UNIQUE HARDWARE
For Your Commodore or Vic

Commodore or Vic Color Problems?
We Can Solve Them All.

You're not alone. Thousands of Commodore 64 owners have "fuzzy" color on their TVs. 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...** For Vic-20 and Commodore 64. 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

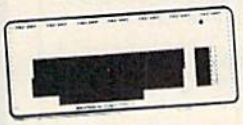
4. **The Monitor "Improver"...** If you have a Commodore 1701 monitor, this cable (3 prong) gives you a picture you won't believe. Better than the cable Commodore built ... by a lot. Try it, you won't be disappointed. (Also hooks your "Old 64" to the 1702.)
$24.95

5. **At last, the "needed" switch for all Vic-20's and Commodore 64's.** Commodore left out something that's really important — a simple reset switch. How many times have you been programming and gotten "hung-up" in your software? The only way to get back in control is to turn off the computer and lose your program and everything you had entered so far. Well, the engineers at Bytes & Pieces have solved that one too ... a reset switch. Now installing this does require you to open your computer, make two simple solder connections, and drill a small hole in the case (to mount the switch). Obviously, this connection will void your warranty, so don't proceed until your computer is out of warranty. But the day that happens, install the reset switch. It's a time saver, and it's guaranteed to return control to you every time. Of course, you're guaranteed to be satisfied.
A steal at **$9.95**

6. **Dust problems? We've got the answer!** There are a lot of cheap dust covers on the market, most of them made from static-filled plastic. But there are some of us who think a lot of our Commodores. We want to protect them and have them look nice at the same time! That's why Bytes & Pieces built the best looking dust covers on the market. They're hand-sewn from leather-like naugahyde in a brown leather-grain pattern. They're custom-built to fit your Commodore 64, and here's the best news of all. You can get matching covers for your disk drives and your cassette unit as well. You made a big investment in your Commodore; spend a few more dollars and protect it from damaging dust for life. Your satisfaction is guaranteed.

Computer dust covers: **$9.95**
Disk drive dust covers: **$8.95**
Dataset dust covers: **$7.95**

Why Blank "Cheat" Sheets?



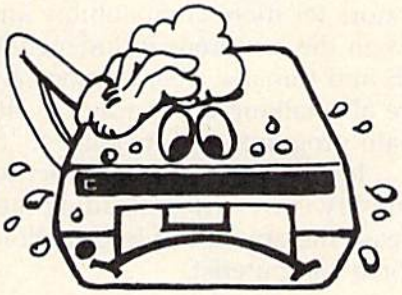
Because They're Better Blank

O.K. So now you've got the best computer in the world, and lots of complex software to run on it. One problem. Unless you work with some of these programs everyday or are a computer genius, who can keep all those commands straight? "F5" in one program means one thing, and "F5" in another program means something else.

A few companies do offer a solution ... a die cut "cheat" sheet that attaches to your keyboard with all the commands of one program printed on it. Great idea, unless you need them for 10 or 20 programs. You could purchase another disk drive for the same investment. Our solution? Simple. A pack of 12 lined cards, die cut to fit your keyboard and just waiting to be filled with those problem commands **you** forget most often. Simple? Yes, but effective. Now you can have **all** your program commands right at your finger tips on YOUR VERY OWN, custom designed "cheat" sheets. Order a couple packs today!

12 for **$15.95**

Is Your Commodore Disk Drive Hot and Bothered?



Most of them are, you know. Commodore makes a great disk drive. Only trouble is, they suffer from read and write problems frequently. And almost always, it means a trip to the shop for a head alignment. Maybe you can afford to have your drive out of commission for a while. And to pay to have your drive repaired. But we've been told that most of these problems occur because the drive has overheated, throwing the head out of alignment because of parts expansion.

The engineers of Bytes & Pieces pondered this problem, and came up with a simple solution. An inexpensive muffin fan that sits on top of the disk drive and blows cooling air through it. No more hot and bothered drives. No more heat-caused read/write problems. A simple, inexpensive solution. And best of all, the fan will work on other computer items as well, as long as they have vent holes in the top. Just set the padded fan on top and your problems with overheating are over.

And we went one step further. We built a surge protector into a second fan model. Most double-outlet surge protectors sell for more than the cost of our fan and surge protectors put together.

So order today. You won't be sorry. Satisfaction guaranteed or your money back.

Muffin fan **$54.95**
Fan with surge protector **$79.95**

Order Today!

Rush me the following:

Qty.	Item	Amount
____	Commodore 64 Interference Stopper @ $15.95	$ ____
____	Vic-20 Interference Stopper @ $15.95	$ ____
____	Color Sharpener @ $18.95	$ ____
____	NEW Color Sharpener Cable @ $24.95	$ ____
____	The Monitor Improver *$24.95	$ ____
____	The Reset Switch @ $9.95	$ ____
Dust Covers		
____	Commodore Dust Cover @ $9.95	$ ____
____	Vic-20 Dust Cover @ $9.95	$ ____
____	1541 Disk Dust Cover @ $8.95	$ ____
____	Dataset Dust Cover @ $7.95	$ ____
Cheat Sheets		
____	Sets of 12 Keyboard Cheat Sheets @ $15.95	$ ____
____	2 Packs (24 Sheets) @ $24.95	$ ____

Qty.	Item	Amount
Muffin Fans		
____	Muffin Fans @ $54.95	$ ____
____	Muffin Fans with Surge Protector @ $79.95	$ ____
	Shipping & Handling	$ 2.00
	Sub Total	$ ____
	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.
550 N. 68th Street,
Wauwatosa, WI 53213
Dealer Inquiries Invited

with *Vidtex*, you wouldn't need to pay $6 an hour to use the on-line word processors.

Integrated Bulletin Boards

The same level of integration and sophistication is just around the corner for bulletin board users.

There is a strong probability that by the end of 1984 we'll see telecommunications packages which include a fully compatible bulletin board system (BBS), terminal program, and word processor, all produced by the same company.

There is growing support among BBS operators for more compatibility and cooperation between the hundreds of independent boards in the US and Canada. Board sysops (system operators) are also talking about ways to share public domain programs and messages.

Ideally, all of these factors will combine to simplify current downloading methods and to increase the amount of information available to the home computerist.

If you have questions or ideas about subjects you'd like to see covered in this column, write to: Home Telecommunications, COMPUTE!'s GAZETTE, *P.O. Box 5406, Greensboro, NC 27403. Or you can send me electronic mail. My CompuServe ID is 75005, 1553. For Delphi, it's BOZART.*

VICreations

Dan Carmichael, Assistant Editor

In looking at a lot of home-brewed programs, I've seen many good, original ideas. However, the best programs are sometimes hindered by bad screen displays and/or poor screen and color usage.

Some programmers use only one quarter of the available screen, while others have the screen so cluttered you can't make heads from tails.

This month, let's talk about some ideas on screen and color usage. With an appealing screen and good colors, you can make your programs attractive and pleasant to use for even a first-time computer user.

Screen Usage

An organized screen adds a lot to a program. When the screen display is neatly presented, the program becomes much easier to use. If the screen display is confusing, the user is more than likely to also become confused.

Use the entire screen if possible. Don't try to place all the information into one area. For example, organize your screen displays into neatly drawn rows and columns. When working with large amounts of data (such as in a data base), scanning rows and columns then becomes easier and less tiring for the user.

If at all possible, don't place too much information on the screen at the same time. Double-space your display when possible. It's much easier on the eye. When the user looks at a readable screen, there's much less chance of error.

Colors

Colors can be very important. Besides their aesthetic value, they can help make programs appealing.

The VIC-20 has eight colors available, so use them to their best advantage. For example, in financial applications, color combinations can instantly signal to the user the status of an account. Using red to display deficits and green to display credits is already standard procedure. Beyond that, you might include yellow for recent payments, black for overdue payments, etc.

However, there are cautions. When writing programs for a wide range of users, you might assume too much. What if the program is being run on a computer with a black and white monitor? Reds and greens (as in the above example) can appear to be the same gray on a black-and-white screen.

I've seen many programs that place red print on top of a blue background. Red and blue tend to offset each other and look nice. However, on a black-and-white monitor, these colors look the same—the print can't be seen. The best way to view your program as others may see it is to turn down the color level on your monitor. This will allow you to preview the program in both color and black and white.

Cursor Controls

The cursor control characters on your VIC can be very powerful tools. Using them in PRINT statements will allow you to move anywhere on the screen without altering the screen display. This program demonstrates some of the uses of the cursor controls:

```
20 PRINT"{CLR}{BLU} NAME"TAB(18)"{RED}AGE
   "                                :rem 196
30 PRINT"{8 DOWN}{PUR} COLOR"TAB(17)"
   {GRN}YEAR"                       :rem 250
35 REM THE FOLLOWING LINE REQUIRES A SHIF
   TED ASTERISK                      :rem 36
40 PRINT"{2 DOWN}{BLU}**************************
   ***"                              :rem 246
50 GOSUB 1000                       :rem 166
55 PRINT"{BLK}WHAT'S YOUR FIRST NAME"
                                    :rem 113
60 INPUT NA$                        :rem 169
70 PRINT"{HOME}{DOWN} "NA$           :rem 15
80 GOSUB1000                        :rem 169
90 GOSUB1000                        :rem 170
100 PRINT"WHAT'S YOUR AGE"           :rem 44
110 INPUT AG                        :rem 170
120 PRINT"{HOME}{DOWN}"TAB(17)AG    :rem 160
130 GOSUB1000                       :rem 213
140 PRINT"WHAT'S YOUR FAVORITE{2 SPACES}C
    OLOR"                            :rem 66
150 INPUT CO$                       :rem 220
160 PRINT"{HOME}{10 DOWN} "CO$      :rem 219
170 GOSUB1000                       :rem 217
180 PRINT"WHAT YEAR WERE YOU":PRINT"
    {3 UP}BORN"                     :rem 238
190 INPUT YR                        :rem 213
200 PRINT"{HOME}{10 DOWN}";TAB(16)YR
                                    :rem 149
210 GOSUB1000                       :rem 212
220 PRINT"REPEAT QUIZ?":PRINT"{2 UP}(TYPE
     Y OR N)"                       :rem 182
230 INPUT AN$:IFAN$="N"THENGOSUB1000:PRIN
    T"PROGRAM ENDED":END             :rem 39
240 GOTO20                           :rem 49
999 END                             :rem 130
1000 PRINT"{HOME}{13 DOWN}"         :rem 130
1010 PRINT"{66 SPACES}"             :rem 147
```

```
1020 PRINT"{UP}{44 SPACES}"          :rem 37
1030 PRINT"{8 UP}":RETURN            :rem 55
```

The program is a simple little quiz that asks for your name, age, favorite color, and the year you were born. It splits the screen, using the top half to display the input information, and the bottom for the user prompts.

Lines 20–40 print the top half of the screen. The cursor is then dropped down to the prompt window, and the first question is displayed. After the question is answered, the cursor is moved to the home (top left corner) position on the screen. This is done with the home (cursor) character. The cursor is then moved to the proper display position (in line 70), and the user's name is printed.

After printing the information, the program goes to the subroutine at line 1000. This simply clears the user prompt area and makes it ready to display the next question. It does this by first moving the cursor to the home position. Then the cursor is moved down 13 lines, and spaces are printed in the prompt window. The spaces are printed over the last prompt, erasing it. The cursor is then moved up a few lines, the next question is displayed, and the whole cycle starts over again.

When using the cursor extensively, it's best to start each print operation at one reference point. The home position is easiest because the cursor can be moved there with one PRINT statement: PRINT "{home}".

Machine Language And Screen Usage

Machine language can be a useful tool when organizing screen displays. Using machine language to move the cursor is easy because your VIC has a built-in (cursor) *plot* routine.

Following is a routine that you can incorporate into your own programs. With two POKEs and a SYS, it will position the cursor anywhere on the screen. The program is a BASIC loader, and will POKE the machine language program into the cassette buffer.

```
10 FORA=885TO892:READB:POKEA,B:NEXT
20 DATA 24,166,251,164,252,76,240,255
```

To use the program, first POKE the column number (0–21) you wish to move the cursor to into memory location 251, e.g., POKE 251,10. Next POKE the row number (0–22) into 252, e.g., POKE 252,10. And then SYS885 to move the cursor to that position and PRINT the desired data. A sample line might look like this:

```
30 POKE 251,10:POKE 252,10:SYS 885: PRINT
"ABC"
```

By effectively organizing and coloring your screen displays, you can give your programs a professional look.

Rescuing Programs From Tape Load Errors

Alan M. Wilson

If you've discovered a clever timesaving technique or a brief but effective programming shortcut, send it to "Hints & Tips," c/o COMPUTE!'s GAZETTE. *If we use it, we'll pay you $35. Due to the volume of items submitted, we regret that we cannot always reply individually to submissions.*

Saving programs on tape is usually very reliable. But at some point, you will encounter the frustrating ?LOAD ERROR.

