLR}{13 RIGHT}{11 DOWN}{RVS}T
    {OFF}APE OR {RVS}D{OFF}ISK"
                                     :rem 108
250 GET T$:IF T$=""THEN250
                                     :rem 119
255 IF T$<>"D"ANDT$<>"T"THEN250
                                     :rem 200
260 IF T$="D"THEN 380
                                     :rem 46
300 POKE 631,76:POKE632,207:POKE633,13:PO
    KE198,3
                                     :rem 189
350 FOR T= 1 TO 1000:NEXT:GOTO1000:rem 82
380 POKE50660,8:POKE50662,8
                                     :rem 255
400 POKE631,76:POKE632,207:POKE633,34:POK
    E634,50:POKE635,34:POKE636,44
                                     :rem 36
405 POKE637,56
                                     :rem 255
410 POKE638,58:POKE639,13:POKE198,9:rem 9
1000 REM SPRITE DATA
                                     :rem 150
1010 DATA48,0,0,56,0,0,60,0,0,62,0,0,45,0
    ,0,36,0,0,4,0,0,2,0,0,2
                                     :rem 11
40000 REM ML DATA
                                     :rem 139
49152 DATA 32,140,197,160,0,177
                                     :rem 252
49158 DATA 78,153,0,212,200,192
                                     :rem 252
49164 DATA 25,208,246,32,93,194
                                     :rem 10
49170 DATA 165,197,201,60,240,23
                                     :rem 44
49176 DATA 169,16,45,0,220,208
                                     :rem 207
49182 DATA 225,165,2,240,6,32
                                     :rem 150
49188 DATA 86,192,76,0,192,32
                                     :rem 172
49194 DATA 48,192,76,0,192,96
                                     :rem 177
49200 DATA 162,21,189,24,197,202
                                     :rem 46
49206 DATA 205,1,208,48,8,189
                                     :rem 163
49212 DATA 24,197,205,1,208,48

```

```

49218 DATA 4,202,16,236,96,189
                                     :rem 217
49224 DATA 48,197,133,75,232,189
                                     :rem 68
49230 DATA 48,197,133,76,108,75
                                     :rem 14
49236 DATA 0,234,162,15,189,72
                                     :rem 209
49242 DATA 197,202,205,1,208,48
                                     :rem 254
49248 DATA 8,189,72,197,205,1
                                     :rem 173
49254 DATA 208,48,4,202,16,236
                                     :rem 206
49260 DATA 96,189,88,197,133,75
                                     :rem 32
49266 DATA 232,189,88,197,133,76
                                     :rem 79
49272 DATA 108,75,0,234,96,24
                                     :rem 161
49278 DATA 173,0,208,233,142,144
                                     :rem 49
49284 DATA 247,74,74,74,74,141
                                     :rem 224
49290 DATA 249,207,32,175,192,76
                                     :rem 65
49296 DATA 96,196,96,234,234,24
                                     :rem 26
49302 DATA 173,0,208,233,74,144
                                     :rem 249
49308 DATA 245,41,240,160,3,81
                                     :rem 199
49314 DATA 253,41,240,81,253,145
                                     :rem 45
49320 DATA 253,32,175,192,76,140
                                     :rem 50
49326 DATA 196,160,3,177,253,74
                                     :rem 12
49332 DATA 74,74,74,10,170,160
                                     :rem 204
49338 DATA 0,189,0,197,145,253
                                     :rem 216
49344 DATA 232,200,189,0,197,145
                                     :rem 52
49350 DATA 253,24,169,8,237,249
                                     :rem 14
49356 DATA 207,170,240,15,177,253:rem 103
49362 DATA 74,145,253,136,177,253:rem 112
49368 DATA 106,145,253,200,202,208
                                     :rem 143
49374 DATA 241,96,24,173,0,208
                                     :rem 210
49380 DATA 233,144,144,8,169,128
                                     :rem 59
49386 DATA 32,32,193,76,198,195
                                     :rem 25
49392 DATA 169,64,32,32,193,76
                                     :rem 223
49398 DATA 181,195,24,173,0,208
                                     :rem 11
49404 DATA 233,144,176,8,169,32
                                     :rem 7
49410 DATA 32,32,193,76,215,195
                                     :rem 3
49416 DATA 169,16,32,32,193,76
                                     :rem 217
49422 DATA 232,195,169,4,32,32
                                     :rem 207
49428 DATA 193,76,249,195,169,2
                                     :rem 27
49434 DATA 32,32,193,76,10,196
                                     :rem 211
49440 DATA 160,4,81,253,145,253
                                     :rem 253
49446 DATA 96,234,234,24,173,0
                                     :rem 211
49452 DATA 208,233,133,144,245,170
                                     :rem 147
49458 DATA 169,0,160,2,145,253
                                     :rem 209
49464 DATA 200,177,253,41,240,72
                                     :rem 48
49470 DATA 145,253,138,162,5,136
                                     :rem 54
49476 DATA 10,145,253,200,177,253:rem 100
49482 DATA 42,145,253,136,177,253:rem 110
49488 DATA 202,208,241,200,177,253
                                     :rem 151
49494 DATA 41,15,145,253,104,24
                                     :rem 254
49500 DATA 113,253,145,253,76,193:rem 101
49506 DATA 194,96,234,162,0,32
                                     :rem 210
49512 DATA 133,193,76,208,194,162:rem 109
49518 DATA 1,32,133,193,76,228
                                     :rem 212
49524 DATA 194,162,2,32,133,193
                                     :rem 0
49530 DATA 76,246,194,162,3,32
                                     :rem 211
49536 DATA 133,193,76,10,195,24
                                     :rem 8
49542 DATA 173,0,208,233,133,144
                                     :rem 43
49548 DATA 214,74,74,74,157,245
                                     :rem 20
49554 DATA 207,160,5,173,245,207
                                     :rem 53
49560 DATA 10,10,10,10,13,246
                                     :rem 130
49566 DATA 207,145,253,200,173,247
                                     :rem 155
49572 DATA 207,10,10,10,10,13
                                     :rem 130
49578 DATA 248,207,145,253,96,234:rem 122
49584 DATA 169,0,160,1,145,253
                                     :rem 208
49590 DATA 24,173,0,208,233,133
                                     :rem 251
49596 DATA 144,240,10,145,253,76
                                     :rem 56
49602 DATA 28,195,234,24,173,0
                                     :rem 206
49608 DATA 208,233,133,144,225,74:rem 103

```



```

49614 DATA 74,74,74,74,162,1 :rem 116
49620 DATA 168,240,6,138,10,136 :rem 253
49626 DATA 208,252,170,138,77,251:rem 110
49632 DATA 207,141,251,207,32,254 :rem 94
49638 DATA 193,76,88,195,96,173 :rem 39
49644 DATA 0,208,233,133,144,248 :rem 49
49650 DATA 10,234,234,141,252,207 :rem 89
49656 DATA 32,254,193,76,48,195 :rem 25
49662 DATA 173,252,207,77,251,207:rem 109
49668 DATA 41,240,77,251,207,160 :rem 56
49674 DATA 2,145,253,96,173,0 :rem 164
49680 DATA 208,233,133,144,248,74:rem 108
49686 DATA 74,74,141,253,207,32 :rem 11
49692 DATA 65,194,76,68,195,169 :rem 39
49698 DATA 64,32,65,194,76,44 :rem 183
49704 DATA 196,169,32,32,65,194 :rem 17
49710 DATA 76,61,196,169,16,32 :rem 219
49716 DATA 65,194,76,78,196,169 :rem 38
49722 DATA 128,32,65,194,76,27 :rem 221
49728 DATA 196,160,3,77,254,207 :rem 16
49734 DATA 141,254,207,76,124,197:rem 112
49740 DATA 234,234,234,234,24,169:rem 104
49746 DATA 128,113,251,145,251,136 :rem 152
49752 DATA 208,246,96,234,234,165:rem 116
49758 DATA 197,201,3,48,247,201 :rem 10
49764 DATA 7,16,243,201,4,208 :rem 156
49770 DATA 4,32,137,194,96,201 :rem 213
49776 DATA 5,208,4,32,149,194 :rem 171
49782 DATA 96,201,6,208,4,32 :rem 111
49788 DATA 161,194,96,32,137,194 :rem 77
49794 DATA 32,149,194,32,161,194 :rem 67
49800 DATA 96,169,1,160,4,81 :rem 111
49806 DATA 78,145,78,141,4,212 :rem 216
49812 DATA 96,169,1,160,11,81 :rem 160
49818 DATA 78,145,78,141,11,212 :rem 9
49824 DATA 96,169,1,160,18,81 :rem 170
49830 DATA 78,145,78,141,18,212 :rem 10
49836 DATA 96,41,15,170,160,16 :rem 212
49842 DATA 169,43,145,251,136,208:rem 110
49848 DATA 251,232,138,168,169,171 :rem 170
49854 DATA 145,251,96,169,5,133 :rem 17
49860 DATA 251,169,7,133,252,160 :rem 55
49866 DATA 3,177,253,76,173,194 :rem 25
49872 DATA 169,117,133,251,169,5 :rem 65
49878 DATA 133,252,160,5,177,253 :rem 63
49884 DATA 74,74,74,74,170,76 :rem 184
49890 DATA 176,194,169,157,133,251 :rem 172
49896 DATA 169,5,133,252,160,5 :rem 219
49902 DATA 177,253,41,15,170,76 :rem 7
49908 DATA 176,194,169,197,133,251 :rem 176
49914 DATA 169,5,133,252,160,6 :rem 211
49920 DATA 177,253,74,74,74,74 :rem 227
49926 DATA 170,76,176,194,169,237:rem 129
49932 DATA 133,251,169,5,133,252 :rem 53
49938 DATA 160,6,177,253,41,15 :rem 216
49944 DATA 170,76,176,194,169,173:rem 128
49950 DATA 133,251,169,4,133,252 :rem 52
49956 DATA 160,1,177,253,74,74 :rem 222
49962 DATA 74,74,170,76,176,194 :rem 27
49968 DATA 169,237,133,251,169,5 :rem 74
49974 DATA 133,252,160,2,177,253 :rem 57
49980 DATA 74,74,74,74,170,76 :rem 181
49986 DATA 176,194,169,181,133,251 :rem 175
49992 DATA 169,6,133,252,160,3 :rem 215
49998 DATA 177,253,41,15,170,76 :rem 22