If it were a simple matter of syntax, you could find the line in the program and fix it. But you don't even have a program; it was lost in the netherworld between the tape and the computer. What can you do?

First Aid

The first thing you can do is try again. If this works, you can forget about the more drastic measures. If you had been working on a program, especially one with a lot of POKEs, you should first turn the computer off and then on again. Certain memory locations are used as pointers, registers, and buffers in tape loads and you may have scrambled them with an accidental POKE.

You should also remember that the program has to be translated into electrical signals which travel through the connecting cable. Magnetic fields from your television or power supply could be getting in the way. It sometimes helps to move your Datassette to a new location, away from these sources of interference.

If you are getting load errors from programs which used to be fine, the fault may lie in your cassette player. After hours of use, the tape heads can get dirty or magnetized. You can buy inexpensive cleaning/demagnetizing kits at electronics or record stores. It's a good practice to clean the heads periodically.

There is one more possibility. If the problem is not the computer, the cable, or the tape head, it may be the tape.

If the tape is defective or the program was recorded wrong, there may still be an answer to your problem.

The Pros And Cons Of Redundancy

You may not know it, but Commodore did you a favor when they designed their cassette storage system. When you SAVE a program, three sections are written to tape. The first is the tape header, with the name of the program and some other information. The second is the program. The third is the program (again).

When a program is LOADed, the header goes into the cassette buffer, the first copy of the program goes into memory, and the second copy is then checked against what is in memory. If they match, the BASIC pointers to the end of memory are set and the program is ready to run. If the two programs (which should be identical twins) don't match, you get a ?LOAD ERROR. It is more than a simple checksum, it is complete redundancy.

One disadvantage of redundancy is that it doubles the time needed to load and save. Another disadvantage is that you can't get to the program if the first copy is perfect and the second is flawed.

The one benefit of saving twice is obvious. Redundancy makes using tape very reliable.

We have the first clue to solving our problem: The program is in memory (though it didn't look exactly like its twin). If you PEEK the first few locations of BASIC memory, you will see the line link, line number, and tokenized program.

Now we have to reset the end-of-program pointers.

The first five bytes of the tape header contain some important information. The first indicates what type of tape file it is (program or data). The starting address is found in bytes two and three, the ending address in four and five. Since the header has been put into the cassette buffer, which begins at 828, we have our second clue. The end-of-program pointer is in locations 831–832.

If you have run into a ?LOAD ERROR, try entering this line (in immediate mode, without a line number):

```
FORX=45TO49STEP2: POKEX,PEEK(831):
  POKEX+1,PEEK(832): NEXT
```

The pointers are reset and the program has been rescued (maybe).

A Few Warnings

If you get a ?LOAD ERROR, do not try to LIST the program. The computer will put two zeros where it thinks the program ends. You'll lose your first line link.

You can do the POKEs listed above before you try LOADing again, but the method does not always work. It will rescue the first copy of the tape program. If the first copy is flawed, you'll get a flawed program. After the POKEs, you can LIST the program. If it looks OK, you should SAVE it immediately (and use a different tape, in case there's a flaw in the one in the Datassette).

This method is pretty reliable, and can save a program you thought was lost.

Let Your CBM-64 "SPEAK"

COMvoice IS AS EASY AS 1-2-3

1) PLUG COMvoice INTO YOUR CBM-64
2) TURN YOUR COMPUTER ON
3) TYPE SPEAK "HELLO, HOW ARE YOU"

SPEAK
SPEAK
SPEAK

AS EASY TO USE AS A PRINT STATEMENT

SPECIAL $99.95
W/EXTERNAL SPEAKER $139.95

DEALER INQUIRIES INVITED

ALSO ASK ABOUT OUR
HOME SECURITY AND ENERGY MANAGEMENT PRODUCTS

VIController
Wireless remote control system for the VIC-20 and CBM-64. Use with BSR and Leviton remote receiver modules. $69.95

COMsense
Input device for the VIC-20 and CBM-64. Provides 4 open/close and 2 analog inputs. $49.95

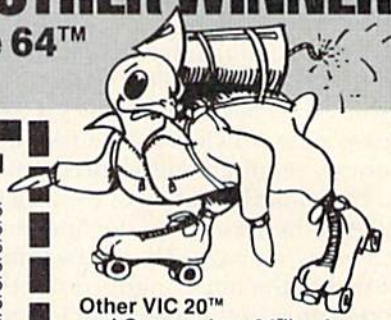
COMclock/AUTOboot
Clock/calendar cartridge for CBM-64 with battery backup and auto-start software in ROM. $69.95

GENESIS COMPUTER CORPORATION
P.O. Box 1143 Bethlehem, PA 18018 (215) 861-0850

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

TOTL SOFTWARE ROLLS OUT ANOTHER WINNER!

DataBase Management for the Commodore 64™
TOTL.INFOMASTER 3.6—only $50 on disk



Money-Saving Bonus Paks of 64 Software

(BP-1)—(disk)
totl.text/ totl.speller/totl.label
reg. price $103 NOW **$79**

(BP-2)—(disk)
totl.business/ totl time manager/ totl.infomaster/totl.text
reg. price $228 NOW **$159**

(BP-3)—(disk)
totl.infomaster/ totl.text/totl.speller
reg. price $129 NOW **$99**

(BP-4)—(disk)
totl.text/ totl.speller/ research assistant
reg. price $118 NOW **$89**

(BP-5)—(tape)
totl.text/totl.label
reg. price $60 NOW **$49**

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

INFORMATION AND ORDER COUPON

	TAPE	DISK
TOTL.TEXT 2.0 (VIC + 8K)	☐ 24.95	☐ 28.95
TOTL.TEXT 2.5 (VIC + 16K)	☐ 34.95	☐ 38.95
TOTL.LABEL 2.1 (VIC + 16K)	☐ 19.95	☐ 23.95
TOTL TIME MGR. 2.1 (VIC + 8K)	☐ 29.95	☐ 33.95
RESEARCH ASST. 2.0 (VIC + 8K)	☐ 29.95	☐ 33.95
TOTL.BUSINESS 3.0 (VIC + 24K)		☐ 84.95
TOTL.TEXT 2.6 (C-64)	☐ 39.95	☐ 43.95
TOTL.SPELLER 3.6 (-C64)		☐ 34.95
TOTL.LABEL 2.6 (C-64)	☐ 19.95	☐ 23.95
TOTL TIME MGR. 2.6 (C-64)	☐ 34.95	☐ 38.95
RESEARCH ASST. 2.0 (C-64)	☐ 34.95	☐ 38.95
TOTL.INFOMASTER 3.6 (C-64)		☐ 49.95
TOTL.BUSINESS 3.6 (C-64)		☐ 94.95
BONUS PAK #_____		_____

Check, Money Order or C.O.D.* also accepted. *C.O.D. orders $2.00 additional (CA residents add 6½% sales tax)

Total _____
C.O.D. Charges/Sales Tax _____
Shipping & Handling $3.00
Amount Enclosed _____

FOR ORDERING ONLY—CALL OUR TOLL FREE NUMBERS
Continental U.S. 1-800-351-1555, California 1-800-351-1551
Hawaii and Alaska 415-943-7877

☐ SEND MORE INFORMATION (no charge for catalog)

Name _____
Street _____
City _____ State _____ Zip _____
Phone ( ) _____ ☐ MC ☐ VISA
Card # _____ Exp. Date _____

TOTL SOFTWARE, INC.
quality you can afford
1555 Third Avenue
Walnut Creek, CA 94596
415/943-7877

Other VIC 20™ and Commodore 64™ software available from your dealer or directly from TOTL Software:
Word Processing (totl.text)
Spelling Checker (totl.speller)
Mailing List & Label (totl.label)
Business Accounting (totl.business)
Time Management (totl time manager)
Keyword Cross Reference (research assistant)

TOTL Offers You...
low prices and high quality • 30 day money-back guarantee on direct purchases • an interactive family of software • product registration • customer support • free informative newsletter • regular upgrades at reduced cost • availability in many stores • 800 numbers for ordering convenience • prompt shipment of direct orders • savings coupons with each order • money-saving bonus paks • two years and 45,000 products strong

COMPUTE!'s Gazette for Commodore

AUTHOR GUIDE

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

NEWS & PRODUCTS

Data Base For 64

Superbase 64 is a data base management and information retrieval system for the Commodore 64 computer, produced by Precision Software, Inc.

The package offers an unlimited number of data bases, with up to 15 files per data base. The number of records per file is restricted only by disk drive capacity. Each record can hold up to 1,108 characters with a maximum of 127 fields.

The system includes search and sort capabilities, and customized applications can be created within the *Superbase 64* environment. The package runs with a 1541 disk drive, or any larger Commodore drive, including a hard disk.

Superbase 64 is available for $99.

Precision Software, Inc.
Suite 1100
820 Second Avenue
New York, NY 10017
(212) 490-1825

VIC Gaming Aid

Reilly Associates has announced the release of *Fantasy Character Generator*, a programmed gaming aid for the VIC-20 computer.

The package is designed to assist the game moderator in fantasy role-playing games by generating any number of characters for a campaign. Among the features are 9 character classes, 8 character races, 13 primary statistics, personal characteristics, listing of possessions, and a number of other character statistics.

Fantasy Character Generator requires an 8K or 16K memory expander, and is available for $8.95 (add $1 for shipping and handling).

Reilly Associates
P.O. Box 17144
Rochester, NY 14617

64 Accounting Package

Clockwork Computers, Inc., has introduced the *CCI Bottom Liner*, a personal and small business accounting system for the 64 which requires a 1541 disk drive and printer.

A detail ledger file includes automatic numbering for a complete audit trail. There is also an annotation area for transaction explanations. The ledger file is linked to the accounts, client, and projects files.

The accounts file may contain up to 700 user-defined accounts in six categories: budget, income, expense, asset, liability, and equity. The client file allows the user to maintain the names,

addresses, contact persons, and phone numbers for up to 500 individuals or companies.

The project file permits the definition of up to 500 projects. Profit and loss reporting is possible as well. The program also allows comparison between budget and expenses for home improvement, childrens' education, or other like projects.

The *CCI Bottom Liner* is available on disk, with a 50-page users' manual, for $74.95.

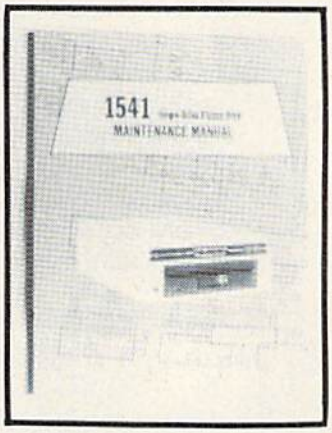
Clockwork Computers, Inc.
4612 Holly Ridge Road
Rockville, MD 20853
(301) 924-5509

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.

GOSUB

How to do your own maintenance, troubleshooting, schematics, theory of operation, cleaning hints, conversion from one power source to another and calibration. These topics and many more will make this manual a valued addition to your reference shelf. Whether you are an amateur electronics technician or a seasoned professional, you will be able to realize the full potential of your VIC-1541 by using this manual. Step-by-step instructions will lead you through the proper methods to get your VIC-1541 up and going in a hurry. The manual is 170 pages long, has two foldouts and over 100 illustrations, including:



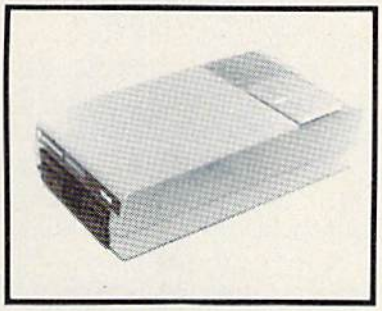
Block Diagrams
Schematics
Waveforms
Isometric (Pictorial) views
Test point locators

With all these illustrations and the detailed theory for each circuit involved, along with step-by-step procedures to follow, the manual is a great time and money saver.

CONTENTS OF MANUAL

Front Matter
Section 1 Introduction
Section 2 Theory of Operation
Section 3 Initial Configuration
Section 4 Performance Test
Section 5 Calibration
Section 6 Disassembly/Reassembly
Section 7 Preventive Maintenance
Section 8 Troubleshooting
Section 9 Schematics and Parts Layout
Appendices

Suggested list price: $39.95



HEAT DISSIPATING KIT
For VIC-1541 and 1541 Disk Drives

FEATURES:
Reduces internal temperatures to safe operating levels.
Does not promote dust migration.
No added noise.
Easy to install.
Increases life expectancy and reliability of disk drive.
Increases operating time and life of disk drive.
Installs on both VIC-1541 and 1541 Disk Drives.

The heat Dissipating kit cools the internal components of the Disk Drive by transferring internal heat to an external heat sink, where the heat is then dissipated into the surrounding air. The kit will lower operating temperatures of the IC's by as much as 20 degrees C (36 degrees F), and thus allow all the IC's to operate within their absolute maximum temperature ratings.

Suggested list price: $24.95

GRIDIRON STRATEGY '64

AND YOU THOUGHT FOOTBALL SEASON WAS OVER GRIDIRON STRATEGY '64 and the Commodore 64 now give you a year-round seat on the 50-yard line. GRIDIRON STRATEGY '64 is a highly realistic simulation of football instincts NOT "Joystick Reflexes". Most football games let you control a few players on a scrolling field. NOT GRIDIRON. In GRIDIRON, you coach the entire team and the colorful field and the stadium styled scoreboard are completely visible at all times. Also, with the use of TEAM DATA DISK '84, the teams you control are the actual pro teams, based on their performances in the '84-'85 season. Disk can be updated every year, so you can constantly keep up with the rise and fall of each team. Finally, compare these features with any other football game on the market, for any other computer:

- Real time game and 30-second play clocks?
- Colorful Graphics, and Sprite animation?
- Realistic sounds of a packed stadium?
- Optional printout copy of plays and statistics?
- Individualzed teams, based on actual performances?
- 96 possible play combinations, infinite results?
- Does not require and charts or dice for results?
- In-depth playbook and strategy sections?

GRIDIRON STRATEGY '64 offers all of these qualities.

ORDER NOW!!!

Suggested retail price:
GRIDIRON STRATEGY '64 - $27.95
TEAM DATA DISK '84 - $14.95

FOR COMMODORE 64 OWNERS —

The Aventure Situation You've Waited For !

WIZARDS, WARLOCKS AND WARRIORS

Outfit a party of up to six adventurers, hand chosen from the characters guild, descend into the depths of a true 3-D dungeon, matching wits with dozens of orcs, wraiths, and other adversaries you've learned to hate. The only difference ... no more dice charts, or pleading for mercy with a ruthless dungeon master!

The first scenario is "Quest of the Dark Orb."; use it to learn, experiment, and increase the stregnth of your characters. 100% machine language programming, Hi-Res graphics, character print out sheets & a book on the nature of the adventure are included.

Suggested list price: $39.95

ORDER FROM:

GOSUB of Slidell, Inc.
P. O. Box 1781
Slidell, LA 70459
(504) 641-8307
MasterCard and VISA
Shipping & Handling $2.00
C. O. D. add $2.00

Dealer and Distributor inquires welcome

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, [k], you should hold down the *Commodore key* while pressing the key inside the special brackets. (The Commodore key is the key in the lower left corner of the keyboard.) Again, if the key is preceded by a number, you should press the key as many times as necessary.

Rarely, you'll see a solitary letter of the alphabet enclosed in braces. These characters can be entered on the Commodore 64 by holding down the CTRL key while typing the letter in the braces. For example, {A} would indicate that you should press CTRL-A. You should never have to enter such a character on the VIC-20, but if you do, you would have to leave the quote mode (press RETURN and cursor back up to the position where the control character should go), press CTRL-9 (RVS ON), the letter in braces, and then CTRL-0 (RVS OFF).

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

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

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

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

When You Read:	Press:	See:
{CLR}	SHIFT + CLR/HOME	
{HOME}	CLR/HOME	
{UP}	SHIFT + CRSR ↑	
{DOWN}	CRSR ↓	
{LEFT}	SHIFT + CRSR →	
{RIGHT}	CRSR →	
{RVS}	CTRL + 9	
{OFF}	CTRL + 0	
{BLK}	CTRL + 1	
{WHT}	CTRL + 2	
{RED}	CTRL + 3	
{CYN}	CTRL + 4	
{PUR}	CTRL + 5	
{GRN}	CTRL + 6	
{BLU}	CTRL + 7	
{YEL}	CTRL + 8	
{1}	C= + 1	
{2}	C= + 2	
{3}	C= + 3	
{4}	C= + 4	
{5}	C= + 5	
{6}	C= + 6	
{7}	C= + 7	
{8}	C= + 8	
{F1}	f1	
{F2}	SHIFT + f1	
{F3}	f3	
{F4}	SHIFT + f3	
{F5}	f5	
{F6}	SHIFT + f5	
{F7}	f7	
{F8}	SHIFT + f7	

MLX Machine Language Entry Program

For Commodore 64 And VIC-20

Charles Brannon, Program Editor

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

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

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

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

Using MLX

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

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

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

To simplify your typing, MLX redefines part of the keyboard as a numeric keypad (lines 581–584):

```
    U I O            7 8 9
H   J K L   become   0 4 5 6
    M , .            1 2 3
```

MLX Commands

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

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

MLX recognizes these commands:

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

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

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

What if you forgot where you stopped typing? Use the Display command to scan memory from the beginning to the end of the program. When you reach the end of your typing, the lines will contain a random pattern of numbers. When you see the end of your typing, press any key to stop the listing. Use the New Address command to continue typing from the proper location.

See program listing on page 134.

The Automatic Proofreader

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

Preparing The Proofreader

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

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

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

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

Using The Proofreader

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

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

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

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

Special Tape SAVE Instructions

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

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

Replace Original Proofreader

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

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

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

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

Automatic Proofreader For VIC And 64

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

Bug-Swatter:
Modifications And Corrections

• The VIC version of "Cassette Beeper" (May) is missing a comma in line 75. The first two numbers in this DATA statement (169, −1141) should be changed to 169, −1, 141. Users of the 64 version have reported that Cassette Beeper works as listed when a program is loaded normally, but not when SHIFT-RUN/STOP (combination LOAD and RUN) is pressed.

• Some readers who own a VIC and a Datassette have reported that they cannot print documents created by "SpeedScript" (January). This is not a bug in SpeedScript, but rather a problem with the cassette drive grabbing (and holding onto) the serial bus. It is an inherent hardware bug. Commodore has suggested that after loading a program, VIC users enter SYS64490 before running it. This frees up the serial bus. The problem will continue to occur each time a tape save or load is executed.

• The next-page command [n] in SpeedScript does not work. It leads to an endless loop of form feeds. Reader Robin Franzel has disassembled and flowcharted the 64 version and has discovered a possible fix. After loading, but before running the 64 version, POKE 5755,133 seems to fix the next-page function. A word of caution: Theoretically, this POKE may affect some other embedded commands, causing the cursor to skip over the next character after the command.

In testing, however, everything worked fine. Readers who received SpeedScript as a bonus in the May GAZETTE DISK should not attempt this POKE; the next-page command works in this version.

• In addition, several readers have reported that when first trying to print a SpeedScript document, a line of seemingly random characters appears on the paper, followed by the regular text. This problem usually happens only the first time something is printed; subsequent printings are flawless.

In testing, we were unable to consistently duplicate the random characters. In hundreds of

tests, it happened only a couple of times. The problem seems to be that the printer buffer (a small amount of RAM located either in the interface or the printer) contains some unwanted characters. Readers who have problems with random characters should try turning the printer off and then on or clearing the printer buffer with this line:

```
OPEN4,4:CMD4:PRINT:PRINT:CLOSE4
```

before running SpeedScript. Another method is to clear SpeedScript memory with the Erase All Text command (SHIFT-CLR/HOME) followed by a print (CTRL-P). The printer will execute a form feed and the buffer should be cleared. If the problem persists, another solution would be to leave some blank spaces at the top of the text, followed by a next-page command. The random characters will then appear on the first page, and succeeding pages will be printed normally.

● Some readers were uncertain about how to (and why) use the File Converter program in "SpeedScript Revisited" (May). When SpeedScript files are saved, they are stored as program files using screen codes (POKE numbers) rather than ASCII codes. Some word processors and most terminal programs use sequential files in ASCII format. Readers who use SpeedScript only for word processing will never need to convert their files. But if you want to upload and download files (using a modem) or if you want to use SpeedScript files with other word processing or spelling checker programs, File Converter allows you to switch back and forth between formats, extending the usefulness of SpeedScript. Converted files may require slight editing, depending on which control codes are used by the word processing or terminal program.

MLX

See article on page 131.

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 REM LINES CHANGED FROM MLX VERSION 2.0
   0 ARE 750,765,770 AND 860       :rem 50
100 PRINT"{CLR}{6}";CHR$(142);CHR$(8);:
    POKE53281,1:POKE53280,1        :rem 67
101 POKE 788,52:REM DISABLE RUN/STOP
                                   :rem 119
110 PRINT"{RVS}{39 SPACES}";       :rem 176
120 PRINT"{RVS}{14 SPACES}{RIGHT}{OFF}
    {*}£{RVS}{RIGHT} {RIGHT}{2 SPACES}
    {*}┬{OFF}{*}£{RVS}£{RVS}
    {14 SPACES}";                  :rem 250
130 PRINT"{RVS}{14 SPACES}{RIGHT} {G}
    {RIGHT} {2 RIGHT} {OFF}£{RVS}£{*}
    {OFF}{*}{RVS}{14 SPACES}";     :rem 35
140 PRINT"{RVS}{41 SPACES}"        :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK} MACHINE LANG
    UAGE EDITOR VERSION 2.01{5 DOWN}"
                                   :rem 237
210 PRINT"{5}{2 UP}STARTING ADDRESS?
    {8 SPACES}{9 LEFT}";           :rem 143
215 INPUTS:F=1-F:C$=CHR$(31+119*F)
                                   :rem 166
220 IFS<256OR(S>40960ANDS<49152)ORS>53247
    THENGOSUB3000:GOTO210          :rem 235
225 PRINT:PRINT:PRINT              :rem 180
230 PRINT"{5}{2 UP}ENDING ADDRESS?
    {8 SPACES}{9 LEFT}";:INPUTE:F=1-F:C$=
    CHR$(31+119*F)                 :rem 20
240 IFE<256OR(E>40960ANDE<49152)ORE>53247
    THENGOSUB3000:GOTO230          :rem 183
250 IFE<STHENPRINTC$;"{RVS}ENDING < START
    {2 SPACES}":GOSUB1000:GOTO 230
                                   :rem 176
260 PRINT:PRINT:PRINT              :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S:POKEV+21,0
                                   :rem 225
310 A=1:PRINTRIGHT$("0000"+MID$(STR$(AD),
    2),5);":";                     :rem 33
315 FORJ=ATO6                      :rem 33
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                   :rem 228
390 IFN=-211THEN 710               :rem 62
400 IFN=-204THEN 790               :rem 64
410 IFN=-206THENPRINT:INPUT"{DOWN}ENTER N
    EW ADDRESS";ZZ                 :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}OUT OF RANGE":GOSUB1000:GOTO410
                                   :rem 225
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                   :rem 238
420 IF N<>-196 THEN 480            :rem 133
430 PRINT:INPUT"DISPLAY:FROM";F:PRINT,"TO
    ";:INPUTT                      :rem 234
```