50004 DATA 176,194,234,234,169,78:rem 108
50010 DATA 133,251,169,5,133,252 :rem 32
50016 DATA 169,1,162,1,160,2 :rem 86
50022 DATA 72,49,253,240,29,138 :rem 252
50028 DATA 168,177,251,201,127,16 :rem 95
50034 DATA 7,169,128,24,113,251 :rem 247
50040 DATA 145,251,232,232,232,232 :rem 125
50046 DATA 104,10,224,17,240,3 :rem 180
50052 DATA 76,100,195,96,138,168 :rem 57
50058 DATA 177,251,201,127,48,234 :rem 97
50064 DATA 169,128,24,113,251,145 :rem 93
50070 DATA 251,76,122,195,49,253 :rem 50
50076 DATA 240,11,138,168,177,251 :rem 98
50082 DATA 201,127,16,14,76,80 :rem 195
50088 DATA 194,138,168,177,251,201 :rem 158
50094 DATA 127,48,3,76,80,194 :rem 164
50100 DATA 96,169,170,133,251,169 :rem 98
50106 DATA 6,133,252,162,6,169 :rem 199
50112 DATA 64,160,4,76,154,195 :rem 202
50118 DATA 169,189,133,251,169,6 :rem 60
50124 DATA 133,252,162,5,169,128 :rem 43
50130 DATA 160,4,76,154,195,169 :rem 0
50136 DATA 90,133,251,169,6,133 :rem 250
50142 DATA 252,162,6,169,32,160 :rem 246
50148 DATA 4,76,154,195,169,109 :rem 12
50154 DATA 133,251,169,6,133,252 :rem 42
50160 DATA 162,6,169,16,160,4 :rem 147
50166 DATA 76,154,195,169,153,133:rem 110
50172 DATA 251,169,7,133,252,162 :rem 45
50178 DATA 15,169,4,160,4,76 :rem 109
50184 DATA 154,195,169,113,133,251 :rem 149
50190 DATA 169,7,133,252,162,15 :rem 251
50196 DATA 169,2,160,4,76,154 :rem 159
50202 DATA 195,169,45,133,251,169:rem 102
50208 DATA 7,133,252,162,15,169 :rem 251
50214 DATA 128,160,3,76,154,195 :rem 253
50220 DATA 169,201,133,251,169,4 :rem 37
50226 DATA 133,252,162,9,169,64 :rem 1
50232 DATA 160,3,76,154,195,169 :rem 2
50238 DATA 25,133,251,169,5,133 :rem 250
50244 DATA 252,162,9,169,32,160 :rem 252
50250 DATA 3,76,154,195,169,105 :rem 1
50256 DATA 133,251,169,5,133,252 :rem 44
50262 DATA 162,9,169,16,160,3 :rem 152
50268 DATA 76,154,195,234,162,49 :rem 64
50274 DATA 160,0,169,95,133,251 :rem 252
50280 DATA 169,4,133,252,138,145 :rem 47
50286 DATA 251,200,200,232,192,16 :rem 81
50292 DATA 208,246,160,1,177,253 :rem 47
50298 DATA 162,255,232,234,74,208:rem 104
50304 DATA 251,138,10,168,24,169 :rem 44
50310 DATA 128,113,251,145,251,96 :rem 88
50316 DATA 162,0,160,0,169,167 :rem 195
50322 DATA 133,251,169,4,133,252 :rem 37
50328 DATA 189,112,197,145,251,200 :rem 147
50334 DATA 200,232,192,24,208,244 :rem 84
50340 DATA 160,3,177,253,74,74 :rem 203
50346 DATA 74,74,10,168,24,169 :rem 212
50352 DATA 128,113,251,145,251,96 :rem 94
50358 DATA 234,234,32,96,196,32 :rem 6
50364 DATA 140,196,32,193,194,32 :rem 50
50370 DATA 198,195,32,181,195,32 :rem 57
50376 DATA 215,195,32,232,195,32 :rem 50
50382 DATA 249,195,32,10,196,32 :rem 1
50388 DATA 208,194,32,228,194,32 :rem 58
50394 DATA 246,194,32,10,195,76 :rem 7

```



```

50400 DATA 0,192,234,234,234,32 :rem 235
50406 DATA 28,195,32,48,195,32 :rem 211
50412 DATA 88,195,32,27,196,32 :rem 212
50418 DATA 44,196,32,61,196,32 :rem 209
50424 DATA 78,196,32,68,195,76 :rem 227
50430 DATA 0,192,30,134,24,142 :rem 182
50436 DATA 139,150,126,159,250,168 :rem 153
50442 DATA 6,179,172,189,243,200 :rem 51
50448 DATA 230,212,143,225,248,238 :rem 145
50454 DATA 46,253,66,74,82,90 :rem 166
50460 DATA 122,130,130,138,138,146 :rem 133
50466 DATA 146,154,170,178,186,194 :rem 165
50472 DATA 202,210,226,234,234,242 :rem 131
50478 DATA 234,234,125,192,149,192 :rem 158
50484 DATA 101,193,109,193,117,193 :rem 151
50490 DATA 125,193,248,192,224,192 :rem 155
50496 DATA 41,193,24,193,16,193 :rem 7
50502 DATA 234,234,83,90,91,98 :rem 211
50508 DATA 107,114,115,122,123,130 :rem 124
50514 DATA 147,154,187,194,211,218 :rem 152
50520 DATA 176,193,33,194,41,194 :rem 52
50526 DATA 197,193,49,194,235,193 :rem 120
50532 DATA 14,194,57,194,234,234 :rem 52
50538 DATA 234,234,234,234,234,234 :rem 145
50544 DATA 3,3,4,4,5,6 :rem 49
50550 DATA 6,7,7,1,1,2 :rem 45
50556 DATA 77,253,207,41,240,77 :rem 5
50562 DATA 253,207,145,253,96,234 :rem 103
50568 DATA 234,234,234,234,173,0 :rem 45
50574 DATA 220,74,176,15,160,50 :rem 250
50580 DATA 204,1,208,208,5,160 :rem 191
50586 DATA 241,140,1,208,206,1 :rem 190
50592 DATA 208,74,176,15,160,242 :rem 51
50598 DATA 204,1,208,208,5,160 :rem 200
50604 DATA 49,140,1,208,238,1 :rem 144
50610 DATA 208,74,176,15,162,24 :rem 250
50616 DATA 236,0,208,208,5,162 :rem 197
50622 DATA 255,142,0,208,206,0 :rem 186
50628 DATA 208,74,176,15,162,255 :rem 57
50634 DATA 236,0,208,208,5,162 :rem 197
50640 DATA 23,142,0,208,238,0 :rem 136
50646 DATA 208,174,244,207,160,0 :rem 43
50652 DATA 200,208,253,202,208,248 :rem 138
50658 DATA 96,169,1,162,1,160 :rem 159
50664 DATA 1,32,186,255,165,2 :rem 154
50670 DATA 162,0,160,199,32,189 :rem 2
50676 DATA 255,169,128,133,157,96 :rem 120
50682 DATA 169,0,166,251,164,252 :rem 52
50688 DATA 32,213,255,96,169,78 :rem 22
50694 DATA 166,253,164,254,32,216 :rem 105
50700 DATA 255,96,0,0,0,239 :rem 43

```

## Program 2: Sound Sculptor—Main Program

```

10 GOTO330 :rem 48
15 INPUTB,E:S=B*25+9758:F=9758+E*25+25:L= :rem 119
9

```

```

16 IFB<0ORE>125ORB>ETHENPRINT"BAD INPUT": :rem 246
GOTO15 :rem 246
20 PRINT"{CLR}{2 DOWN}":FORI=STOS+47STEP6 :rem 224
:rem 224
21 IFI>FTHENNEXT:L=3:PRINT"QK="QK":QB="QB :rem 126
":GOTO35":GOTO35 :rem 126
25 PRINTI;"DATA";:FORJ=0TO5:PRINTPEEK(I+J :rem 4
){LEFT},"":NEXTJ:PRINTCHR$(20):NEXTI :rem 4
30 PRINT"QK="QK"{LEFT}:QB="QB"{LEFT}:S="S :rem 70
+48"{LEFT}:F="F"{LEFT}:L="L"{LEFT}:GOT :rem 105
O20" :rem 121
35 POKEQK,L:FORK=1TOL:POKEQB+K,13:NEXTK:P :rem 66
RINT"{HOME}":END :rem 250
40 PRINT"{CLR}{2 DOWN}":FORM=0TO7:PRINTM: :rem 128
NEXTM :rem 210
45 POKEQK,8:FORK=1TO8:POKEQB+K,13:NEXTK:P :rem 214
RINT"{HOME}":END :rem 206
50 A=PEEK(61)+256*PEEK(62)+3:POKE786,INT( :rem 238
A/256):POKE785,A-256*PEEK(786):rem 250
55 IFERTHENPOKEA-2,0:POKEA-1,0:POKE45,PEE :rem 155
K(785):POKE46,PEEK(786) :rem 140
56 IFERTHENCLR:QK=198:QB=630:GOTO15 :rem 140
:rem 210
60 REM VOICE DISPLAY :rem 214
65 PRINT" VOICE#"V"UCCCCCCCCCI VOICE#"V:P :rem 206
OKE2,0 :rem 238
70 PRINT" UCCCCCCCCCKFREQUENCYJCCCCCCCCCI :rem 238
:rem 5
75 PRINT" BOCTAVE{7 SPACES}1 2 3 4 5 6 7 :rem 155
{SPACE}8B :rem 7
80 PRINT" B{28 SPACES}B :rem 155
85 PRINT" BNOTE C C D D E F F G G A A BB :rem 7
:rem 7
90 PRINT" B{7 SPACES}#{3 SPACES}# :rem 75
{5 SPACES}#{3 SPACES}#{3 SPACES}# :rem 6
{2 SPACES}B :rem 246
95 PRINT" JCCCCCCCCCCCCCCCCCCCCCCCCCCCCCK :rem 140
:rem 246
100 PRINT"{10 SPACES}UCCCCCCCCCI :rem 140
105 PRINT" UCCCCCCCCCKENVELOPEJCCCCCCCCCI :rem 157
:rem 224
110 PRINT" B{2 SPACES}ATTACK{4 SPACES}+++ :rem 157
+++++B :rem 124
115 PRINT" B{2 SPACES}DECAY{5 SPACES}++++ :rem 124
+++++B :rem 124
120 PRINT" B{2 SPACES}SUSTAIN{3 SPACES}++ :rem 124
+++++B :rem 124
125 PRINT" B{2 SPACES}RELEASE{3 SPACES}++ :rem 124
+++++B :rem 124
130 PRINT" JCCCCCCCCCCCCCCCCCCCCCCCCCCCCCK :rem 44
:rem 58
135 PRINT" UCCCCCCCCCCCCCCCCCCCCCCCCCCCCCI :rem 228
:rem 184
140 PRINT" B @M@M@M UCCCCCCCCCI{2 SPACES}N :rem 226
MNMNM{2 SPACES}B :rem 184
145 PRINT" B{8 SPACES}BWAVEFORMB :rem 226
{10 SPACES}B :rem 44
150 PRINT" B OLLOLOL JCCCCCCCCCK{2 SPACES}" :rem 135
:rem 47
151 PRINT"NOISE{3 SPACES}B{10 SPACES}B :rem 61
{28 SPACES}B" :rem 192
155 PRINT" BPULSE WIDTH ++++++B :rem 135
:rem 47
160 PRINT" JCCCCCCCCCCCCCCCCCCCCCCCCCCCCCK :rem 61
:rem 192
165 PRINT" UCCCCCCCCCCCCCCCCCCCCCCCCCCCCCI :rem 192
:rem 192
170 PRINT" BSYNCHRONIZATION{3 SPACES}USE :rem 192
{SPACE}VOICE B :rem 192

```