```
440 IFF<SORF>EORT<SORT>ETHENPRINT"AT LEAS
    T";S;"{LEFT}, NOT MORE THAN";E:GOTO43
    0                                  :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":";         :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHT$("00
    "+MID$(STR$(N),2),3);",";          :rem 66
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                                       :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
    T:GOTO310                          :rem 50
480 IFN<0 THEN PRINT:GOTO310           :rem 168
490 A(J)=N:NEXTJ                       :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT        :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(146
    );                                 :rem 94
511 IFN=-1THENA=6:GOTO315              :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHEN530
                                       :rem 122
520 PRINT:PRINT"LINE ENTERED WRONG : RE-E
    NTER":PRINT:GOSUB1000:GOTO310:rem 176
530 GOSUB2000                          :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54
    272,0:POKE54273,0                  :rem 227
550 AD=AD+6:IF AD<E THEN 310           :rem 212
560 GOTO 710                           :rem 108
570 N=0:Z=0                            :rem 88
580 PRINT"{££}";                       :rem 81
581 GETA$:IFA$=""THEN581               :rem 95
582 AV=-(A$="M")-2*(A$=",")-3*(A$=".")-4*
    (A$="J")-5*(A$="K")-6*(A$="L"):rem 41
583 AV=AV-7*(A$="U")-8*(A$="I")-9*(A$="O"
    ):IFA$="H"THENA$="0"               :rem 134
584 IFAV>0THENA$=CHR$(48+AV)           :rem 134
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN670                      :rem 229
590 IFA>128THENN=-A:RETURN             :rem 137
600 IFA<>20 THEN 630                   :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
    {OFF}{LEFT} {LEFT}";:GOTO690       :rem 62
620 GOTO570                            :rem 109
630 IFA<48ORA>57THEN580                :rem 105
640 PRINTA$;:N=N*10+A-48               :rem 106
650 IFN>255 THEN A=20:GOSUB1000:GOTO600
                                       :rem 229
660 Z=Z+1:IFZ<3THEN580                 :rem 71
670 IFZ=0THENGOSUB1000:GOTO570         :rem 114
680 PRINT",";:RETURN                   :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                       :rem 149
691 FORI=1TO3:T=PEEK(S%-I)             :rem 67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                       :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1);:RETURN
                                       :rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***{3 DOWN}"
                                       :rem 236
715 PRINT"{2 DOWN}(PRESS {RVS}RETURN{OFF}
     ALONE TO CANCEL SAVE){DOWN}":rem 106
720 F$="":INPUT"{DOWN} FILENAME";F$:IFF$=
    ""THENPRINT:PRINT:GOTO310          :rem 71
730 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
     {RVS}D{OFF}ISK: (T/D)"            :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"THEN740
                                       :rem 36
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$:
    OPEN15,8,15,"S"+F$:CLOSE15         :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                   :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                       :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                                 :rem 69
765 K=S:POKE254,K/256:POKE253,K-PEEK(254)
    *256:POKE780,253                   :rem 17
766 K=E+1:POKE782,K/256:POKE781,K-PEEK(78
    2)*256:SYS65496                    :rem 235
770 IF(PEEK(783)AND1)OR(191ANDST)THEN780
                                       :rem 111
775 PRINT"{DOWN}DONE.{DOWN}":GOTO310
                                       :rem 113
780 PRINT"{DOWN}ERROR ON SAVE.{2 SPACES}T
    RY AGAIN.":IFDV=1THEN720           :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO720               :rem 103
790 PRINT"{CLR}{RVS}*** LOAD ***{2 DOWN}"
                                       :rem 212
795 PRINT"{2 DOWN}(PRESS {RVS}RETURN{OFF}
     ALONE TO CANCEL LOAD)"            :rem 82
800 F$="":INPUT"{2 DOWN} FILENAME";F$:IFF
    $=""THENPRINT:GOTO310              :rem 144
810 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
     {RVS}D{OFF}ISK: (T/D)"            :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"THEN820
                                       :rem 34
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                       :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                   :rem 2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                       :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                                 :rem 70
850 POKE780,0:SYS65493                 :rem 11
860 IF(PEEK(783)AND1)OR(191ANDST)THEN870
                                       :rem 111
865 PRINT"{DOWN}DONE.":GOTO310         :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.{2 SPACES}T
    RY AGAIN.{DOWN}":IFDV=1THEN800
                                       :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO800               :rem 102
1000 REM BUZZER                        :rem 135
1001 POKE54296,15:POKE54277,45:POKE54278,
     165                               :rem 207
1002 POKE54276,33:POKE 54273,6:POKE54272,
     5                                 :rem 42
1003 FORT=1TO200:NEXT:POKE54276,32:POKE54
     273,0:POKE54272,0:RETURN          :rem 202
2000 REM BELL SOUND                    :rem 78
2001 POKE54296,15:POKE54277,0:POKE54278,2
     47                                :rem 152
2002 POKE 54276,17:POKE54273,40:POKE54272
     ,0                                :rem 86
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN
                                       :rem 57
3000 PRINTC$;"{RVS}NOT ZERO PAGE OR ROM":
     GOTO1000                          :rem 89
```

Power BASIC

See article on page 110.

Program 1: Color Chart—VIC Version

```
10 FOR ADRES=828TO 874:READ DATTA:POKE AD
   RES,DATTA:NEXT ADRES                :rem 250
20 PRINT CHR$(147):A=PEEK(648)*256:FOR I=
   A TO A+512:POKE I,160:NEXT I        :rem 58
```

```
30 PRINT:FOR I=0 TO 15:PRINT:PRINT TAB(7)
   ;:FOR J=0 TO 7                    :rem 170
40 POKE 646,J:PRINTCHR$(J+48);:NEXT J,I:P
   RINT:PRINT                        :rem 164
50 POKE 646,1:PRINTCHR$(18);"THIS CHART S
   HOWS ALL{2 SPACES}";              :rem 228
60 PRINT"COMBINATIONS OF LETTER";    :rem 93
70 PRINT"AND BACKGROUND COLORS";     :rem 248
80 SYS828                            :rem 9
828 DATA 169,41,133,251,169,9        :rem 165
834 DATA 141,15,144,162,15,120       :rem 188
840 DATA 173,4,144,197,251,208       :rem 205
846 DATA 249,173,15,144,24,105       :rem 205
852 DATA 16,234,234,234,234,234      :rem 249
858 DATA 234,234,141,15,144,165      :rem 254
864 DATA 251,24,105,4,133,251        :rem 143
870 DATA 202,16,223,48,209           :rem 2
```

Program 2: Color Chart—64 Version

```
40 FOR I=49152 TO 49188: READ A: POKE I,A
   : NEXT:POKE 53280,11              :rem 175
50 PRINT CHR$(147):FOR I=1024 TO I+1000:
   {SPACE}POKE I,160: POKE I+54272,11:NEX
   TI                                :rem 204
60 FOR I=0 TO 15: FOR J=0 TO 15      :rem 237
70 P=1196+(40*I)+J: POKE P,J+1: POKE P+54
   272,J: NEXT J,I                   :rem 174
80 PRINT TAB(15)CHR$(5)"COLOR CHART":FOR
   {SPACE}I=1 TO 19:PRINT:NEXT       :rem 100
85 PRINT"THIS CHART SHOWS ALL COMBINATION
   S OF{3 SPACES}"                   :rem 112
86 PRINT "FOREGROUND AND BACKGROUND COLOR
   S.{6 SPACES}"                     :rem 237
87 PRINT "FOREGROUND INCREASES FROM LEFT
   {SPACE}TO RIGHT"                  :rem 88
88 PRINT "BACKGROUND INCREASES FROM TOP T
   O BOTTOM";                        :rem 152
90 SYS 12*4096                       :rem 200
100 DATA 169,90,133,251,169,0,141,33,208,
    162,15,120                       :rem 191
105 DATA 173,17,208,48,251,173,18,208
                                     :rem 35
110 DATA 197,251,208,249,238,33,208,24,10
    5,8,133,251,202,16,233,48,219:rem 121
```

Beekeeper

See article on page 42.

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: Beekeeper For VIC-20

```
10 POKE51,0:POKE52,28:POKE56,28:POKE55,0:
   CLR:DIMSP(8)                      :rem 172
20 PRINT"{CLR}":POKE36869,255:POKE36878,1
   5                                 :rem 18
30 FORI=7168TO7679:POKEI,PEEK(I+25600):NE
   XT:V=1:SC=0                       :rem 159
40 GOSUB500:GOSUB800:GOSUB600        :rem 27
45 GOSUB700                          :rem 128
50 S=PEEK(S2):SW=(SAND16)/16:F=(SAND32)/3
   2                                 :rem 178
60 POKES1,127:S=PEEK(S3):SR=(SAND128)/128
   :POKES1,255                       :rem 82
70 IFF=1THEN130                      :rem 109
80 I=1:A=SH-32:J=P1:POKE36877,200:rem 124
90 J=J+SP(A):IFJ<SAORJ>8185THEN120:rem 62
100 IFPEEK(J)<>32THENPOKEBN,32:GOSUB310:G
    OTO120                           :rem 146
110 POKEBN,32:POKEJ,42:POKEBBN+CM,3:BN=J:
    I=I+1:IFI<8THEN90                :rem 80
120 POKEBN,32:POKE36877,0            :rem 183
130 IFSW=1THENIFSR=1THEN200          :rem 205
140 CC=1:IFSW=0THENCC=-1             :rem 185
150 POKE36874,0:IFP1=P2THENJ=P1:GOSUB310
                                     :rem 187
160 IFSH=40THENIFCC=1THENCC=-7       :rem 106
170 IFSH=33THENIFCC=-1THENCC=7       :rem 109
180 SH=SH+CC:POKEP1,SH:POKE36874,150
                                     :rem 165
200 A=SH-32:MN=P1:P1=P1+SP(A):IFP1<SAORP1
    >SETHENP1=MN                     :rem 67
210 IFPEEK(P1)<>32THENJ=P1:POKEMN,32:GOSU
    B300                             :rem 17
220 POKEMN,32:POKEP1,SH:IFW=1THEN250
                                     :rem 244
230 P2=INT(RND(1)*21)+8054:MC=INT(RND(1)*
    6)+1:SX=41:SY=32:BC=3:W=1        :rem 33
240 IFMC=2THENSX=44:SY=43:BC=5       :rem 94
250 A=SGN(P2-P1):M0=P2:P2=P2-A*H:IFABS(P2
    -P1)>12THENP2=P2-A*21            :rem 123
260 IFP2<SAORP2>SETHENP2=M0          :rem 221
270 IFA=0THENJ=P2:GOSUB310:GOTO50 :rem 52
280 POKECM+P2,MC:POKEM0,SY:POKEP2,SX:POKE
    CM+M0,BC:GOTO50                  :rem 108

300 A=PEEK(J):IFA=43ORA<41THENP1=MN:GOTO4
    80                               :rem 153
310 FORI=1TO5:POKEJ,42:POKEJ+CM,2:POKE368
    77,150:POKE36874,200:POKEJ,32:NEXT
                                     :rem 69
320 POKEJ+CM,3:N=SQ:IFJ=P2THENW=0:SC=SC+1
    50                               :rem 186
330 IFP1=P2ORA=41THENSQ=SQ-1:P1=8043:SH=3
    9                                :rem 64
340 IFJ>8119THENSC=SC+50:AQ=AQ-1 :rem 162
350 SC=SC+50:IFSC>HITHENHI=SC        :rem 82
360 IFSC>99999THENSC=0               :rem 75
370 PRINT"{HOME}{YEL}SCORE:"SC;TAB(13)"SH
    IPS:"SQ:POKEP1+CM,3:POKEP1,SH:POKE368
    77,0                             :rem 97
380 IFSQ=NTHEN410                    :rem 29
390 FORI=1TO5:FORS=180TO235STEP2:POKE3687
    6,S:FORA=1TO10:NEXT:NEXT         :rem 195
400 POKE36876,0:FORJ=1TO100:NEXT:NEXT
                                     :rem 44
410 IFAQ=0THENV=V+1:PRINT"{CLR}{RED}
    {11 DOWN}{8 RIGHT}SWARM"V:FORI=1TO400
    0:NEXT:GOTO45                    :rem 104
420 IFSQ<>0THEN480                   :rem 62
425 POKE36877,0:POKE36874,0:POKE36876,0
                                     :rem 217
430 PRINT"{HOME}{RED}{7 DOWN}{5 RIGHT}* G
    AME OVER *":PRINT"{4 DOWN}{2 RIGHT}TR
    Y AGAIN? [Y OR N]"               :rem 112
440 GETA$:IFA$=""THEN440             :rem 83
450 IFA$="Y"THENRESTORE:GOTO20       :rem 143
460 IFA$<>"N"THEN440                 :rem 97
470 PRINT"{CLR}":FORI=36874TO36878:POKEI,
    0:NEXT:POKE52,30:POKE56,30:POKE36869,
    240:END                          :rem 250
```

```
480 RETURN                              :rem 124
500 POKE36879,125:PRINT"{RED}{7 DOWN}
    {5 RIGHT}*{BLU}BEEKEEPER{RED}*"
                                        :rem 223
510 PRINT"{GRN}{7 DOWN}{5 RIGHT}HI SCORE=
    {BLU}"HI                            :rem 135
520 FORI=1TO8000:NEXT:RETURN            :rem 51
600 POKE36879,31:PRINT"{CLR}{RED}{DOWN}
    {RIGHT}USE JOYSTICK TO PLAY":PRINT"
    {BLU}{2 DOWN}{RIGHT}BEE"TAB(11)")"SPC
    (5)"100"                            :rem 241
610 PRINT"{GRN}{DOWN}{RIGHT}CLOVER"TAB(11
    )"+"SPC(6)"50":PRINT"{RED}{DOWN}
    {RIGHT}CRAB"TAB(11)","SPC(5)"200"
                                        :rem 55
620 PRINT"{YEL}{DOWN}{RIGHT}HIVE"TAB(11)"
    @"SPC(6)"50":PRINT"{BLU}{2 DOWN}
    {RIGHT}DIFFICULTY LEVELS..."        :rem 54
630 PRINT"{RED}{DOWN}{4 RIGHT}[1] BEGINNE
    R":PRINT"{DOWN}{4 RIGHT}[2] ADVANCED"
    :PRINT"{DOWN}{4 RIGHT}[3] MASTER"
                                        :rem 53
640 PRINT"{BLU}{DOWN}{RIGHT}KEY IN YOUR L
    EVEL:"                              :rem 57
650 GETA$:IFA$=""THEN650                :rem 89
660 H=VAL(A$):IFH<1ORH>3THEN650         :rem 92
670 RETURN                              :rem 125
700 P1=8043:SH=39:CM=30720:S1=37154:SQ=8:
    SA=7702:SE=8075                     :rem 181
710 CT=38400:CE=38905:S2=37137:S3=37152:A
    Q=66:BN=SA                          :rem 131
720 PRINT"{CLR}":POKE36879,INT(RND(1)*5)+
    10:FORI=CTTOCE:POKEI,3:NEXT         :rem 35
730 FORI=1TO50:A=INT(RND(1)*372)+SA:POKEC
    M+A,5:POKEA,43:NEXT                 :rem 155
740 FORI=8076TO8119:POKEI+CM,7:POKEI,0:NE
    XT                                  :rem 170
750 FORI=8120TO8185:POKEI+CM,INT(RND(1)*6
    )+1:POKEI,41:NEXT                   :rem 0
760 PRINT"{HOME}{YEL}SCORE:"SC;TAB(13)"SH
    IPS:"SQ:POKEP1+CM,3:POKEP1,SH:POKE368
    74,150:RETURN                       :rem 225
800 FORI=1TO8:READSP(I):NEXT            :rem 29
810 DATA 1,23,22,21,-1,-23,-22,-21:rem 85
820 FORI=7432TO7503:READA:POKEI,A:NEXT:FO
    RI=7520TO7527:READA:POKEI,A:NEXT
                                        :rem 148
830 FORI=7168TO7175:POKEI,255:NEXT:FORI=7
    512TO7519:POKEI,PEEK(I+25960):NEXT
                                        :rem 126
840 DATA0,96,112,120,207,120,112,96,8,92,
    60,108,244,126,6,1                  :rem 123
850 DATA16,254,254,108,56,16,16,16,16,58,
    60,54,47,126,96,128                 :rem 210
860 DATA0,6,14,30,243,30,14,6,128,96,126,
    47,54,60,58,16                      :rem 194
870 DATA16,16,16,56,108,254,254,16,1,6,12
    6,244,108,60,92,8                   :rem 95
880 DATA195,231,231,126,60,219,189,36,66,
    102,129,189,126,219,189,36          :rem 60
890 RETURN                              :rem 129
```

Program 2: Beekeeper For 64

```
1 POKE56,48:POKE55,0:CLR                :rem 173
5 POKE53280,2:POKE53281,0               :rem 140
10 DIMSP(8)                             :rem 103
20 PRINT"{CLR}"                         :rem 198
30 V=1:SC=0:CM=54272                    :rem 111
40 GOSUB500:GOSUB800:GOSUB600           :rem 27
41 POKECM+24,15:POKECM+5,17:POKECM+6,241:
   POKECM,0:POKECM+12,17:POKECM+13,241
                                        :rem 12
42 POKECM+7,0                           :rem 227
45 GOSUB700                             :rem 128
50 S=PEEK(56320):SW=(SAND4)/4:F=(SAND16)/
   16                                   :rem 203
60 SR=(SAND8)/8                         :rem 94
70 IFF=1THEN130                         :rem 109
80 I=1:A=SH-32:J=P1:POKECM+8,200:POKECM+1
   1,129                                :rem 174
90 J=J+SP(A):IFJ<SAORJ>2023THEN120:rem 47
100 IFPEEK(J)<>32THENPOKEBN,32:GOSUB310:G
    OTO120                              :rem 146
110 POKEBN,32:POKEJ,42:POKEBBN+CM,3:BN=J:
    I=I+1:IFI<8THEN90                   :rem 80
120 POKEBN,32:POKECM+11,128             :rem 48
130 IFSW=1THENIFSR=1THEN200             :rem 205
140 CC=1:IFSW=0THENCC=-1                :rem 185
150 POKECM+1,50:POKECM+4,33:IFP1=P2THENJ=
    P1:GOSUB310                         :rem 186
160 IFSH=40THENIFCC=1THENCC=-7          :rem 106
170 IFSH=33THENIFCC=-1THENCC=7          :rem 109
180 SH=SH+CC:POKEP1,SH:POKECM+4,32:rem 87
200 A=SH-32:MN=P1:P1=P1+SP(A):IFP1<SAORP1
    >SETHENP1=MN                        :rem 67
210 IFPEEK(P1)<>32THENJ=P1:POKEMN,32:GOSU
    B300                                :rem 17
220 POKEMN,32:POKEP1,SH:IFW=1THEN250
                                        :rem 244
230 P2=INT(RND(1)*21)+1783:MC=INT(RND(1)*
    6)+1:SX=41:SY=32:BC=3:W=1           :rem 35
240 IFMC=2THENSX=44:SY=43:BC=5          :rem 94
250 A=SGN(P2-P1):M0=P2:P2=P2-A*H:IFABS(P2
    -P1)>25THENP2=P2-A*40               :rem 128
260 IFP2<SAORP2>SETHENP2=M0             :rem 221
270 IFA=0THENJ=P2:GOSUB310:GOTO50       :rem 52
280 POKECM+P2,MC:POKEM0,SY:POKEP2,SX:POKE
    CM+M0,BC:GOTO50                     :rem 108
300 A=PEEK(J):IFA=43ORA<41THENP1=MN:GOTO4
    80                                  :rem 153
310 FORI=1TO5:POKEJ,42:POKEJ+CM,2:POKECM+
    7,50:POKECM+11,129                  :rem 85
311 POKECM+1,60:POKECM+4,33:POKEJ,32:NEXT
                                        :rem 233
320 POKEJ+CM,3:N=SQ:IFJ=P2THENW=0:SC=SC+1
    50                                  :rem 186
330 IFP1=P2ORA=41THENSQ=SQ-1:P1=1764:SH=3
    9                                   :rem 67
340 IFJ>1903THENSC=SC+50:AQ=AQ-1        :rem 156
350 SC=SC+50:IFSC>HITHENHI=SC           :rem 82
370 PRINT"{HOME}{YEL}SCORE:"SC;TAB(32)"SH
    IPS:"SQ:POKEP1+CM,3:POKEP1,SH:POKECM+
    11,128                              :rem 219
380 IFSQ=NTHEN410                       :rem 29
390 FORI=1TO5:FORS=10TO80STEP2:POKECM+1,S
    :POKECM+4,33:FORA=1TO10:NEXT:NEXT
                                        :rem 33
400 POKECM+4,32:FORJ=1TO100:NEXT:NEXT
                                        :rem 66
410 IFAQ=0THENV=V+1:PRINT"{CLR}{RED}
    {11 DOWN}{8 RIGHT}SWARM"V:FORI=1TO400
    0:NEXT:GOTO45                       :rem 104
420 IFSQ<>0THEN480                      :rem 62
430 PRINT"{HOME}{RED}{8 DOWN}"SPC(14)"* G
    AME OVER *"                         :rem 169
435 PRINTSPC(11)"{4 DOWN}TRY AGAIN? [Y OR
     N]"                                :rem 232
440 GETA$:IFA$=""THEN440                :rem 83
450 IFA$="Y"THENRESTORE:GOTO20          :rem 143
460 IFA$<>"N"THEN440                    :rem 97
```

```
470 PRINT"{CLR}":END                    :rem 16
480 RETURN                              :rem 124
500 PRINT"{RED}{8 DOWN}{14 RIGHT}*{CYN}BE
    EKEEPER{RED}*"                      :rem 55
510 PRINT"{GRN}{8 DOWN}{14 RIGHT}HI SCORE
    ={YEL}"HI                           :rem 28
520 FORI=1TO3000:NEXT:RETURN            :rem 46
600 PRINT"{CLR}"SPC(10)"{RED}{DOWN}USE JO
    YSTICK TO PLAY"                     :rem 57
605 PRINTSPC(10)"{BLU}{2 DOWN}BEE"TAB(20)
    ")"SPC(5)"100"                      :rem 73
610 PRINTSPC(10)"{GRN}{DOWN}CLOVER"TAB(20
    )"+"SPC(6)"50"                      :rem 9
615 PRINTSPC(10)"{RED}{DOWN}CRAB"TAB(20)"
    ,"SPC(5)"200"                       :rem 134
620 PRINTSPC(10)"{YEL}{DOWN}HIVE"TAB(20)"
    @"SPC(6)"50"                        :rem 0
625 PRINTSPC(10)"{BLU}{2 DOWN}DIFFICULTY
    {SPACE}LEVELS..."                   :rem 143
630 PRINTSPC(14)"{RED}{DOWN}[1] BEGINNER"
                                        :rem 102
635 PRINTSPC(14)"{DOWN}[2] ADVANCED":PRIN
    TSPC(14)"{DOWN}[3] MASTER"          :rem 171
640 PRINTSPC(11)"{BLU}{DOWN}KEY IN YOUR L
    EVEL:"                              :rem 181
650 GETA$:IFA$=""THEN650                :rem 89
660 H=VAL(A$):IFH<1ORH>3THEN650         :rem 92
670 RETURN                              :rem 125
700 P1=1764:SH=39:SQ=8:SA=1064:SE=1823
                                        :rem 171
710 CT=55296:CE=56295:AQ=66:BN=SA:rem 145
720 PRINT"{CLR}":FORI=CTTOCE:POKEI,3:NEXT
                                        :rem 240
730 FORI=1TO50:A=INT(RND(1)*758)+SA:POKEC
    M+A,5:POKEA,43:NEXT                 :rem 163
740 FORI=1824TO1903:POKEI+CM,7:POKEI,0:NE
    XT                                  :rem 158
750 FORI=1904TO2023:POKEI+CM,INT(RND(1)*6
    )+1:POKEI,41:NEXT                   :rem 244
760 PRINT"{HOME}{YEL}SCORE:"SC;TAB(32)"SH
    IPS:"SQ:POKEP1+CM,3:POKEP1,SH:RETURN
                                        :rem 171
800 FORI=1TO8:READSP(I):NEXT:IFPEEK(251)=
    123THENPOKE53272,29:RETURN          :rem 217
805 PRINT"{7 UP}{9 RIGHT}REDEFINING CHARA
    CTERS"                              :rem 37
807 POKE251,123                         :rem 40
810 DATA 1,41,40,39,-1,-41,-40,-39
                                        :rem 103
815 POKE56334,PEEK(56334)AND254:POKE1,PEE
    K(1)AND251                          :rem 191
816 FORI=0TO511:POKEI+12288,PEEK(I+53248)
    :NEXT                               :rem 237
820 FORI=12552TO12623:READA:POKEI,A:NEXT:
    FORI=12640TO12647:READA:POKEI,A:NEXT
                                        :rem 80
830 FORI=12288TO12295:POKEI,255:NEXT:FORI
    =12632TO12639:READA:POKEI,A:NEXT
                                        :rem 34
835 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334
    )OR1                                :rem 143
836 POKE53272,(PEEK(53272)AND240)+12
                                        :rem 192
840 DATA0,96,112,120,207,120,112,96,8,92,
    60,108,244,126,6,1                  :rem 123
850 DATA16,254,254,108,56,16,16,16,16,58,
    60,54,47,126,96,128                 :rem 210
860 DATA0,6,14,30,243,30,14,6,128,96,126,
    47,54,60,58,16                      :rem 194
870 DATA16,16,16,56,108,254,254,16,1,6,12
    6,244,108,60,92,8                   :rem 95
880 DATA195,231,231,126,60,219,189,36,66,
    102,129,189,126,219,189,36          :rem 60
885 DATA24,24,102,102,24,24,60,0        :rem 71
890 RETURN                              :rem 129
```

Ultrafont +

See article on page 28.

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.