```

175 PRINT" BRING MODULATION{6 SPACES}#"SR
    ".{2 SPACES}B :rem 54
180 PRINT" JCCCCCCCCCCCCCCCCCCCCCCCCCK
    "; :rem 142
185 PRINT"HOME" :rem 130
190 A=(V-1)*7:S=S+A:POKE254,S/256:POKE253
    ,S-256*PEEK(254) :rem 221
195 SYSVCH :rem 127
200 GOTO455 :rem 105
205 PRINT"CLR";:POKE2,255 :rem 153
210 REM FILTER DISPLAY :rem 83
215 PRINT"[RVS]CCCCCCCCCCCCFILTER SETTING
    SCCCCCCCCCCCC[OFF]" :rem 79
220 PRINT"UCCCCCCCCCCCCICCCCCCCCCCCCCCI
    :rem 16
225 PRINT"BFILTER TYPEBBCUTOFF FREQUENCYB
    :rem 209
230 PRINT"B{11 SPACES}BB+++++++B
    :rem 252
235 PRINT"B HIGH PASS BJCCCCCCCCCCCCCK
    :rem 233
240 PRINT"B{11 SPACES}BUCCCCCCCCCCCCCCI
    :rem 151
245 PRINT"B BAND PASS BBVOICES{2 SPACES}F
    ILTEREDB :rem 182
250 PRINT"B{11 SPACES}BB 1{3 SPACES}2
    {3 SPACES}3{3 SPACES}E{2 SPACES}B
    :rem 41
255 PRINT"B LOW{2 SPACES}PASS BJCCCCCCCC
    CCCCCCK :rem 189
260 PRINT"JCCCCCCCCCKUCCCCCCCCCCCCCCI
    :rem 11
265 PRINT"{13 SPACES}B{3 SPACES}RESONANCE
    {4 SPACES}B :rem 110
270 PRINT"{13 SPACES}B+++++++B
    :rem 124
275 PRINT"{13 SPACES}JCCCCCCCCCCCCCK
    :rem 18
280 PRINT"[RVS]CCCCCCCCCCCCCCCCCCCCCCCC
    CCCCCCCCCCCCC[OFF]"; :rem 194
285 PRINT"{13 SPACES}UCCCCCCCCCCCCCCI
    :rem 28
290 PRINT"{13 SPACES}B{3 SHIFT-SPACE}
    {SHIFT-SPACE}VOLUME{5 SHIFT-SPACE}B
    :rem 70
295 PRINT"{13 SPACES}B+++++++B
    :rem 131
300 PRINT"{13 SPACES}JCCCCCCCCCCCCCK
    :rem 7
305 PRINT"{13 SPACES}UCCCCCCCCCCCCCCI
    :rem 21
310 PRINT"{13 SPACES}B[RVS]VOICE #3 OUTPU
    T[OFF] B :rem 40
315 PRINT"{13 SPACES}JCCCCCCCCCCCCCK
    :rem 13
320 S=S+21:POKE254,S/256:POKE253,S-256*PE
    EK(254):SYSFCH:GOTO455 :rem 247
325 REM INITIALIZATION :rem 166
330 SS=9758:POKE78,30:POKE79,38:SN=0:VCH=
    50360:FCH=50405:POKE53236,10 :rem 10
335 POKE53248,24:POKE53249,50:POKE51,29:P
    OKE52,38:POKE55,29:POKE56,38 :rem 5
340 PRINT"CLR" :rem 251
345 PRINT"{11 DOWN}"TAB(7)"WELCOME TO SOU
    ND SCULPTOR" :rem 127
350 FORL=1TO2000:NEXT :rem 23
355 PRINT"CLR" :rem 1
360 PRINT"{3 DOWN}"TAB(15)"[RVS]MAIN MENU
    [OFF]" :rem 109
365 PRINT"{2 DOWN}"TAB(14)"CHOOSE ONE:"
    :rem 63
370 PRINT"{2 DOWN}"TAB(7)"[RVS]F1[OFF] DE
    SIGN/REVIEW SOUNDS" :rem 226
375 PRINT:PRINTTAB(7)"[RVS]F3[OFF] LOAD S
    OUND FILE " :rem 124
380 GETAS:IFAS<"[F1]"ORAS>"[F3]"THEN380
    :rem 235
385 ONASC(AS)-132GOTO430,670 :rem 38
390 REM JOYSTICK SPEED :rem 97
395 PRINT"[CLR]{12 DOWN}{3 SPACES}SELECT
    {SPACE}A SPEED BETWEEN 0 AND 15."
    :rem 228
400 PRINT"{4 SPACES}0 - SLOWEST{6 SPACES}
    15 - FASTEST" :rem 160
405 INPUTPS :rem 204
410 IFPS<0ORPS>15THENPRINT"NUMBER NOT ACC
    EPTABLE":GOTO405 :rem 165
415 POKE53236,16-PS:GOTO455 :rem 64
420 STOP :rem 220
425 REM SOUND DESIGN/REVIEW :rem 195
430 PRINT"CLR" :rem 251
435 PRINT"{11 DOWN} WHICH SOUND DO YOU WI
    SH TO WORK ON?" :rem 177
440 PRINT"{2 SPACES}(NUMBER BETWEEN 0 & 1
    250 PLEASE)" :rem 67
445 INPUTSN :rem 206
450 IFSN<0ORSN>1250THENPRINT"NUMBER NOT A
    CCEPTABLE":GOTO445 :rem 11
455 POKE53269,0:PRINT"[CLR]{RVS}SOUND #";
    SN"[OFF]{HOME}{3 DOWN}"TAB(15)"CHOOSE
    ONE:" :rem 53
460 S=SS+SN*25 :rem 45
465 POKE79,S/256:POKE78,S-256*PEEK(79)
    :rem 186
470 PRINT:PRINTTAB(8)"[RVS]1[OFF] - DISPL
    AY VOICE #1" :rem 117
475 PRINT:PRINTTAB(8)"[RVS]2[OFF] - DISPL
    AY VOICE #2" :rem 124
480 PRINT:PRINTTAB(8)"[RVS]3[OFF] - DISPL
    AY VOICE #3" :rem 122
485 PRINT:PRINTTAB(8)"[RVS]4[OFF] - DISPL
    AY FILTER SETTINGS" :rem 235
490 PRINT:PRINTTAB(8)"[RVS]5[OFF] - CLEAR
    SOUND" :rem 139
495 PRINT:PRINTTAB(8)"[RVS]6[OFF] - NEW S
    OUND NUMBER" :rem 221
500 PRINT:PRINTTAB(8)"[RVS]7[OFF] - CHANG
    E JOYSTICK SPEED" :rem 28
505 PRINT:PRINTTAB(8)"[RVS]8[OFF] - QUIT"
    :rem 222
510 GETCS:IFCS<"1"ORCS>"8"THEN510 :rem 69
515 ONVAL(CS)GOTO520,525,530,535,540,430
    {SPACE},395,550 :rem 19
520 V=1:SR=3:POKE53269,1:GOTO65 :rem 87
525 V=2:SR=1:POKE53269,1:GOTO65 :rem 91
530 V=3:SR=2:POKE53269,1:GOTO65 :rem 89
535 POKE53269,1:GOTO205 :rem 60
540 FORL=0TO24:POKE5+L,0:NEXT:GOTO455
    :rem 91
545 REM QUIT :rem 197
550 PRINT"CLR">{7 DOWN}" :rem 117
555 PRINT TAB(14)"CHOOSE ONE:" :rem 218
560 PRINT:PRINTTAB(6)"[RVS]F1[OFF] - SAVE
    SOUND FILE" :rem 177
565 PRINT:PRINTTAB(6)"[RVS]F3[OFF] - CONV
    ERT TO DATA STATEMENTS" :rem 198
570 PRINT:PRINTTAB(6)"[RVS]F5[OFF] - END"
    :rem 181
575 GETAS:IFAS<"[F1]"ORAS>"[F5]"THEN575
    :rem 248
580 ONASC(AS)-132GOTO610,585,685:rem 247
585 PRINT"CLR">{8 DOWN}" :rem 142

```



```

590 PRINT"[2 SPACES]ENTER SOUNDS YOU WANT
    TO CONVERT" :rem 196
595 PRINT"[6 SPACES](START,END)"; :rem 145
600 ER=1:GOTO50 :rem 115
605 REM SAVE SOUNDS ROUTINE :rem 176
610 PRINT"[CLR]{8 DOWN}" :rem 131
615 PRINT"[2 SPACES]ENTER SOUNDS YOU WISH
    TO SAVE" :rem 209
620 PRINT"[6 SPACES](START,END)"; :rem 134
625 INPUTB,E:IFB<00RE>12500RB>ETHENPRINT"
    BAD INPUT":GOTO625 :rem 16
630 S=B*25+9758:F=9758+E*25+25 :rem 49
635 POKE79,S/256:POKE78,S-256*PEEK(79):PO
    KE254,F/256:POKE253,F-256*PEEK(254)
    :rem 117
640 INPUT"WHAT DO YOU WISH TO NAME THE FI
    LE";NM$:IFNM$=""THEN640 :rem 254
645 T=LEN(NM$):POKE2,T :rem 58
650 FORJ=1TOT:POKE50944-J+T,ASC(RIGHT$(NM
    $,J)):NEXTJ :rem 213
655 SYS50659 :rem 168
660 SYS50692 :rem 161
665 PRINT:PRINTNM$" FILE HAS BEEN SAVED":
    PRINT"THANK YOU":END :rem 16
670 REM LOAD ROUTINE :rem 199
671 IFPEEK(50660)=1THEN PRINT"[CLR]":POKE
    2,0:SYS50659:SYS50682:GOTO430 :rem 38
673 INPUT"[7 RIGHT]FILENAME";NM$:T=LEN(NM
    $):POKE2,T:IFT=0THEN673 :rem 168
674 FOR J=1TOT:POKE50944-J+T,ASC(RIGHT$(N
    M$,J)):NEXTJ :rem 219
675 SYS50659:SYS50682 :rem 232
676 IF ST=66 THENPRINT"[7 RIGHT]FILE NOT
    [SPACE]FOUND":GOTO673 :rem 237
679 GOTO430 :rem 118
680 REM END :rem 89
685 PRINT"[CLR]THANK YOU":END :rem 139

```

```

10 R=INT(RND(1)*6)+1:A2$=MID$(A$,R,1):IFA
    1$=A2$THEN10 :rem 174
11 A2=VAL(A2$) :rem 221
12 R=INT(RND(1)*6)+1:A3$=MID$(A$,R,1):IFA
    1$=A3$ORA2$=A3$THEN12 :rem 193
13 A3=VAL(A3$) :rem 225
14 R=INT(RND(1)*6)+1:A4$=MID$(A$,R,1):IFA
    1$=A4$ORA2$=A4$ORA3$=A4$THEN14:rem 215
15 A4=VAL(A4$) :rem 229
16 X$="":POKE1,135:FORL=1TO100:NEXTL:POK
    ES1,0:INPUT"SELECT COLORS";X$ :rem 138
17 IFLEN(X$)<>4THENGOSUB106:GOTO93 :rem 118
18 FORE=1TO4:V=VAL(MID$(X$,E,1)) :rem 116
19 IFV<1ORV>6THENGOSUB106:GOTO93 :rem 76
20 NEXTE :rem 230
21 X=X+1:B=0:W=0:AA$=A1$+A2$+A3$+A4$
    :rem 143
22 FORJ=1TO4 :rem 218
23 G(J)=VAL(MID$(X$,J,1)) :rem 86
24 C(J)=VAL(MID$(AA$,J,1)) :rem 125
25 IFG(J)=C(J)THENB=B+1:G(J)=0:C(J)=0
    :rem 77
26 NEXTJ :rem 241
27 FORJ=1TO4:IFC(J)=0THEN33 :rem 136
28 H=0:FORK=1TO4 :rem 208
29 IFC(J)=0THEN32 :rem 217
30 IFC(J)<>G(K)THEN32 :rem 193
31 H=1:G(K)=0:C(J)=0 :rem 41
32 NEXTK:W=W+H :rem 135
33 NEXTJ :rem 239
34 ONXGOTO35,36,37,38,39,40,41,42,43,44
    :rem 49
35 PRINT"[HOME] 1 ";:GOTO45 :rem 148
36 PRINT"[HOME]{2 DOWN} 2 ";:GOTO45
    :rem 184
37 PRINT"[HOME]{4 DOWN} 3 ";:GOTO45
    :rem 220
38 PRINT"[HOME]{6 DOWN} 4 ";:GOTO45:rem 0
39 PRINT"[HOME]{8 DOWN} 5 ";:GOTO45
    :rem 36
40 PRINT"[HOME]{10 DOWN} 6 ";:GOTO45
    :rem 63
41 PRINT"[HOME]{12 DOWN} 7 ";:GOTO45
    :rem 99
42 PRINT"[HOME]{14 DOWN} 8 ";:GOTO45
    :rem 135
43 PRINT"[HOME]{16 DOWN} 9 ";:GOTO45
    :rem 171
44 PRINT"[HOME]{18 DOWN}10 "; :rem 26
45 X1=VAL(LEFT$(X$,1)):X2=VAL(MID$(X$,2,1
    )):X3=VAL(MID$(X$,3,1)):X4=VAL(RIGHT$(
    X$,1)) :rem 87
46 P=0:T=0 :rem 34
47 P=P+1:ONX1GOTO51,52,53,54,55,56
    :rem 140
48 P=P+1:ONX2GOTO51,52,53,54,55,56
    :rem 142
49 P=P+1:ONX3GOTO51,52,53,54,55,56
    :rem 144
50 P=P+1:ONX4GOTO51,52,53,54,55,56
    :rem 137
51 PRINT"[BLK]{RVS}{2 SPACES}{OFF} ";:POK
    ES2,135:GOTO57 :rem 56
52 PRINT"[WHT]{RVS}{2 SPACES}{OFF} ";:POK
    ES2,159:GOTO57 :rem 180
53 PRINT"[RED]{RVS}{2 SPACES}{OFF} ";:POK
    ES2,175:GOTO57 :rem 202

```

## Mind Boggle

(Article on page 60.)

### BEFORE TYPING...

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

### Program 1: vic version