```
49152 :076,200,196,000,001,003,220
49158 :004,000,173,048,002,072,049
49164 :173,045,002,141,048,002,167
49170 :141,079,002,032,043,193,252
49176 :104,141,048,002,169,100,076
49182 :133,252,169,000,133,251,200
49188 :133,167,169,216,133,168,254
49194 :169,008,141,040,002,169,059
49200 :002,141,042,002,169,005,153
49206 :141,041,002,174,003,192,095
49212 :173,079,002,205,048,002,057
49218 :208,002,162,006,142,080,154
49224 :002,160,000,177,253,170,066
49230 :173,063,002,240,003,076,123
49236 :229,192,169,207,145,251,253
49242 :138,010,170,176,008,173,253
49248 :080,002,145,167,076,108,162
49254 :192,173,004,192,145,167,207
49260 :200,192,008,208,221,024,193
49266 :165,251,105,008,133,251,003
49272 :133,167,165,252,105,000,174
49278 :133,252,105,116,133,168,009
49284 :024,165,253,105,008,133,052
49290 :253,165,254,105,000,133,024
49296 :254,056,238,079,002,206,211
49302 :041,002,173,041,002,208,105
49308 :156,056,173,079,002,233,087
49314 :005,141,079,002,056,165,098
49320 :253,233,039,133,253,165,220
49326 :254,233,000,133,254,206,230
49332 :040,002,173,040,002,240,165
49338 :003,076,052,192,206,042,245
49344 :002,173,042,002,240,030,169
49350 :169,008,141,040,002,024,070
49356 :173,079,002,105,032,141,224
49362 :079,002,024,165,253,105,070
49368 :248,133,253,165,254,105,094
49374 :000,133,254,076,052,192,161
49380 :096,134,097,169,000,141,097
49386 :043,002,006,097,046,043,215
49392 :002,006,097,046,043,002,180
49398 :174,043,002,169,207,145,218
49404 :251,200,169,247,145,251,235
49410 :136,189,003,192,145,167,066
49416 :200,145,167,200,192,008,152
49422 :208,215,076,113,192,169,219
49428 :000,141,026,208,165,001,049
49434 :041,251,133,001,096,165,201
49440 :001,009,004,133,001,169,093
49446 :001,141,026,208,096,169,167
49452 :000,133,254,173,048,002,142
```

```
49458 :133,253,006,253,038,254,219
49464 :006,253,038,254,006,253,098
49470 :038,254,024,169,112,101,248
49476 :254,133,254,096,032,043,112
49482 :193,160,000,177,253,073,162
49488 :255,145,253,200,192,008,109
49494 :208,245,032,008,192,096,099
49500 :169,102,133,252,169,218,111
49506 :133,168,169,132,133,251,060
49512 :133,167,162,008,169,000,231
49518 :133,097,160,000,165,097,250
49524 :145,251,230,097,173,058,046
49530 :002,145,167,200,192,032,092
49536 :208,240,024,165,251,105,097
49542 :040,133,251,133,167,165,255
49548 :252,105,000,133,252,105,219
49554 :116,133,168,202,208,216,165
49560 :096,032,043,203,173,044,231
49566 :002,141,024,208,169,200,134
49572 :013,063,002,141,022,208,101
49578 :169,000,141,032,208,141,093
49584 :033,208,032,092,193,173,139
49590 :058,002,141,134,002,165,172
49596 :209,133,243,024,165,210,148
49602 :105,116,133,244,164,211,143
49608 :177,209,073,128,145,209,117
49614 :177,243,072,173,134,002,239
49620 :145,243,032,228,255,240,075
49626 :251,170,164,211,177,209,120
49632 :073,128,145,209,104,145,004

49638 :243,138,032,210,255,032,116
49644 :225,255,208,203,032,075,210
49650 :203,169,000,141,134,002,123
49656 :169,012,141,032,208,076,118
49662 :094,196,032,019,193,169,189
49668 :112,133,252,173,082,002,246
49674 :133,254,162,008,169,000,224
49680 :133,253,133,251,168,177,107
49686 :253,145,251,200,208,249,048
49692 :230,254,230,252,202,208,124
49698 :242,165,252,201,128,240,238
49704 :007,169,208,133,254,076,119
49710 :012,194,032,031,193,096,092
49716 :169,112,133,252,169,116,235
49722 :133,254,169,000,133,253,232
49728 :133,251,168,162,004,177,191
49734 :251,073,255,145,253,200,223
49740 :208,247,230,254,230,252,217
49746 :202,208,240,096,032,043,135
49752 :193,160,000,177,253,010,113
49758 :008,074,040,042,145,253,144
49764 :200,192,008,208,242,076,002
49770 :008,192,032,043,193,160,222
49776 :000,177,253,074,008,010,122
49782 :040,106,145,253,200,192,030
49788 :008,208,242,076,008,192,090
49794 :032,043,193,160,000,177,223
49800 :253,133,097,200,177,253,225
49806 :136,145,253,200,200,192,244
49812 :008,208,245,165,097,136,239

49818 :145,253,076,008,192,032,092
49824 :043,193,160,007,177,253,225
49830 :133,097,136,177,253,200,138
49836 :145,253,136,016,247,200,145
49842 :165,097,145,253,076,008,154
49848 :192,032,043,193,160,000,036
49854 :169,000,133,097,162,008,247
49860 :177,253,010,102,097,202,013
49866 :208,250,165,097,145,253,040

49872 :200,192,008,208,233,076,101
49878 :008,192,032,043,193,160,074
49884 :008,169,000,153,048,002,088
49890 :136,208,250,169,007,133,105
49896 :097,152,170,169,000,133,185
49902 :007,177,253,074,145,253,123
49908 :038,007,202,016,251,166,156
49914 :097,165,007,029,049,002,087
49920 :157,049,002,198,097,165,156
49926 :097,016,224,200,192,008,231
49932 :208,215,136,185,049,002,039
49938 :145,253,136,016,248,076,124
49944 :008,192,032,043,193,160,140
49950 :000,152,145,253,200,192,204
49956 :008,208,249,076,008,192,009
49962 :120,169,127,141,013,220,064
49968 :169,001,141,026,208,169,250
49974 :177,141,018,208,169,027,026
49980 :141,017,208,169,075,141,043
49986 :020,003,169,195,141,021,103
49992 :003,088,096,173,018,208,146
49998 :201,177,208,039,169,242,090
50004 :141,018,208,173,044,002,158
50010 :141,024,208,173,022,208,098
50016 :041,239,013,063,002,141,083
50022 :022,208,173,057,002,141,193
50028 :033,208,169,001,141,025,173
50034 :208,104,168,104,170,104,204
50040 :064,169,177,141,018,208,129
50046 :169,158,141,024,208,173,231

50052 :032,208,141,033,208,169,155
50058 :200,141,022,208,238,037,216
50064 :208,169,001,141,025,208,128
50070 :076,049,234,085,064,000,146
50076 :064,064,000,076,064,000,168
50082 :076,064,000,076,064,000,186
50088 :076,064,000,064,064,000,180
50094 :085,064,000,000,000,085,152
50100 :080,000,064,016,000,064,148
50106 :016,000,064,016,000,064,090
50112 :016,000,064,016,000,064,096
50118 :016,000,064,016,000,064,102
50124 :016,000,085,080,000,000,129
50130 :000,000,255,255,255,000,207
50136 :001,001,001,000,255,001,219
50142 :000,000,255,001,000,000,222
50148 :255,001,018,085,076,084,235
50154 :082,065,070,079,078,084,180
50160 :032,043,146,095,069,082,195
50166 :082,079,082,032,079,078,166
50172 :032,083,065,086,069,047,122
50178 :076,079,065,068,095,018,147
50184 :084,146,065,080,069,032,228
50190 :079,082,032,018,068,146,183
50196 :073,083,075,063,095,070,223
50202 :073,076,069,078,065,077,208
50208 :069,058,095,069,078,084,229
50214 :069,082,032,067,079,076,187
50220 :079,082,032,075,069,089,214
50226 :095,085,083,069,032,082,240
50232 :079,077,032,083,069,084,224
50238 :063,032,040,089,047,078,155
50244 :041,095,018,085,146,080,021
50250 :080,069,082,067,065,083,008
50256 :069,032,079,082,032,018,136
50262 :076,146,079,087,069,082,113

50268 :063,095,169,230,160,195,236
50274 :133,251,132,252,160,040,042
50280 :169,032,153,191,103,136,120
```

```
50286 :208,250,177,251,200,201,117
50292 :095,208,249,136,132,097,009
50298 :152,074,073,255,056,105,069
50304 :020,168,162,024,024,032,046
50310 :240,255,160,000,177,251,193
50316 :032,210,255,200,196,097,106
50322 :144,246,096,133,251,132,124
50328 :252,160,040,169,032,153,190
50334 :191,103,136,208,250,162,184
50340 :024,160,000,024,032,240,132
50346 :255,160,000,177,251,201,190
50352 :095,240,006,032,210,255,246
50358 :200,208,244,096,174,076,156
50364 :002,240,008,160,000,200,030
50370 :208,253,202,208,250,096,131
50376 :173,002,221,009,003,141,237
50382 :002,221,173,000,221,041,096
50388 :252,009,002,141,000,221,069
50394 :169,100,141,136,002,169,167
50400 :147,032,210,255,169,000,013
50406 :141,134,002,169,008,032,204
50412 :210,255,160,000,152,153,142
50418 :128,099,200,016,250,168,079
50424 :185,153,195,153,128,099,137
50430 :200,192,023,208,245,160,002
50436 :000,185,176,195,153,192,137
50442 :099,200,192,032,208,245,218
50448 :169,156,141,044,002,169,185
50454 :012,141,032,208,169,128,200
50460 :141,138,002,032,042,195,066
50466 :169,048,141,076,002,169,127
50472 :011,141,057,002,169,007,171
50478 :169,000,141,048,002,141,035
50484 :045,002,141,063,002,173,222
50490 :006,192,009,008,141,058,216
50496 :002,173,004,192,141,034,098
50502 :208,173,005,192,141,035,056
50508 :208,032,008,192,032,092,128
50514 :193,169,203,205,007,192,027
50520 :240,014,141,007,192,162,076
50526 :208,142,082,002,032,000,048
50532 :194,076,120,197,169,051,139
50538 :160,196,032,098,196,032,052
50544 :228,255,240,251,201,078,085
50550 :240,029,169,070,160,196,214
50556 :032,098,196,032,228,255,197
50562 :240,251,162,208,201,076,244
50568 :208,002,162,216,142,082,180
50574 :002,032,000,194,032,008,154
50580 :192,032,094,196,169,142,205
50586 :141,248,103,169,143,141,075
50592 :249,103,169,003,141,021,078
50598 :208,169,024,141,000,208,148
50604 :169,000,141,016,208,169,107
50610 :051,141,001,208,169,176,156
50616 :141,003,208,169,053,141,131
50622 :002,208,169,000,141,029,227
50628 :208,141,023,208,141,038,187
50634 :208,169,003,141,028,208,191
50640 :169,000,141,059,002,141,208
50646 :060,002,173,000,220,072,229
50652 :041,015,073,015,141,061,054
50658 :002,104,041,016,141,062,080
50664 :002,032,228,255,240,006,227
50670 :032,109,199,076,216,197,043
50676 :032,186,196,173,062,002,127
50682 :208,003,032,000,199,173,097
50688 :062,002,073,016,141,075,113
50694 :002,173,061,002,240,204,176
50700 :174,061,002,189,208,195,073
50706 :172,063,002,240,001,010,250
50712 :024,109,059,002,141,059,162
50718 :002,024,173,060,002,125,160
50724 :219,195,141,060,002,174,059
50730 :059,002,016,025,162,000,050
50736 :142,059,002,173,045,002,215
50742 :240,015,206,045,002,162,212
50748 :007,173,063,002,240,002,035
50754 :162,006,142,059,002,174,099
50760 :059,002,224,040,144,022,051
50766 :162,039,142,059,002,173,143
50772 :045,002,201,219,176,010,225
50778 :105,001,141,045,002,162,034
50784 :032,142,059,002,172,060,051
50790 :002,016,022,160,000,140,186
50796 :060,002,173,045,002,201,079
50802 :032,144,010,233,032,141,194
50808 :045,002,160,007,140,060,022
50814 :002,172,060,002,192,016,058
50820 :144,022,160,015,140,060,161
50826 :002,173,045,002,201,192,241
50832 :176,010,105,032,141,045,141
50838 :002,160,008,140,060,002,010
50844 :173,059,002,172,060,002,112
50850 :074,074,074,192,008,144,216
50856 :002,105,031,109,045,002,206
50862 :141,048,002,041,224,074,192
50868 :074,105,176,141,003,208,119
50874 :173,048,002,041,031,010,235
50880 :010,010,105,053,141,002,001
50886 :208,169,000,105,000,133,045
50892 :097,173,060,002,010,010,044
50898 :010,105,051,141,001,208,214
50904 :173,059,002,010,010,010,224
50910 :038,097,105,024,141,000,115
50916 :208,165,097,105,000,141,176
50922 :016,208,173,048,002,205,118
50928 :081,002,240,009,032,008,100
50934 :192,173,048,002,141,081,115
50940 :002,076,216,197,032,043,050
50946 :193,173,060,002,041,007,222
50952 :168,173,059,002,041,007,202
50958 :073,007,170,232,134,097,215
50964 :056,169,000,042,202,208,185
50970 :252,174,063,002,208,048,005
50976 :133,097,173,075,002,208,208
50982 :022,169,000,141,064,002,180
50988 :141,038,208,177,253,037,130
50994 :097,208,008,169,001,141,162
51000 :064,002,141,038,208,165,162
51006 :097,073,255,049,253,174,195
51012 :064,002,240,002,005,097,222
51018 :145,253,032,008,192,096,032
51024 :133,098,074,005,098,073,049
51030 :255,049,253,166,097,202,084
51036 :133,097,173,066,002,074,125
51042 :042,202,208,252,005,097,136
51048 :145,253,076,008,192,141,151
51054 :065,002,174,137,199,221,140
51060 :137,199,240,004,202,208,082
51066 :248,096,202,138,010,170,218
51072 :189,173,199,072,189,172,098
51078 :199,072,096,034,133,137,037
51084 :134,138,077,082,147,018,224
51090 :145,017,157,029,070,135,187
51096 :139,049,050,051,052,019,000
51102 :136,140,033,034,035,036,060
51108 :086,083,076,024,004,006,187
51114 :131,084,107,194,085,194,197
51120 :158,194,129,194,184,194,205
```

```
51126 :215,194,025,195,071,193,051
51132 :248,199,014,200,036,200,061
51138 :058,200,082,200,117,200,027
51144 :160,200,172,200,172,200,024
51150 :172,200,172,200,189,200,059
51156 :214,200,236,200,014,201,253
51162 :014,201,014,201,014,201,095
51168 :085,201,136,202,020,203,047
51174 :036,203,160,203,051,194,053
51180 :239,199,152,193,162,255,156
51186 :154,032,129,255,076,200,064
51192 :196,173,060,002,041,007,215
51198 :133,097,056,173,060,002,007
51204 :233,008,056,229,097,141,000
51210 :060,002,076,078,200,173,087
51216 :060,002,041,007,133,097,100
51222 :024,173,060,002,105,008,138
51228 :056,229,097,141,060,002,101
51234 :076,078,200,173,059,002,110
51240 :041,007,133,097,056,173,035
51246 :059,002,233,008,056,229,121
51252 :097,141,059,002,076,078,249
51258 :200,173,059,002,041,007,028
51264 :133,097,024,173,059,002,040
51270 :105,008,056,229,097,141,194
51276 :059,002,104,104,076,041,206
51282 :198,032,043,193,032,019,087
51288 :193,160,007,024,165,254,123
51294 :105,096,141,106,200,165,139
51300 :253,141,105,200,185,000,216
51306 :208,145,253,136,016,248,088
51312 :032,031,193,076,008,192,132
51318 :169,016,141,063,002,169,166
51324 :001,141,029,208,032,008,031
51330 :192,173,058,002,009,008,060
51336 :141,058,002,032,092,193,142
51342 :169,050,141,065,002,032,089
51348 :173,200,173,059,002,041,028
51354 :254,141,059,002,076,078,252
51360 :200,169,000,141,063,002,223
51366 :141,029,208,032,008,192,008
51372 :096,056,173,065,002,233,029
51378 :049,141,066,002,170,189,027
51384 :003,192,141,038,208,096,094
51390 :173,059,002,013,060,002,243
51396 :208,003,141,045,002,169,252
51402 :000,141,059,002,141,060,093
51408 :002,032,008,192,076,078,084
51414 :200,032,072,193,032,072,047
51420 :193,032,043,193,160,000,073
51426 :177,253,153,067,002,200,054
51432 :192,008,208,246,096,032,246
51438 :043,193,160,000,185,067,118
51444 :002,145,253,200,192,008,020
51450 :208,246,076,008,192,144,100
51456 :005,028,159,156,030,031,153
51462 :158,129,149,150,151,152,127
51468 :153,154,155,169,035,160,070
51474 :196,032,098,196,032,228,032
51480 :255,240,251,162,000,221,129
51486 :255,200,240,008,232,224,165
51492 :016,208,246,076,094,196,104
51498 :056,173,065,002,233,033,092
51504 :168,138,153,003,192,192,126
51510 :003,240,010,192,000,240,227
51516 :019,153,033,208,076,080,117
51522 :201,174,063,002,240,002,236
51528 :009,008,141,058,002,032,066
51534 :092,193,032,008,192,076,159
51540 :094,196,169,122,160,201,002
51546 :032,098,196,032,228,255,163
51552 :056,233,048,048,248,201,162
51558 :010,176,244,133,097,056,050
51564 :169,009,229,097,010,010,120
51570 :010,010,141,076,002,076,173
51576 :094,196,067,085,082,083,215
51582 :079,082,032,086,069,076,038
51588 :079,067,073,084,089,032,044
51594 :040,048,045,057,041,063,176
51600 :095,160,000,140,078,002,107
51606 :169,164,032,210,255,169,125
51612 :157,032,210,255,032,228,046
51618 :255,240,251,172,078,002,136
51624 :133,097,169,032,032,210,073
51630 :255,169,157,032,210,255,228
51636 :165,097,201,013,240,039,167
51642 :201,020,208,013,192,000,052
51648 :240,209,136,169,157,032,111
51654 :210,255,076,147,201,041,104
51660 :127,201,032,144,194,192,070
51666 :020,240,190,165,097,153,051
51672 :000,002,032,210,255,200,147
51678 :076,147,201,169,095,153,039
51684 :000,002,152,096,032,231,229
51690 :255,169,007,160,196,032,029
51696 :098,196,032,228,255,240,009
51702 :251,162,001,201,084,240,161
51708 :011,162,008,201,068,240,174
51714 :005,104,104,076,094,196,069
51720 :141,077,002,160,001,169,046
51726 :001,032,186,255,169,025,170
51732 :160,196,032,149,196,032,017
51738 :145,201,208,007,173,077,069
51744 :002,201,084,208,237,173,169
51750 :077,002,201,068,208,066,148
51756 :169,064,141,020,002,169,097
51762 :048,141,021,002,169,058,233
51768 :141,022,002,160,000,185,054
51774 :000,002,153,023,002,200,186
51780 :204,078,002,208,244,169,205
51786 :044,153,023,002,169,080,033
51792 :153,024,002,173,065,002,243
51798 :201,083,208,012,169,044,035
51804 :153,025,002,169,087,153,169
51810 :026,002,200,200,200,200,158
51816 :200,200,200,076,124,202,082
51822 :160,000,185,000,002,153,098
51828 :020,002,200,204,078,002,110
51834 :208,244,152,162,020,160,044
51840 :002,032,189,255,169,160,167
51846 :133,178,096,032,232,201,238
51852 :032,043,203,169,000,133,208
51858 :253,133,251,169,112,133,173
51864 :252,162,255,160,119,169,245
51870 :251,032,216,255,176,011,075
51876 :032,183,255,208,006,032,112
51882 :075,203,076,094,196,032,078
51888 :075,203,032,231,255,173,121
51894 :077,002,201,068,240,015,017
51900 :169,244,160,195,032,098,062
51906 :196,032,228,255,240,251,116
51912 :076,094,196,169,000,032,255
51918 :189,255,169,015,162,008,236
51924 :160,015,032,186,255,032,124
51930 :192,255,162,015,032,198,048
51936 :255,160,000,032,207,255,109
51942 :201,013,240,007,153,000,076
51948 :002,200,076,227,202,169,088
```

```
51954 :095,153,000,002,032,204,216
51960 :255,169,000,160,002,032,098
51966 :098,196,162,015,032,201,190
51972 :255,169,073,032,210,255,230
51978 :169,013,032,210,255,032,209
51984 :231,255,076,195,202,032,239
51990 :232,201,032,043,203,169,134
51996 :000,032,213,255,176,141,077
52002 :076,075,203,169,004,141,190
52008 :136,002,000,120,169,000,211
52014 :141,026,208,169,255,141,218
52020 :013,220,169,049,141,020,152
52026 :003,169,234,141,021,003,117
52032 :169,000,141,021,208,169,004
52038 :147,088,076,210,255,032,110
52044 :042,195,169,003,141,021,135
52050 :208,032,008,192,032,092,134
52056 :193,076,094,196,248,169,040
52062 :000,141,000,001,141,001,122
52068 :001,224,000,240,021,202,020
52074 :024,173,000,001,105,001,154
52080 :141,000,001,173,001,001,173
52086 :105,000,141,001,001,076,186
52092 :101,203,216,173,001,001,051
52098 :009,048,141,002,001,173,248
52104 :000,001,041,240,074,074,054
52110 :074,074,009,048,141,001,233
52116 :001,173,000,001,041,015,123
52122 :009,048,141,000,001,096,193
52128 :096,056,165,045,233,002,245
52134 :133,045,165,046,233,000,020
52140 :133,046,169,024,133,057,222
52146 :169,246,133,058,169,000,185
52152 :141,079,002,133,251,133,155

52158 :253,169,112,133,254,169,000
52164 :208,133,252,032,019,193,009
52170 :160,000,177,251,209,253,228
52176 :208,058,200,192,008,208,058
52182 :245,238,079,002,024,165,199
52188 :253,105,008,133,253,133,081
52194 :251,165,254,105,000,133,110
52200 :254,105,096,133,252,201,249
52206 :216,208,217,169,000,168,192
52212 :145,045,200,145,045,024,080
52218 :165,045,105,002,133,045,233
52224 :165,046,105,000,133,046,239
52230 :032,031,193,076,051,165,042
52236 :160,000,024,165,045,105,255
52242 :041,145,045,200,165,046,148
52248 :105,000,145,045,200,165,172
52254 :057,145,045,200,165,058,188
52260 :145,045,200,169,131,145,103
52266 :045,174,079,002,032,092,210
52272 :203,200,173,002,001,145,004
52278 :045,200,173,001,001,145,107
52284 :045,200,173,000,001,145,112
52290 :045,200,132,097,160,000,188
52296 :132,098,177,253,170,032,166
52302 :092,203,164,097,169,044,079
52308 :145,045,200,173,002,001,138
52314 :145,045,173,001,001,200,143
52320 :145,045,173,000,001,200,148
52326 :145,045,200,132,097,164,117
52332 :098,200,192,008,208,214,004
52338 :164,097,169,000,145,045,222
52344 :160,000,177,045,072,200,006
52350 :177,045,133,046,104,133,252
52356 :045,230,057,208,002,230,136
52362 :058,076,215,203,013,013,204
```

Bonking Barrels

See article on page 50.

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:

Bonking Barrels—VIC Version

```
12 PRINT"{CLR}"                          :rem 199
13 GOTO33                                 :rem 3
14 REM MAN JUMPS                         :rem 180
15 POKEE%,39:GOSUB19:POKEE%,37:POKES,P:P=
   P+3:POKEE%,38:GOSUB19                 :rem 95
16 IFPEEK(E%-22)=LLTHEN77                :rem 217
17 POKEE%,32:E%=E%-22:POKEE%,38:POKES,0:P
   OKEE%,32:E%=E%-22:POKEE%,38:RETURN
                                         :rem 180
18 REM MOVE BARRELS                      :rem 143
19 POKES,0                               :rem 103
20 FORX=1TOZ%:POKEB%(X),C                 :rem 25
21 IFPEEK(B%(X)+I%)=WTHENB%(X)=B%(X)+D(X)
                                          :rem 73
22 IFPEEK(B%(X))=WTHENB%(X)=B%(X)-T%*D(X)
                                          :rem 86
23 IFPEEK(B%(X)+I%)<>WTHENB%(X)=B%(X)+I%
                                           :rem 9
24 IFPEEK(B%(X))>36ANDPEEK(B%(X))<40THEN7
   7                                     :rem 184
25 POKEB%(X),LL                          :rem 106
26 IFB%(X)>JTHENGOSUB29                  :rem 169
27 NEXTX:RETURN                           :rem 26
28 REM RESET BARRELS                     :rem 220
29 FORX=1TOZ%:POKEB%(X),C:B%(X)=SR+INT(RN
   D(X)*T%):NEXTX                        :rem 182
30 REM SCORE                              :rem 2
31 PRINT"{HOME}";SPC(242);SPC(244);"{RVS}
   SCORE{4 SPACES}{4 LEFT}"SC;"{9 SPACES}
   {6 LEFT}";                            :rem 221
32 FORQ=3TOA+1STEP-1:PRINT"{OFF}%";:NEXT:
   PRINT"{HOME}":RETURN                  :rem 149
33 SR=7681:K=38400-SR+1                   :rem 13
34 POKE36879,14                           :rem 56
35 POKE36869,255                         :rem 111
36 BL$="{RVS}{YEL}{22 SPACES}"            :rem 76
37 AL$= "{OFF}{PUR}$$$$$$$$$$$$$$$$$$$$$$
   "                                     :rem 226
38 J=SR+449:V=36878:S=36876               :rem 24
39 FORI=7424TO7424+8*8-1:READA:POKEI,A:NE
   XT:GOSUB91                            :rem 148
40 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,56
   ,108,206,174,234,230,108,56:REM BARREL
                                         :rem 178
41 DATA56,108,246,250,134,222,108,56:REM
   {SPACE}BARREL                         :rem 242
42 DATA255,153,189,153,255,0,0,0:REM BEAM
                                         :rem 131
43 DATA56,56,16,56,84,16,56,40:REM MAN NO
   RMAL                                  :rem 194
44 DATA186,186,84,56,16,16,56,40,0,0,0,0,56
   ,56,16,124,170:REM MAN STRETCHING,SQUA
   TING                                  :rem 81
45 PRINT"{CLR}":A=0:SC=0:Z%=4             :rem 28
```