```

1 PRINT"[CLR]{5 RIGHT}{10 DOWN}MIND BOGGL
    E":CLR :rem 195
2 FOR T= 1 TO 2000:NEXT :rem 185
3 DIMC(4),G(4) :rem 205
4 S1=36875:S2=S1+1:POKES1+3,15:POKES1+4,1
    20:X=0 :rem 197
5 PRINT"[CLR]{19 DOWN}" :rem 220
6 PRINT"[2 SPACES]{BLK}{RVS} 1{OFF} {WHT}
    {RVS} 2{OFF} {RED}{RVS} 3{OFF} {CYN}
    {RVS} 4{OFF} {PUR}{RVS} 5{OFF} {GRN}
    {RVS} 6{OFF}{BLK}" :rem 174
7 PRINT"[BLK]I CHOOSE 4 COLORS NOW":FORL=
    1TO100:POKES2,INT(RND(1)*128)+128:FORM=
    1TO10 :rem 210
8 NEXTM:NEXTL:POKES2,0:GOSUB106 :rem 36
9 A$="123456":R=INT(RND(1)*6)+1:A1$=MID$(
    A$,R,1):A1=VAL(A1$) :rem 202

```



```

54 PRINT"{CYN}{RVS}{2 SPACES}{OFF} ";:POK
   ES2,191:GOTO57 :rem 76
55 PRINT"{PUR}{RVS}{2 SPACES}{OFF} ";:POK
   ES2,201:GOTO57 :rem 66
56 PRINT"{GRN}{RVS}{2 SPACES}{OFF} ";:POK
   ES2,209:GOTO57 :rem 205
57 FORL=1TO99:NEXTL:POKES2,0:FORL=1TO250:
   NEXTL:ONPGOTO48,49,50 :rem 1
58 ONTGOTO79,80,81,82 :rem 198
59 PRINT" ";:IFB=0THEN66 :rem 141
60 GOSUB91 :rem 80
61 ONBGOTO62,63,64,65 :rem 169
62 PRINT"{BLK}Q";:GOTO66 :rem 180
63 PRINT"{BLK}QQ";:GOTO66 :rem 134
64 PRINT"{BLK}QQQ";:GOTO66 :rem 88
65 PRINT"{BLK}QQQQ";:GOTO95 :rem 44
66 IFW=0THEN73 :rem 88
67 GOSUB92 :rem 88
68 ONWGOTO69,70,71,72 :rem 198
69 PRINT"{BLK}{RVS}Q{OFF}":GOTO73 :rem 34
70 PRINT"{BLK}{RVS}QQ{OFF}":GOTO73 :rem 235
71 PRINT"{BLK}{RVS}QQQ{OFF}":GOTO73 :rem 189
72 PRINT"{BLK}{RVS}QQQQ{OFF}": :rem 178
73 PRINT"{BLK}{HOME}{20 DOWN}":PRINT"
   {21 SPACES}" :rem 61
74 PRINT"{HOME}{20 DOWN}" :rem 163
75 IFX<>10THEN16 :rem 196
76 FORL=1TO15:FORM=200TO220+L*2:POKES2,M:
   NEXTM:NEXTL:POKES2,0 :rem 154
77 PRINT"{HOME}{19 DOWN}":PRINT"
   {3 SPACES}"; :rem 219
78 T=T+1:ONA1GOTO51,52,53,54,55,56 :rem 129
79 T=T+1:ONA2GOTO51,52,53,54,55,56 :rem 131
80 T=T+1:ONA3GOTO51,52,53,54,55,56 :rem 124
81 T=T+1:ONA4GOTO51,52,53,54,55,56 :rem 126
82 PRINT"{4 SPACES}":PRINT"{BLK}CORRECT C
   OLORS RETURN" :rem 154
83 IFPEEK(197)<>15THEN83 :rem 139
84 GETT$:PRINT"{RED}{CLR}{2 DOWN} TOO BAD
   YOU MISSED!":PRINT"{2 DOWN} {BLU}10 T
   RIES IS ENOUGH." :rem 0
85 FORL=1TO6:POKES2,160:FORM=1TO400:NEXTM
   :POKES2,0:FORM=1TO400:NEXTM:NEXTL :rem 89
86 GETT$:PRINT"{CLR}{2 DOWN}{BLK}
   {2 SPACES}WANT TO PLAY AGAIN?":PRINT:P
   RINT:PRINT"{5 SPACES}YES OR NO?" :rem 207
87 IFPEEK(197)=11THENGETT$:GOTO4 :rem 226
88 IFPEEK(197)=28THENPOKE36879,27:GOTO90 :rem 157
89 GOTO87 :rem 25
90 GETT$:PRINT"{CLR}{9 DOWN}{8 SPACES}
   {YEL}{RVS}CHICKEN{OFF}":PRINT:PRINT:ST
   OP :rem 61
91 FORL=200TO254:POKES2,L:NEXTL:POKES2,0:
   RETURN :rem 57
92 FORL=200TO128STEP-1:POKES1,L:NEXTL:POK
   ES1,0:RETURN :rem 210
93 PRINT"ILLEGAL INPUT!":POKES1+2,200:FOR
   L=1TO500:NEXTL:POKES1+2,0 :rem 110
94 FORL=1TO999:NEXTL:GOSUB106:GOTO16 :rem 76

```

```

95 PRINT:PRINT:PRINT"{GRN}{2 SPACES}YOU W
   I N I RETURN" :rem 248
96 FORM=250TO240STEP-1:POKES2,M:NEXTM:FOR
   M=240TO250:POKES2,M:NEXTM:POKES2,0 :rem 105
97 IFPEEK(197)<>15THEN96 :rem 148
98 PRINT"{CLR}{5 DOWN}" :rem 42
99 IFX=1THENPRINT"{RED}{5 SPACES}LUCKY GU
   ESS!":GOTO104 :rem 27
100 IFX=2ORX=3THENPRINT"{GRN}{6 SPACES}EX
   PERT!!!":GOTO104 :rem 177
101 IFX=4ORX=5ORX=6THENPRINT"{BLU}
   {4 SPACES}PRETTY GOOD!":GOTO104 :rem 26
102 IFX=7ORX=8THENPRINT"{PUR}{9 SPACES}SO
   SO!":GOTO104 :rem 101
103 PRINT"{2 SPACES}YOU BARELY GOT IT!" :rem 201
104 FORL=1TO50:FORM=248TO253:POKES2,M:NEX
   TM:FORM=253TO248STEP-1:POKES2,M:NEXTM
   :NEXTL :rem 3
105 POKES2,0:GOTO86 :rem 135
106 PRINT"{2 UP}":PRINT"{21 SPACES}":PRIN
   T"{2 UP}":RETURN :rem 220

```

## Program 2: 64 Version

```

100 DIMC(4),G(4):POKE53281,15 :rem 41
101 SO = 54272:FOR T=SO TO SO+24:POKET,0:
   NEXT:X=0 :rem 48
102 POKESO+24,15:POKESO+5,17:POKESO+6,241 :rem 181
300 PRINT"{CLR}{13 DOWN}{14 RIGHT}{BLK}MI
   ND BOGGLE{5 DOWN}":FORT=1TO 1000:NEXT :rem 231
340 PRINT"{CLR}{20 DOWN}" :rem 79
350 PRINT "{4 RIGHT}{BLK} {RVS}{2 SPACES}
   1 {OFF} {WHT}{RVS}{2 SPACES}2 {OFF}
   {RED}{RVS}{2 SPACES}3 {OFF} {BLU}
   {RVS}{2 SPACES}4 {OFF} {PUR}{RVS}
   {2 SPACES}5 {OFF} {GRN}{RVS}
   {2 SPACES}6 {OFF}{BLK}" :rem 4
365 PRINTSPC(6)"{BLK}I AM CHOOSING 4 COLO
   RS NOW{BLK}":FORL=1TO100 :rem 54
366 POKES2,INT(RND(1)*128)+128:FOR M=1 TO
   10:NEXTM:NEXTL:POKES2,0:GOSUB4050 :rem 228
400 A$="123456":R=INT(RND(1)*6)+1:A1$=MID
   $(A$,R,1):A1=VAL(A1$) :rem 37
401 R=INT(RND(1)*6)+1:A2$=MID$(A$,R,1):IF
   A1$=A2$THEN401 :rem 22
402 A2=VAL(A2$) :rem 17
403 R=INT(RND(1)*6)+1:A3$=MID$(A$,R,1):IF
   A1$=A3$ORA2$=A3$THEN403 :rem 41
404 A3=VAL(A3$) :rem 21
405 R=INT(RND(1)*6)+1:A4$=MID$(A$,R,1):IF
   A1$=A4$ORA2$=A4$ORA3$=A4$THEN405 :rem 63
406 A4=VAL(A4$) :rem 25
500 FOR T= 1TO 1000:NEXT :rem 27
510 POKESO,100:POKESO+1,100:POKESO+4,17:F
   ORL=1TO100:NEXTL:POKESO+4,16 :rem 23
540 POKE214,21:POKE211,5:PRINT:PRINT"
   {34 SPACES}" :rem 30
550 X$="":POKE214,21:POKE211,0:PRINT:INPU
   T "SELECT COLORS":X$ :rem 157
600 IFLEN(X$)<>4THENGOSUB4050:GOTO3300 :rem 48
700 FORE=1TO4:V=VAL(MID$(X$,E,1)):rem 162
705 IFV<1ORV>6THENGOSUB4050:GOTO3300 :rem 10

```



```

710 NEXTE                                     :rem 28
900 X=X+1:B=0:W=0:AA$=A1$+A2$+A3$+A4$      :rem 197
901 FORJ=1TO4                                :rem 16
902 G(J)=VAL(MID$(X$,J,1))                  :rem 140
903 C(J)=VAL(MID$(AA$,J,1))                 :rem 179
904 IFG(J)=C(J)THENB=B+1:G(J)=0:C(J)=0      :rem 131
905 NEXTJ                                     :rem 39
906 FORJ=1TO4:IFC(J)=0THEN912               :rem 244
907 H=0:FORK=1TO4                            :rem 6
908 IFC(J)=0THEN911                          :rem 69
909 IFC(J)<>G(K)THEN911                      :rem 54
910 H=1:G(K)=0:C(J)=0                      :rem 95
911 NEXTK:W=W+H                             :rem 189
912 NEXTJ                                     :rem 37

1000 ONXGOTO1100,1105,1110,1115,1120,1125
      ,1130,1135,1140,1145                  :rem 60
1100 PRINT"[HOME]{7 RIGHT}{DOWN} 1 ";:GOT
      O1200                                  :rem 36
1105 PRINT"[HOME]{7 RIGHT}{2 DOWN} 2 ";:G
      O1200                                  :rem 59
1110 PRINT"[HOME]{7 RIGHT}{3 DOWN} 3 ";:G
      O1200                                  :rem 73
1115 PRINT"[HOME]{7 RIGHT}{4 DOWN} 4 ";:G
      O1200                                  :rem 96
1120 PRINT"[HOME]{7 RIGHT}{5 DOWN} 5 ";:G
      O1200                                  :rem 110
1125 PRINT"[HOME]{7 RIGHT}{6 DOWN} 6 ";:G
      O1200                                  :rem 133
1130 PRINT"[HOME]{7 RIGHT}{7 DOWN} 7 ";:G
      O1200                                  :rem 147
1135 PRINT"[HOME]{7 RIGHT}{8 DOWN} 8 ";:G
      O1200                                  :rem 170
1140 PRINT"[HOME]{7 RIGHT}{9 DOWN} 9 ";:G
      O1200                                  :rem 184
1145 PRINT"[HOME]{7 RIGHT}{10 DOWN}10 ";
      :rem 192
1200 X1=VAL(LEFT$(X$,1)):X2=VAL(MID$(X$,2
      ,1)):X3=VAL(MID$(X$,3,1))              :rem 174
1210 X4=VAL(RIGHT$(X$,1))                  :rem 141
1220 P=0:T=0                               :rem 125
1300 P=P+1:ONX1GOTO1700,1701,1702,1703,17
      04,1705                                :rem 49
1400 P=P+1:ONX2GOTO1700,1701,1702,1703,17
      04,1705                                :rem 51
1500 P=P+1:ONX3GOTO1700,1701,1702,1703,17
      04,1705                                :rem 53
1600 P=P+1:ONX4GOTO1700,1701,1702,1703,17
      04,1705                                :rem 55
1700 PRINT"[BLK]{RVS}{3 SPACES}{OFF} ";:P
      OKESO+1,100:GOTO1750                   :rem 108
1701 PRINT"[WHT]{RVS}{3 SPACES}{OFF} ";:P
      OKESO+1,124:GOTO1750                   :rem 232
1702 PRINT"[RED]{RVS}{3 SPACES}{OFF} ";:P
      OKESO+1,140:GOTO1750                   :rem 254
1703 PRINT"[BLU]{RVS}{3 SPACES}{OFF} ";:P
      OKESO+1,166:GOTO1750                   :rem 10
1704 PRINT"[PUR]{RVS}{3 SPACES}{OFF} ";:P
      OKESO+1,150:GOTO1750                   :rem 129
1705 PRINT"[GRN]{RVS}{3 SPACES}{OFF} ";:P
      OKESO+1,185:GOTO1750                   :rem 12
1750 POKESO+4,17:FORL=1TO99:NEXTL:POKESO+
      4,16:FORL=1TO250:NEXTL                 :rem 29
1751 ONPGOTO1400,1500,1600                 :rem 158
1761 ONTGOTO2072,2073,2074,2075           :rem 175
1800 PRINT " ";:IFB=0THEN2000              :rem 62
1810 GOSUB3100                              :rem 14
1820 ONBGOTO1900,1901,1902,1903           :rem 149
1900 PRINT"[BLK]Q";:GOTO2000              :rem 108

1901 PRINT"[BLK]QQ";:GOTO2000              :rem 62
1902 PRINT"[BLK]QQQ";:GOTO2000            :rem 16
1903 PRINT"[BLK]QQQQ";:GOTO4000           :rem 228
2000 IFW=0THEN2060                          :rem 12
2010 GOSUB3200                              :rem 8
2020 ONWGOTO2050,2051,2052,2053           :rem 151
2050 PRINT"[BLK]{RVS}Q[OFF]";:GOTO2060
      :rem 216
2051 PRINT"[BLK]{RVS}QQ[OFF]";:GOTO2060
      :rem 170
2052 PRINT"[BLK]{RVS}QQQ[OFF]";:GOTO2060
      :rem 124
2053 PRINT"[BLK]{RVS}QQQQ[OFF]"           :rem 19
2060 PRINT"[BLK]{HOME}[15 DOWN]";:PRINT"
      {21 SPACES}"                          :rem 70
2062 PRINT"[HOME]{9 RIGHT}[15 DOWN]"
      :rem 178
2065 IFX<>10THEN500                        :rem 83
2066 POKESO+4,32:FORL=1TO15:FORM=200TO220
      +L*2:POKESO+1,M:NEXTM:NEXTL           :rem 37
2068 POKESO+4,32                            :rem 145
2070 PRINT"[HOME]{14 DOWN}";:PRINT"
      {10 SPACES}";                        :rem 225
2071 T=T+1:ONALGOTO1700,1701,1702,1703,17
      04,1705                                :rem 40
2072 T=T+1:ONA2GOTO1700,1701,1702,1703,17
      04,1705                                :rem 42
2073 T=T+1:ONA3GOTO1700,1701,1702,1703,17
      04,1705                                :rem 44
2074 T=T+1:ONA4GOTO1700,1701,1702,1703,17
      04,1705                                :rem 46
2075 PRINT"{4 SPACES}";:PRINT"[BLK]
      {5 SPACES}CORRECT COLORS{2 SPACES}PR
      ESS RETURN"                          :rem 139
2080 IFPEEK(197)<>1THEN2080                :rem 20
2085 GETT$:PRINT"[RED]{CLR}{9 DOWN}
      {10 RIGHT}TOO BAD YOU MISSED!"
      :rem 56
2086 PRINT"[2 DOWN]{10 RIGHT}{BLU}10 TRIE
      S IS ENOUGH."                        :rem 124
2090 POKESO+4,33                            :rem 141
2096 FORL=1TO6:POKESO+1,80:FORM=1TO400:NE
      XTM:POKESO+4,32:FORM=1TO400:NEXTM
      :rem 243
2098 NEXTL:POKESO+4,32                      :rem 89
3000 GETT$:PRINT"[CLR]{8 DOWN}{9 RIGHT}
      {BLK} WANT TO PLAY AGAIN?{2 DOWN}"
      :rem 170
3005 PRINT "[15 RIGHT]YES OR NO?";:rem 186
3010 IFPEEK(197)=25THENGETT$:POKE198,0:GO
      TO101                                  :rem 1
3012 IFPEEK(197)=39THEN3014                :rem 12
3013 GOTO3010                              :rem 196
3014 GETT$:PRINT"[CLR]{10 DOWN}[15 RIGHT]
      {BLK}CHICKEN!!"{2 SPACES}:PRINT:PRIN
      T:END                                :rem 129
3100 POKESO+4,17:FORL=100TO154:POKESO+1,L
      :NEXTL:POKESO+4,16:RETURN             :rem 187
3200 POKESO+4,17:FORL=100TO48STEP-1:POKE
      S O+1,L:NEXTL:POKESO+4,16:RETURN
      :rem 40
3300 PRINT"[UP]ILLEGAL INPUT{UP}";:FORL=1
      TO500:NEXTL                           :rem 60
3301 FORL=1TO999:NEXTL:GOSUB4050:GOTO500
      :rem 6
4000 PRINT:PRINT:PRINT"[BLK]{7 RIGHT}YOU
      {SPACE}W I N !! PRESS RETURN";:rem 24
4005 POKESO+4,33:FORM=180TO100STEP-5:POKE
      S 2,M:NEXTM:FORM=100TO200STEP 5
      :rem 176

```



```

4006 POKESO+1,M:NEXT M:POKE SO+4,32
                                     :rem 49
4010 IFPEEK(197)<>1THEN4005         :rem 14
4011 PRINT"{CLR}{10 DOWN}{9 RIGHT}";
                                     :rem 20
4015 IFX=1THENPRINT"{RED}{5 SPACES}LUCKY
      {SPACE}GUESS!":GOTO4020       :rem 164
4016 IFX=2ORX=3THENPRINT"{GRN}{6 SPACES}E
      XPERT!!!":GOTO4020             :rem 28
4017 IFX=4ORX=5ORX=6THENPRINT"{BLU}
      {4 SPACES}PRETTY GOOD!":GOTO4020
                                     :rem 133
4018 IFX=7ORX=8THENPRINT"{PUR}{9 SPACES}S
      O SO!":GOTO4020               :rem 208
4019 PRINT"{2 SPACES}YOU BARELY GOT IT!"
                                     :rem 3
4020 POKESO+4,33:FORL=1TO20:FORM=148TO200
      STEP5:POKES2,M:NEXTM          :rem 131
4021 FOR M=200TO 148STEP-5:POKESO+1,M:NEX
      TM:NEXTL:POKESO+4,32          :rem 14
4030 GOTO3000                       :rem 195
4050 PRINT"{32 SPACES}":RETURN      :rem 180

```

#### 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.

### COMPUTE!'s Gazette Subscriber Services

Please help us serve you better. If you need to contact us for any of the reasons listed below, write to us at:

**COMPUTE!'s Gazette**  
P.O. Box 961  
Farmingdale, NY 11737

or call the Toll Free number listed below.

**Change of Address.** Please allow us 6-8 weeks to effect the change; send your current mailing label along with your new address.

**Renewal.** Should you wish to renew your Gazette subscription before we remind you to, send your current mailing label with payment or charge number or call the Toll Free number listed below.

**New Subscription.** A one-year (12-month) U.S. subscription to *COMPUTE!'s Gazette* is \$20 (2 years, \$36; 3 years, \$54. For subscription rates outside the U.S., see staff page). Send us your name and address or call the Toll Free number listed below.

**Delivery Problems.** If you receive duplicate issues of *COMPUTE!'s Gazette*, if you experience late delivery, or if you have problems with your subscription, please call the Toll Free number listed below.

**COMPUTE!'s Gazette**  
**800-334-0868**  
**In NC 919-275-9809**

## This Publication is available in Microform.



## University Microfilms International

Please send additional information

for \_\_\_\_\_ (name of publication)

Name \_\_\_\_\_

Institution \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_

State \_\_\_\_\_ Zip \_\_\_\_\_

300 North Zeeb Road  
Dept. P.R.  
Ann Arbor, Mi. 48106



# 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 I` 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.



# COMPUTE!'s First Book of VIC Games

**Authors:** COMPUTE! Magazine Contributors

**Price:** \$12.95

**On Sale:** Now

The VIC-20 is a versatile computer. Its programmable color, graphics, and sound can add a lot to game programs.

Over the past few years, COMPUTE! Magazine has published a wide variety of games for the VIC. But some back issues are hard to find or unavailable.

That's why the editors of COMPUTE! have chosen the best games, revised them, and added previously unpublished games, putting them together into *COMPUTE!'s First Book of VIC Games*. Each game has been tested and debugged and is ready to type into a VIC-20.

*First Book of VIC Games* contains fast-action arcade games that require quick reflexes, as well as strategy games that test logical skills. For \$12.95 you get two dozen games, including:

**Time Bomb** — You hear the time bomb ticking, and you have to find your way through the maze to defuse it. But the maze is larger than the screen, and you can't always see where the dead ends are.

**Closeout** — The local department store is having a sale. You try to snatch as many sale items as possible, while avoiding the horde of bargain hunters who are trying to thwart you.

**Cryptic Numbers** — A good logic game. The computer picks a pattern of numbers that you have to guess. After each turn you learn how close you were.

**Air Defense** — Missiles are falling out of the sky onto your city. Aim carefully; you get only one shot at each missile.

**Thunderbird** — Your goal is to break out of the playing field by using the thunderbird that controls the satellite.

**Sky Diver** — Put on your parachute, jump out of the plane, and try to land on one of the targets. Watch out for wind currents that may blow you off course.

**Deflector** — A ball is bouncing around the screen. You can aim it toward targets by strategically placing deflectors that change its direction.

**Hidden Maze** — Lost in a maze, you're trying to get out. But it is dark, and you can see only a few spaces in front of you.

**Outpost** — Your small fortress is under siege. You have two types of lasers and some torpedoes. Unfortunately, the energy supply is dwindling and the computer is on the blink. The supply ship may (or may not) show up in time to make repairs.

*First Book of VIC Games* is more than just a book of program listings. Each program is annotated and explained; you can modify the games if you like or use the many programming techniques in your own games. Also included is a useful program you can use to draw mazes for games you write. Three chapters show you how to develop a game program. Another tells you how to take advantage of the VIC's sound, graphics, and color capabilities. The index lists references where you can learn more about programming. And *First Book of VIC Games* is spiral bound to lie flat while you are typing in programs.

To order directly, call  
**TOLL FREE**  
**800-334-0868**  
(919-275-9809 In NC)

## ORDER NOW

Available at computer dealers and bookstores nationwide. Or send a check or money order to COMPUTE! Books, P. O. Box 5406, Greensboro, NC 27403.

Add \$2 shipping and handling. Outside the U.S. add \$5 for air mail, \$2 for surface mail. All orders prepaid, U.S. funds only.  
© 1983, COMPUTE! Publications, Inc. VIC-20 is a trademark of Commodore Electronics Ltd.

Copy Cat	
Mark and Dan Powell	94
Outpost	
Tim Parker	97
Cryptic Numbers	
C.G. McGaffin	105
Word Hunt	
M Eric Jansing and Bob Meyers, Jr.	112
Lost Fox	
Warren Pugh	120
Pharaoh's Treasure	
Clark and Kathryn H. Kidd	128
Part 5: Scrolling	
Grand Prix Foo	135
J Mark Vittek	137
Part 6: Dexterity	
Thunderbird	143
Dave Sanders	
Juggler	145
J Doug Ferguson	153
Deflector	
Frank J. Tyniwo	158
Jumping Jack	
Paul Burger	
Skydiver	
J Alan Crossley	
The Hawkmen of Dindrin	
J Esteban V. Aguilar, Jr.	
Appendix A: Creating Your Own	
Charles Bond	
Appendix B: Writing Your Own	
Dan Carmichael	
Appendix C: A Beginner's Guide	
Typing In Programs	
Listing Conventions	
Index	
Special Requirements: J=joystick M=memory expansion	

## Table of Contents

Foreword	v
Part 1: Playing Games With Your VIC	1
VIC Features: Color, Graphics, Sound, etc.	
Dan Carmichael	3
Writing Your First Game	
Richard Mansfield	9
Writing A Simulation Game	
Richard Mansfield	14
Writing An Arcade Game	
Richard Mansfield	19
Part 2: Maze Games	23
Time Bomb	
J Doug Smoak	25
Hidden Maze	
J Gary Boden	29
(Translated for the VIC by Charles Brannon)	
Superchase	
J Anthony Godshall	33
Lochinvar's Maze	
Clark and Kathryn H. Kidd	38
Part 3: Action Games	43
Closeout	
L.L. Beh	45
(Translated for the VIC by Charles Brannon)	
Marble Hunt	
J Ronny Ong	51
Balloons	
Aaron Bobick	55
Richthofen's Revenge	
J Marc Sugiyama, Todd Koumrian, Chris Metcalf	59
Chameleon	
J Clark and Kathryn H. Kidd	75
Air Defense	
T.L. Wahl	80
Part 4: Brain Testers	
MathMan	87
Andy Hayes	89
Special Requirements: J=joystick M=memory expansion	



# COMPUTE!'s First Book Of VIC

**Authors:** COMPUTE! Magazine contributors

**Price:** \$12.95

**On Sale:** Now

Finally, it's VIC's turn!

Users of other popular personal computers have been enjoying their **COMPUTE! Books**: *COMPUTE!'s First Book Of PET/CBM*, *Programming The PET/CBM*, and others.

Now, there's a book devoted exclusively to the Commodore VIC-20™ Computer: *COMPUTE!'s First Book Of VIC*.

The editors of **COMPUTE!** Magazine – the leading resource for the VIC-20 – gathered together the best VIC-20 articles published since the summer of 1981 and added some new material. The result is more than 200 pages of valuable information – information that goes beyond the instruction manuals. In the **COMPUTE!** tradition, it is carefully edited to be easily understood and useful for beginners and experts alike.

*COMPUTE!'s First Book Of VIC* is spiral-bound to lie flat, and includes ready-to-type program listings and articles such as "The Joystick Connection: Meteor Maze," "STARFIGHT3," "Train Your PET To Run VIC Programs," "Renummer BASIC Lines The Easy Way," "High Resolution Plotting," "Custom Characters For The VIC," "VIC Memory – The Uncharted Adventure," and "A Simple Monitor For The VIC."

At only \$12.95, less than most computer manuals, *COMPUTE!'s First Book Of VIC* is among the best resources a VIC user can own.

Available at computer dealers and bookstores nationwide. To order directly call TOLL FREE 800-334-0868. In North Carolina call 919-275-9809. Or send check or money order to **COMPUTE! Books**, P.O. Box 5406, Greensboro, NC 27403.

Add \$2 shipping and handling. Outside the U.S. add \$5 for air mail, \$2 for surface mail. All orders prepaid, U.S. funds only. VIC-20 is a trademark of Commodore Electronics Limited.

v Introduction .....	Robert Lock
<b>Chapter One: Getting Started.</b>	
3 The Story Of The VIC .....	Michael S. Tomczyk
11 Computer Genesis: From Sticks And Stones To VIC .....	Dorothy Kunkin Heller / David Thornburg
20 Super Calculator .....	Jim Butterfield
24 Large Alphabet .....	Doug Ferguson
26 Using A Joystick .....	David Malmberg
39 Extended Input Devices: Paddles And The Keyboard .....	Mike Bassman / Salomon Lederman
46 Game Paddles .....	David Malmberg
<b>Chapter Two: Diversions – Recreation And Education.</b>	
59 The Joystick Connection: Meteor Maze .....	Paul L. Bupp / Stephen P. Drop
67 ZAPII .....	Dub Scroggin
72 STARFIGHT3 .....	David R. Mizner
78 Alphabetizer .....	Jim Wilcox
80 Count The Hearts .....	Christopher J. Flynn
<b>Chapter Three: Programming Techniques.</b>	
89 PRINTing With Style .....	James P. McCallister
97 Train Your PET To Run VIC Programs .....	Lyle Jordan
99 User Input .....	Wayne Kozun
103 Amortize .....	Amihai Glazer
106 Append .....	Wayne Kozun
109 Printing The Screen .....	C. D. Lane
113 The Confusing Quote .....	Charles Brannon
115 Alternate Screens .....	Jim Butterfield
119 Timekeeping .....	Keith Schleiffer
125 Renummer BASIC Lines The Easy Way .....	Charles H. Gould
127 Automatic Line Numbers .....	Jim Wilcox
129 Putting The Squeeze On Your VIC-20: Getting The Most Out Of 5000 Bytes .....	Stanley M. Berlin
141 An Easy Way To Relocate VIC Programs On Other Commodore Computers .....	Greg and Ross Sherwood
<b>Chapter Four: Color And Graphics.</b>	
147 Kaleidoscope And Variations .....	Kenneth Knox
148 High Resolution Plotting .....	Paul F. Schatz
154 VIC Color Tips .....	Charles Brannon
157 The Window .....	Charles Brannon
160 Custom Characters For The VIC .....	David Malmberg
<b>Chapter Five: Maps And Specifications.</b>	
173 How To Use The 6560 Video Interface Chip .....	Dale Gilbert
179 Browsing The VIC Chip .....	Jim Butterfield
186 VIC Memory – The Uncharted Adventure .....	David Barron / Michael Kleinert
189 Memory Map Above Page Zero .....	Jim Butterfield
<b>Chapter Six: Machine Language.</b>	
195 TINYMONI: A Simple Monitor For The VIC .....	Jim Butterfield
202 Entering TINYMONI Directly Into Your VIC-20 .....	Russell Kavanagh
211 Index .....	



# Home Energy Applications On Your Personal Computer

**Author:** David E. Pitts

**Price:** \$14.95

**On Sale:** Now

Are you caught in the crunch of spiraling utility bills? Are you shivering from the icy touch of winter drafts, or sweating while expensive air-conditioned breezes seep out of your home into the summer heat?

Maybe your home appliances are burning more energy than they should. Perhaps your home would be warmer with some more insulation and weatherproofing. Or maybe a new air conditioner or a ceiling fan could take the edge off those humid summer nights.

But can you justify the expense of such improvements? Will the insulation, for example, pay for itself after one winter, or will it take three?

Now, with the aid of your personal computer, you can find out. A new book from **COMPUTE!** Publications, *Home Energy Applications On Your Personal Computer*, can help you determine if those expensive home improvements you've been thinking of will send a chill down your spine or leave you with a warm feeling inside.

*Home Energy Applications On Your Personal Computer*, written by David E. Pitts, lets you track and analyze your utility costs so you'll know exactly where your energy dollars are going. But beyond that, it suggests various ways of cutting those costs by making common home improvements and shows you how those improvements can help pay for themselves. *Home Energy Applications* is packed with programs ready to type into your personal computer, whether you have an Apple, Atari, Commodore 64, VIC-20, PET/CBM, Radio Shack TRS-80 Color Computer, Texas Instruments TI-99/4A, or OSI.

Once you give information on your geographical area (anywhere in the mainland United States), your current expenses, and details about your house itself, this book and your personal computer will provide an in-depth, specific, objective report on what you can do to evaluate ways to reduce your energy costs. It's the perfect application for the speed and power of a personal computer.

*Home Energy Applications On Your Personal Computer* is carefully written and edited to be easy to read and use, because it comes to you from the publishers of **COMPUTE!**, the leading magazine of home, educational, and recreational computing. It is quality-produced throughout, including metal spiral binding so that its 250 pages lie flat while you're typing in programs.

Available at computer dealers and bookstores nationwide. To order directly call TOLL FREE 800-334-0868. In North Carolina call 919-275-9809. Or send check or money order to **COMPUTE! Books**, P.O. Box 5406, Greensboro, NC 27403.

Add \$2 shipping and handling. Outside the U.S. add \$5 for air mail, \$2 for surface mail. All orders prepaid, U.S. funds only.

172 *COMPUTE!'s Gazette* May 1984

## Table of Contents

Introduction .....	v
Energy Data Base .....	1
Energy Workbook .....	23
Energy Plot .....	61
Electric Usage Estimator .....	91
Home Heating And Cooling Audit .....	111
Heat Conduction .....	151
Buying A New Air Conditioner .....	169
Window Heat Loss / Gain .....	183
Window Shading Analysis .....	209
Ceiling Fan Analysis .....	227
Appendix .....	241



# PRODUCT MART

## COMPUTE!'s Gazette

TOLL FREE  
Subscription  
Order Line

**800-334-0868**

In NC 919-275-9809

## LAIR of the LIZARDMEN

In an abandoned corner of desert lie ancient, unsolved mysteries. Danger, death and forgotten treasure await you in

LAIR OF THE LIZARDMEN  
text adventure for C-64 or VIC-20  
(VIC requires 8k expansion)  
Specify disk or tape, VIC or 64

\$15.00 disk or tape. Add \$1 shipping.  
California residents add 6 1/2% tax.

WEREWOLF SOFTWARE  
109 Minna Street  
Suite 353  
San Francisco, CA 94105

## Want A Greener Green Thumb? Shake Hands with PLANTIN' PAL



If you enjoy Gardening, you'll love  
PLANTIN' PAL.

With PLANTIN' PAL you can:

- Take your custom-designed layout to your garden
- Plant the right amount of each vegetable for your needs
- Identify and beat bugs & diseases
- Create your individual planting & fertilizing schedule
- And so much more

Get friendly with Plantin' Pal today.

On disk for Commodore 64.  
Printer optional but preferred.



\$29.95



To order call (612)925-2591 or write:

**Home & Hobby Software**  
4936 Morgan Ave. S., Minneapolis, MN 55409

## VIDEO FITNESS SYSTEM



**Shape Up Your Body by using  
Your Commodore or Atari!**

VFS compels you to exercise to make your computer game "fire".

Burn away excess fat. Shape and strengthen muscles. Look and feel better. All of the benefits of physical fitness, just by playing your favorite electronic games.

VFS is the **Interactive** home exercise machine. FREE booklet #1 explains it all. Send for it.

### VIDEO FITNESS SYSTEM

Box 1221, No. Riverside, IL 60546

## "THE REUNION"®

(Brings the Commodore family together again)  
JUST RELEASED! VIC 20/C64

\*NOW - "THE REUNION" instantly interfaces two systems for instant use: (A) VIC 20, disk drive, printer and Datasette or (B) C-64, disk drive, printer and Datasette.

\*Simply select "VIC 20" or "C-64" on "THE REUNION" and state of the art design permits instant operation of selected system with modem expansion, etc.

\*Now SAVE and LOAD VIC and C-64 programs on same disk. Ends switching disks or cables.

\*Simple two minute installation. Full year warranty.

Please specify Model A - for disk drive and printer or Model B for Datasette.

Special introductory price - \$29.95.

Please add \$2.00 for shipping (Canadian: \$4.00).  
Florida resident 5% tax. Send check or money order to: HyTech

P.O. Box 466  
Bay Pines, FL 33504

\*Reg. T.M. of Commodore Bus. Mach.  
© HyTech.

## 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 (transferable 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.

The COVOX  
Speech Processor **ONLY \$89.95**  
**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)

• VIC 20 • • CBM 64 •

## SAMMY THE SPELLING SPIDER®



THE WHOLE FAMILY WILL LOVE  
THIS NEW GAME

Animated hidden word spelling game for all ages to enjoy. Spell the hidden word before SAMMY does. Create your own word lists for specific teaching aids or even different languages.

\$13.95 postage & handling included.  
Disk add \$2.50. COD add \$1.80

Send for free Catalog

"Quality Software at Integrity Prices"



INTEGRITY SOFTWARE P.O. Box 29 • Bristol, VT 05443  
18021 453-3122

Dealer Inquiries Invited



## COMMODORE 64 INTERFACING BLUE BOOK

Written by a college professor in a friendly and informative style, this book is a gold mine of practical information on how to build a variety of useful interfaces for your computer. Over 30 projects including cassette interface, printer interface, expansion bus, light pen, RS-232 interface, more user ports, thermometer, speech synthesizer and two other voice projects, A/D and D/A converters, 5 telephone interfaces including a sophisticated voice mail project, motor controllers, AC power control, logic analyzer, capacitance meter, spectrum analyzer, home security system and many more. \$16.95 postpaid. Foreign orders add \$3 for AIR MAIL shipping.

## VIC 20 INTERFACING BLUE BOOK

30 hardware interfacing projects for the VIC 20. Similar to the 64 book above. See what your VIC can do with a little interfacing. \$14.95 postpaid.

**microsignal**  
Dept. C, P.O. BOX 22  
MILLWOOD, NY 10546

## ETT 64 Electronic Typing Teacher

### The Most Complete Typing Instruction Course for Commodore 64

Cassette Version . . . \$24.95  
38 Lessons

Disk Version . . . . . \$29.95  
64 Lessons

Demo Tape of ETT64 \$2.00

Demo Disk of ETT64 \$3.00

Full purchase price of demo  
applied to purchase.

## Knight Writer Software

P. O. Box 598 Westland Mi. 48185



313-728-0946



DEALER & SCHOOL DISCOUNTS AVAILABLE  
C.O.D. Extra - U.S. Shipping Add 5%

## 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

## Educational Software For The COMMODORE

# VIC-20



Wide Variety Of Subjects  
Available For All Ages

FOR FREE BROCHURE WRITE:

Schoolmaster  
Programming Company  
P.O. Box 194, Pomona, CA 91769

## 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.

20 64	<input type="checkbox"/> Programmer's Aid <sup>1</sup>	20 64	<input type="checkbox"/> Graphic printer (1515 & 1525) <sup>1</sup>
	<input type="checkbox"/> Super Expander <sup>1</sup>		<input type="checkbox"/> HES Vic Forth <sup>2</sup>
	<input type="checkbox"/> Vic Typewriter <sup>1</sup>		<input type="checkbox"/> HES Writer <sup>1</sup>
	<input type="checkbox"/> Victim 1 <sup>1</sup>		<input type="checkbox"/> Wordpro 3 plus
	<input type="checkbox"/> Term 64 <sup>1</sup>		<input type="checkbox"/> Easy Script <sup>1</sup>
	<input type="checkbox"/> Quick Brown Fox		<input type="checkbox"/> Basic
	<input type="checkbox"/> Hesmon 64 <sup>2</sup>		<input type="checkbox"/> Paper Clip
	<input type="checkbox"/> Calc Result (advanced)		<input type="checkbox"/> Script 64
	<input type="checkbox"/> Pilot (Commodore)		<input type="checkbox"/> PractiCalc 64/plus

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) Product of Human Engineering Software. VIC-20 is a trademark of Commodore Business Machines, Inc.

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

## EPROM PROGRAMMER

FOR  
PET<sup>1</sup> • COMMODORE-64<sup>1</sup> • VIC-20<sup>1</sup>  
DELUXE INCLUDES:

- MACH. LANG. MONITOR
- mini-ASSEM/EDITOR



• FILE MANAGER \$99.50 •  
for software development & EPROM coding.  
PROGRAMS OVER 40 popular device  
types including 25xx and 27xx series  
up to 32K BYTES (incl. some E<sup>2</sup> types).  
MENU DRIVEN software runs all 3 CPU's  
connects to USER PORT, L.I.F. socket incl.

**ECONOMY-2716-64**  
read, pgm, & ver. ONLY  
basic programmer when  
editing & file storage  
ARE NOT needed L.I.F. incl. \$59.50 •



<sup>1</sup> PET, COMMODORE 64, & VIC 20 are trademarks of CBM, INC.

(215) 256-6933 **DAZCO** (215) 256-6933



Box 287 Lederach, Pa. 19450  
• +\$2. ship. & Pa. res. add 6%



## (ABS)OLUTE SOFTWARE INTRODUCES COMPUFILE THE (ABS)OLUTE DATA BASE FOR COMMODORE 64 COMPUTERS

### Some Features.....

- True Relative Disk Files
- Multiple Data Bases Allowed On A Single Disk
- Completely Menu Driven. Including Disk Directory
- Up To 20 User Defined Fields Per Record
- No Software Limit On Number Of Records Per Data Base
- Searches & Sorts On Any Field
- Adds, Deletes, Modifies Records
- Clear & Thorough Documentation

And The Best Feature: **Compufile** has an  
Introductory Price Of Only \$39.95!!

Look For **Compufile** At Your Local Commodore  
Dealer, Or Send \$39.95 To.....

(ABS)OLUTE SOFTWARE  
1780 AUSTIN HIGHWAY  
SAN ANTONIO, TEXAS 78218  
(512) 828-2679

Dealer Inquiries Invited

(ABS) The First Name In Commodore Computing

## COMMODORE SOFTWARE AFICIONADOS

You'd like free software,  
we'd like reviews. The  
Book Company seeks  
additional reviewers for  
The Book of  
Commodore Software.

For details, write and  
send a sample review to:

**The Book Company**  
11223 S. Hindry Ave.  
Los Angeles, CA 90045

## SOFTWARE



## HANDBOOK

THIS IS A COMPLETE 84 PAGE MANUAL THAT  
SHOWS HOW TO DUPLICATE PROTECTED VIC-20  
SOFTWARE. WE INCLUDE ALL SCHEMATICS, PC  
LAYOUTS AND SOFTWARE LISTINGS. OUR METHODS  
DO NOT REQUIRE EXPENSIVE MOTHERBOARDS.  
PLANS INCLUDED FOR...STANDARD RECORDER  
INTERFACE/TAPE DUBBER...CARTRIDGE COPIER  
(8K OR 16K TO TAPE OR DISK) AND MORE! WE  
ALSO HAVE CIRCUITS AS KITS!

VIC-20 BOOK.....\$9.95 US

C-64 BOOK.....AVAILABLE SOON!...\$ASE FOR INFO

THIS MANUAL DOES NOT CONDONE PIRACY

DISTRIBUTING COPIED SOFTWARE IS ILLEGAL

PSIDAC, 7326 N. ATLANTIC, PORTLAND, OR 97217

-CHECK OR MONEY ORDER ONLY-

[www.commodore.ca](http://www.commodore.ca)



## VIC-20 COMMODORE 64

### THE RECIPE BOX

Now you can easily store and recall your favorite recipes on your Commodore computer. THE RECIPE BOX is a complete menu-driven disk system that comes with these additional features:

**SEARCH BY INGREDIENT** — Only have a pound of hamburger in the freezer? Let THE RECIPE BOX show you all the recipes that you have on file that use hamburger, or any other ingredient you choose.

**SEARCH BY CATEGORY** — Code your recipes as to breakfast, lunch, dinner, snacks, etc.

**SEARCH BY CATEGORY/INGREDIENT** — Any combination of the above.

**AUTOMATIC MEASUREMENT** — THE RECIPE BOX will automatically scale up or down the amount of ingredients you need according to how many servings you want.

**SCREEN OR PRINTED OUTPUT** — Have printed copies to use in the kitchen or give to friends.

THE RECIPE BOX requires one disk drive and will run on a 5K VIC-20, Commodore 64. Please specify. Send check or money order for \$21.95 to:

**Aries Marketing Co.**  
P.O. Box 4196  
4200 Shannon Drive  
Baltimore, Md. 21205  
Md. residents add 5% sales tax

## REFUNDERS!

### Use your Commodore 64

"REFUND INFO" is a filing program designed specifically for refund and coupon offers. Record your refund offers as you mail them. The use of menus for all functions makes it easy to use, even with limited computer knowledge.

- Automatically alphabetizes
- Three search modes
- Two listing modes
- Update at any time
- Compute cash and non-cash totals received between any specified dates
- Output data to screen or printer
- Addresses envelopes

"REFUND INFO" runs on tape or disk.

Each file holds 200 records.

TAPE - \$17.95 DISK - \$19.95

"REFUND INFO II" runs only on disk.

Each file holds 700 records.

DISK - \$23.95

Send check or money order.

Kansas residents add 3% sales tax.

For more info send SASE.

## FLATLAND SOFTWARE

P.O. Box 8721 • Wichita, KS 67208

## Numeric key pad for Commodore VIC-20 and 64



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

**\$59.95**

Retail

**Deer**

**Technologies**



4321 Airwest S.E.  
Grand Rapids, MI 49508  
(616) 698-5000

VISA and Mastercard accepted

## 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

## UNICORN ADVENTURE

TEXT & GRAPHIC

Introductory

Offer ...

**\$24.95**

TAPE or  
DISK



For ...  
C-64

YOUR MOTHER, A  
BEAUTIFUL WHITE  
UNICORN — YOU  
MUST FIND WHERE  
HE HAS TAKEN HER!

— Dealer Inquiries Welcome —

## LOGGINS ENTERPRISES

2814 - 23rd Avenue West  
Bradenton, Florida 33505

## SYSTEMS TO BOOT

SOFTWARE DEVELOPMENT AND DISTRIBUTION



### PRESENTS

THE FIVE ADVENTURES THAT WILL DEMAND QUICK  
REFLEXES OF THE MIND, NOT THE HAND!

- ROCKET V-2 only \$15.95
  - DR. GEARDORF'S LASER only \$15.95
  - THE ISLES OF SSENKRAD (Part 1) only \$15.95
  - BENEATH THE ISLES OF SSENKRAD (Part 2) only \$15.95
  - SAND SCREAM only \$15.95
- (All available for 13K VIC-20 or C-64)

We Accept Mastercard/Visa, Certified Check, M.O., Allow 2 wks. for Personal Checks.

ATARI, TRS-80 Versions Coming Soon

Add \$2.00 for Disk, Add \$1.50 for Postage/Handling  
RUSH ORDER TO: SYSTEMS TO BOOT

2109 S. Fieldcrest • Wichita, Kansas 67209

Ph. 316-721-1312

\*ATARI, Commodore, TRS-80 are registered trademarks

SPECTRUM SOFTWARE  
P.O. BOX 11471  
SAN FRANCISCO, CA  
94101

## PUBLIC DOMAIN SOFTWARE

TELECOMMUNICATIONS  
UTILITIES  
GAME/MUSIC/GRAPHICS  
CPM PROGRAMS  
BUSINESS/EDUCATION

THE BEST OF PUBLIC DOMAIN  
ALL PROGRAMS TESTED AND  
WELL DOCUMENTED

### SAMPLE DISK \$8

CONTAINS A SELECTION OF PROGRAMS  
FROM ALL OUR CATEGORIES  
PLEASE SPECIFY  
C64 OR VIC20 / TAPE OR DISK

**C64 BBS PROGRAM**  
THE FIRST RELEASE IN OUR OWN LINE  
OF SOFTWARE  
LOW COST BULLETIN BOARD PROGRAM  
USERS COMMODORE 64, 1650 AUTODIAGN  
AND 1541 DRIVE. SEE OUR CATALOG  
FOR PRICE AND SPECIFICATIONS.

### FREE CATALOG

WRITE DIRECT FOR FAST RESPONSE OR  
CIRCLE 11 ON READER SERVICE CARD

## AT LAST! EASY BASIC DISK

For Commodore 64™ Users

### FEATURES

- Three Dimensional Graphics
- 44 Additional Basic Commands
- Full Function Keyboard
- Sprite & Game Programming

**ONLY \$29.95**

ADD \$2.50 SHIPPING & HANDLING

UTAH — ADD SALES TAX

• CHECK, MO, VISA or MC  
(SEND CREDIT CARD EXP. DATE)



## CLONEWARE

## CLOWENWARE

P.O. Box 587 C.G.  
Pleasant Grove, UT 84062

Dealer inquiries welcome

Send self addressed envelope for free brochure.

Commodore 64 is a trademark of Commodore Business Machine

## 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:

1. Protected Version (Protected Version can be run but not listed or modified) (Please specify Disk or Cassette) ..... \$25
  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





# ADVERTISERS INDEX

## Reader Service Number/Advertiser Page

102	Aardvark Action Software	119
103	(Abs)olute Software	174
104	Academy Software	24
	Academy Software	58
105	Advanced Ideas	13
	Aries Marketing Co.	175
106	Artificial Intelligence Research Group	175
	Assembly Technology	106
107	Avalon Hill Game Company	25
	Basic Byte, Inc.	86
108	Batteries Included	31
109	Batteries Included	57
110	Bear Technologies	175
	Big Bytes	106
	The Book Company	174
	Boston Educational Computing, Inc.	77
111	Bytes & Pieces, Inc.	81
112	Bytes & Pieces, Inc.	140
113	Cadmean Corp.	133
114	Cardco, Inc.	IBC
	Century Micro Products	101
	Cheatsheet Products	174
	The CHF Company	103
	Cloneware	175
	Commodore Computers	BC
115	CompuServe	21
	Computer Mail Order	99
116	ComputerMat	115
	Computer Place	133
	Computer Warehouse	130
117	Continental Software	23
	Covox Co.	173
	Creative Software	4
118	Datamost, Inc.	65
	Dazco	174
	Dow Jones News/Retrieval	75
119	Eastern House	81
120	Eastern House	107
	Educomp	44
	Elcomp Publishing, Inc.	63
	Electronic Arts	28,29
	Electronic Arts	43
	Epyx	51
	Flatland Software	175
121	French Silk	85
	The Furniture Byte	103
122	Futurehouse	83
	Genealogy Software	174
123	GOSUB of Slidell, Inc.	111
124	Handic Software Inc.	11

## Reader Service Number/Advertiser Page

	H & H Enterprises	175
125	Home & Hobby Software	173
	HyTech	173
126	Integrity Software	173
127	Jamestown Software	133
128	Knight Writer Software	174
129	Limbic Systems Inc.	107
	Loggins Enterprises	175
130	Lynn Computer Service	85
131	MFJ Enterprises Incorporated	101
132	Microlab, Inc.	61
133	The Micropерipheral Corporation	124
	Micro Sci Corp.	69
	Micro Sci Corp.	71
	Microsignal	174
	Micro Software International, Inc.	73
134	Micro Ware	16
135	Micro Ware	66
	Micro World Electronix, Inc.	75
136	Micro Worx	126
137	Mirage Concepts, Inc.	15
	MMG Micro Software	105
	Multi-Pac Software	114
138	Oakwood Computer Products	101
139	Orbyte Software	69
140	Orbyte Software	71
141	Panther Computer Corporation	IFC
	Parallel Systems	130
142	PC Gallery	77
	Penguin Products	62
143	Precision Software, Inc.	1
144	Professional Software Inc.	9
	Pro-Line Software	67
145	Protecto Enterprises	88,89
146	Protecto Enterprises	90,91
147	Protecto Enterprises	92,93
	Psidac	174
	PSI-WARE	124
	Quicksilver Inc.	53
148	Richvale Telecommunications	45
	Scarborough Systems, Inc.	7
	Schoolmaster Programming Company	174
149	The 64 Club	130
150	'64 Shopper	114
151	SJB Distributors Inc.	103
152	Skyles Electric Works	59
	SM Software Inc.	112
	SM Software Inc.	112
	SM Software Inc.	113
	SM Software Inc.	113

## Reader Service Number/Advertiser Page

	Softlaw Corporation	97
153	SoftPeople Inc.	37
154	The Software Buyer's Report	78
	Software Design, Inc.	41
155	Software Discounters of America	135
	Software Plus	120
156	Software Warehouse Outlet	133
157	Spectrum Software	175
	Spinnaker	2,3
158	Star Micronics Inc.	35
159	subLOGIC Corporation	39
160	Such A Deal	129
161	Synapse	55
162	Systems Management Associates	109
163	Systems Management Associates	131
	Systems To Boot	175
	Taylor-made Software	140
164	Timeworks, Inc.	27
165	Toll Software, Inc.	86
	Tussey Mountain Software	136
	Ultrabyte	124
	Video Fitness System	173
	Virginia Micro Systems	114
	Werewolf Software	173
	York 10	120

COMPUTE! Books	48,49
COMPUTE!'s First Book of VIC	171
COMPUTE!'s First Book of VIC Games	170
COMPUTE!'s GAZETTE Back Issues	117
COMPUTE!'s GAZETTE DISK Subscription	33
COMPUTE!'s GAZETTE Subscription	17
COMPUTE!'s PC & PCjr	79
Home Energy Applications	172



# You asked for it . . . CARDCO has it!

Another exclusive from Cardco!

## The LQ/1 (Letter Quality) PRINTER

In response to your demand and need, CARDCO now has available for the first time . . . a "Commodore-ready" "LETTER QUALITY PRINTER". Just plug it in and print.

Offering standard friction-feed and optional tractor-feed, the CARDCO LQ/1 PRINTER comes complete with built-in interfacing for all Commodore Personal Computers, as well as compatible input for PC, PC Jr., TRS-80 and other personal computers.

## "CARDPRINT" C/?+G PRINTER INTERFACE with Graphics

For printers that are not Commodore-ready, Cardco offers the C/?+G PRINTER INTERFACE including all cables and connectors for the following printers: all new Epson MX, RX and FX series, Star Gemini 10X and Delta 10, Prowriter, C.-Itoh 8510, NEC 8023, Okidata 82, 83, 84, 92, 93 and 94, Mannesman Tally Spirit and MT-160, Seikosha GX-100, BMC BX-80 and the Gorilla Banana. Prints the full Commodore character set; graphics, characters, reversed characters and reversed graphics characters.

## "WRITE NOW" WORD PROCESSOR SOFTWARE

An excellent time saver, CARDCO OFFERS THE C02 "Write Now" program with built-in 80 column display. You see exactly what will print. All special codes can be transmitted to printers maintaining justification. Easy full-screen editing; works with any printer.

## "MAIL NOW" MAILING LIST SOFTWARE

CARDCO D/01 quickly (in memory) sorts by zip, category, name and state; fully compatible with "WRITE NOW". Other fine features include: User-oriented; menu-driven operation; each disk supports 600 entries. Format can print single, double or triple labels across.

All Cardco Products are available at your local dealers. Write for illustrated literature.



### cardco, inc.

313 Matheson Wichita, Kansas 67214 (316) 267-6525

"The world's largest manufacturer of Commodore accessories."

Commodore™ is a registered trademark of Commodore Business Systems, Inc.

[www.commodore.ca](http://www.commodore.ca)



# Commodore 64 Magic Desk I

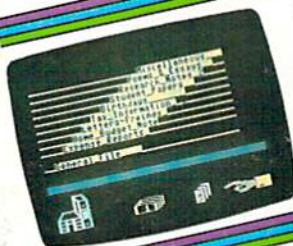
Only From Commodore—The Excitement and Simplicity of Magic Desk!



Only Commodore brings you the magic of MAGIC DESK... the next generation of "user friendly" software! Imagine using your computer to type, file and edit personal letters and papers *without learning any special commands!* All MAGIC DESK commands are PICTURES. Just move the animated hand to the picture of the feature you want to use (like the TYPEWRITER) and you're ready to go.



The MAGIC DESK Typewriter works just like a real ELECTRIC TYPEWRITER... and it's COMPUTERIZED. All the filing is *electronic*. Excellent sound effects and screen animation make typing fun, whether you're typing letters, reports or memos... and the built-in filing feature makes MAGIC DESK useful for keeping names and addresses, home inventory lists, insurance information and more.



Your COMMODORE 64, COMMODORE DISK DRIVE and MAGIC DESK are an unbeatable combination. Filing operations are automatically linked to your Commodore disk drive—but you don't have to know any commands—just "file" the pages you type in the file cabinet and your text is automatically saved on diskette. There are 3 file drawers with 10 file folders in each drawer and 10 pages in each folder.



To PRINT a page you've typed, just "point" at the picture of the printer and your pages are automatically printed on your COMMODORE PRINTER or PRINTER/PLOTTER. If you want to erase what you've typed, the WASTE-BASKET under the desk lets you "throw away" pages. There's even a DIGITAL CLOCK which helps you keep track of time while you're typing.



Not only is MAGIC DESK easy to use... it's hard to make a mistake! Just press the COMMODORE key and one of several "help menus" appears to tell you exactly what to do next. Special messages show you how the various picture commands work and help you when you make a mistake. Help messages also show you how to use the printer, filing cabinet, digital clock and wastebasket.

Another reason why Commodore is number one in the world of microcomputers—Come join us.

**commodore**  
**COMPUTERS**  
First In Quality Software