```
46 REM SET UP SCREEN                    :rem 159
47 PRINT"{HOME}";:BL=60                 :rem 245
48 FORI=1TO10:PRINTBL$;AL$;:NEXT:PRINTBL$
   ;BL$;"{HOME}"                        :rem 31
49 X=SR+21:POKEV,15                     :rem 134
50 FORB=1TOBL                           :rem 45
51 R=INT(RND(X)*454)                    :rem 133
52 POKEX+R,32:POKES-1,195+B:POKEX+R+K,7:N
   EXTB                                 :rem 32
53 FORG=-1TO20STEP1                     :rem 163
54 POKEX+G*22,36:POKES-1,200+(G*2):POKEX+
   G*22+K,5                             :rem 20
55 POKEX+21+(G*22),36:POKES-1,0:POKEX+21+
   (G*22)+K,5:NEXTG                     :rem 18
56 REM PREP FOR LOOP                    :rem 167
57 E%=J:D(1)=1:D(2)=-1:D(3)=1:D(4)=-1:L=1
   :LM=17:RM=18:UP=47:BM=39             :rem 89
58 O=0:W=36:C=32:KB=197:T%=20:I%=22:LL=34
   :D=37:Z=245:P=128:TT%=1:TH%=3:GOSUB29
                                        :rem 45
59 REM MAIN GAME LOOP                   :rem 203
60 GOSUB19                              :rem 80
61 IFPEEK(E%)=LLTHENE%=E%+22:POKEE%+K,7:P
   OKEE%+I%,36:POKEE%+I%+K,4:GOTO77
                                        :rem 74
62 IFPEEK(E%-I%)=LLTHEN77               :rem 228
63 IFPEEK(E%+I%)=LLTHENPOKEE%,C:E%=E%+44:
   GOTO77                               :rem 60
64 IFLL=34THENLL=35:GOTO66              :rem 128
65 LL=34                                :rem 167
66 POKEE%,C                             :rem 116
67 IFPEEK(KB)=LMTHENIFPEEK(E%-L)<>WTHENE%
   =E%-L:GOSUB110:POKEE%,D:GOTO71:rem 224
68 IFPEEK(KB)=RMTHENIFPEEK(E%+L)<>WTHENE%
   =E%+L:GOSUB110:POKEE%,D:GOTO71:rem 227
69 IFPEEK(KB)=BMTHENIFPEEK(E%-I%)=WTHENPO
   KEE%-I%,C:SC=SC-5:P=P-10:GOTO71
                                        :rem 241
70 IFPEEK(KB)=UPTHENIFPEEK(E%-I%)<>WTHENG
   OSUB15:SC=SC+2*L:P=P+3:POKEE%,D:rem 40
71 IFF=1THENF=0:GOTO77                  :rem 110
72 POKEE%,D:POKES,P:IFP<128OR P>200THENP=
   128                                  :rem 178
73 IF E%<SRTHENE%=J:TT%=TT%+L:GOSUB31:IFS
   C>35THENZ%=RND(X)*4+1                :rem 65
74 IFTT%>TH%THENGOTO47                  :rem 153
75 GOTO60                               :rem 11
76 REM LOSE MAN                         :rem 96
77 N=15:POKES+1,235:POKEV,N:FORQ=1TO4:POK
   EB%(Q),32:NEXT                       :rem 169
78 POKEE%,38:POKEE%-22,34:GOSUB103:POKEE%
   ,37:GOSUB103:POKEE%,39:GOSUB103:rem 84
79 POKEE%-22,194:POKEE%,35:GOSUB103:POKEE
   %+1,173:POKEE%-1,173:POKEE%+22,194
                                        :rem 51
80 POKEE%-21,206:POKEE%-23,205:POKEE%+23,
   205:POKEE%+21,206:GOSUB103           :rem 22
81 POKEE%-66,174:POKEE%-2,174:POKEE%+2,17
   4:POKEE%+66,174:GOSUB103:POKEE%-42,174
                                        :rem 3
82 POKEE%-46,174:POKEE%+42,174:POKEE%+46,
   174:GOSUB103                         :rem 12
83 POKE36877,0:A=A+1:SC=SC-8:GOSUB31
                                        :rem 128
84 TT%=1:IFA<3GOTO47                    :rem 196
85 PRINT"{CLR}{4 DOWN}{RVS}{6 SPACES}GAME
    OVER"                               :rem 125
86 PRINT"{RVS}{DOWN} PRESS P PLAY AGAIN":
   PRINT"{RVS}{DOWN}{4 SPACES}PRESS E TO
   {SPACE}END";                         :rem 138
87 GETYY$:IFYY$="P"THEN 45              :rem 45
88 IF YY$="E" THEN PRINT"{CLR}":POKE36869
   ,240:END                             :rem 122
89 GOTO87                               :rem 25
90 REM                                  :rem 77
91 PRINT"{CLR}{3 DOWN}{4 RIGHT}{RVS}{YEL}
   *INSTRUCTIONS*"                      :rem 46
92 PRINT"{RVS}{DOWN}{5 RIGHT}A IS LEFT":P
   RINT"{RVS}{5 RIGHT}D IS RIGHT":PRINT"
   {RVS}{4 RIGHT}F3 IS UP"              :rem 79
93 PRINT"{RVS}{4 RIGHT}F1 WILL BLAST
   {12 SPACES}SPACE ABOVE "             :rem 193
94 PRINT"{RVS}{DOWN}{4 RIGHT}2 FOR EACH U
   P":PRINT"{RVS}{3 RIGHT}-5 FOR USING BL
   AST"                                 :rem 93
95 PRINT"{RVS}{3 RIGHT}-8 FOR GETTING HIT
   "                                    :rem 235
96 PRINT "{RVS}{DOWN}{3 RIGHT}{DOWN}PRESS
    SPACE BAR{CYN}"                     :rem 56
97 IFPEEK(197)<>32THEN97                :rem 148
100 RETURN                              :rem 113
102 REM TIME AND SOUND FOR LOSE MAN ROUTI
    NE                                  :rem 30
103 N=N-2:IFN<0THENN=0                  :rem 57
104 POKEV,N:IFPEEK(E%)=38ORPEEK(E%)=39THE
    NFORQ=1TO200:NEXT                   :rem 146
105 FORQ=1TO50:NEXT:RETURN              :rem 215
110 IFPEEK(E%)<>34ANDPEEK(E%)<>35THENRETU
    RN                                  :rem 134
111 F=1:RETURN                          :rem 97
```

Program 2:

Bonking Barrels—64 Version

```
100 PRINT"{CLR}{CYN}{3 DOWN}{7 SPACES}RED
    EFINING CHARACTER SET"              :rem 27
110 GOTO390                             :rem 103
120 REM MAN JUMPS                       :rem 226
130 POKEE%,39                           :rem 197
140 FOR X=1TOZ%:GOSUB250:IF PEEK(B%(X))>3
    6ANDPEEK(B%(X))<40 THEN910          :rem 109
145 IF PEEK(B%(X)+D(X))>36ANDPEEK(B%(X)+D
    (X))<40THEN910                      :rem 72
147 IF PEEK(B%(X)-1)>36ANDPEEK(B%(X)-1)<4
    0THEN910                            :rem 214
150 GOSUB 310:NEXTX                     :rem 123
160 POKEE%,37                           :rem 198
170 POKES,P                             :rem 150
180 P=P+3                               :rem 212
190 POKEE%,38                           :rem 202
200 FORX=1TOZ%:GOSUB250:IF PEEK(B%(X))>36
    ANDPEEK(B%(X))<40 THEN910           :rem 106
203 IF PEEK(B%(X)+D(X))>36ANDPEEK(B%(X)+D
    (X))<40THEN910                      :rem 67
205 IF PEEK(B%(X)-1)>36ANDPEEK(B%(X)-1)<4
    0THEN910                            :rem 209
210 GOSUB 310:NEXTX                     :rem 120
220 IFPEEK(E%-40)=LLTHEN 910            :rem 50
230 POKEE%,32:E%=E%-40:POKEE%,38:POKES,0:
    POKEE%,32:E%=E%-40:POKEE%,38:RETURN
                                        :rem 225
240 REM MOVE BARRELS                    :rem 188
250 POKES+1,O                           :rem 240
260 POKEB%(X),C                         :rem 70
270 IFPEEK(B%(X)+I%)=WTHENB%(X)=B%(X)+D(X
    )                                   :rem 127
280 IFPEEK(B%(X))=WTHENB%(X)=B%(X)-T%*D(X
    )                                   :rem 140
290 IFPEEK(B%(X)+I%)<>WTHENB%(X)=B%(X)+I%
                                        :rem 63
300 RETURN                              :rem 115
```

```
310 POKEB%(X),LL                        :rem 151
320 IFB%(X)>JTHENGOSUB350               :rem 3
330 RETURN                              :rem 118
340 REM RESET BARRELS                   :rem 9
350 FORX=1TOZ%:POKEB%(X),C:B%(X)=SR+INT(R
    ND(X)*T%):NEXTX                     :rem 227
360 REM SCORE                           :rem 249
370 PRINT"{HOME}{23 DOWN}{RVS}SCORE
    {4 SPACES}{4 LEFT}"SC;"{9 SPACES}
    {6 LEFT}";                          :rem 5
380 FORQ=3TOA+1STEP-1:PRINT"{OFF}%";:NEXT
    :PRINT"{HOME}":RETURN               :rem 203
390 POKE53281,0:POKE53280,6             :rem 247
400 POKE56,48:CLR                       :rem 225
410 POKE56334,PEEK(56334)AND254:POKE1,PEE
    K(1)AND251                          :rem 182
420 BL$="{RVS}{YEL}{40 SPACES}"         :rem 121
430 AL$="{OFF}{PUR}$$$$$$$$$$$$$$$$$$$$
    $$$$$$$$$$$$$$$$$$"                 :rem 151
440 FORI=0TO1023:POKEI+12288,PEEK(I+53248
    ):POKEI+13312,PEEK(I+53248):NEXT
                                        :rem 8
450 POKE1,PEEK(1)OR4                    :rem 161
460 POKE56334,PEEK(56334)OR1            :rem 71
470 POKE53272,(PEEK(53272)AND240)+12
                                        :rem 186
480 FORI=12544TO12544+8*8-1:READA:POKEI,A
    :NEXT:GOSUB1080                     :rem 129
490 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    56,108,206,174,234,230,108,56       :rem 21
500 DATA56,108,246,250,134,222,108,56:REM
     BARREL                             :rem 34
510 DATA255,153,189,153,255,0,0,0:REM BEA
    M                                   :rem 179
520 DATA56,56,16,56,84,16,56,40:REM MAN N
    ORMAL                               :rem 242
530 DATA186,186,84,56,16,16,56,40,0,0,0,5
    6,56,16,124,170                     :rem 244
540 SR=1025:K=55296-SR+1                :rem 62
550 J=SR+820:V=54296:S=54272:FORI=STOV:PO
    KEI,0:NEXTI                         :rem 243
560 POKEV,15:POKES+5,130:POKES+6,72
                                        :rem 64
570 SR=1025:K=55296-SR+1                :rem 65
580 PRINT"{CLR}":A=0:SC=0:Z%=4          :rem 80
590 REM SET UP SCREEN                   :rem 211
600 PRINT"{CLR}";:BL=80                 :rem 162
610 FORI=1TO10:PRINTBL$;AL$;:NEXT:PRINTBL
    $;BL$;"{HOME}"                      :rem 74
620 X=SR+39:POKES+4,33:POKES+24,15
                                        :rem 245
630 FORB=1TOBL                          :rem 97
640 R=INT(RND(X)*908)                   :rem 189
650 POKEX+R,32:POKEX+R+K,7:POKES+1,B+10:N
    EXT:POKES+4,0                       :rem 73
660 POKES+4,17:FORG=-1TO20              :rem 25
670 POKEX+G*40,36:POKES+1,50+(G*2):POKEX+
    G*40+K,5                            :rem 25
680 POKEX+39+(G*40),36:POKES+1,0:POKEX+39
    +(G*40)+K,5:NEXTG:POKES+4,0         :rem 205
690 REM PREP FOR LOOP                   :rem 219
700 E%=J:D(1)=1:D(2)=-1:D(3)=1:D(4)=-1:L=
    1:LM=10:RM=18:UP=5:BM=4:O=0         :rem 5
710 W=36:C=32:KB=197:T%=38:I%=40:LL=34:D=
    37:Z=245:P=128:TT%=1:TH%=3:GOSUB350
                                        :rem 152
720 REM MAIN GAME LOOP                  :rem 246
730 FOR X=1TOZ%:GOSUB250:IFPEEK(B%(X))>36
    ANDPEEK(B%(X))<40 THEN910           :rem 114
735 IF PEEK(B%(X)+D(X))>36ANDPEEK(B%(X)+D
    (X))<40THEN910                      :rem 77
737 REM{3 SPACES}IF PEEK(B%(X)-1)>36ANDPE
    EK(B%(X)-1)<40THEN910               :rem 191
740 GOSUB 310:NEXTX                     :rem 128
750 IFPEEK(E%)=LLTHENE%=E%+40:POKEE%+K,7:
    POKEE%+I%,36:POKEE%+I%+K,4:GOTO910
                                        :rem 171
760 IFPEEK(E%-I%)=LLTHEN910             :rem 69
770 IFPEEK(E%+I%)=LLTHENPOKEE%,C:E%=E%+80
    :GOTO910                            :rem 157
780 IFLL=34THENLL=35:GOTO800            :rem 225
790 LL=34                               :rem 220
800 POKEE%,C                            :rem 160
810 IFPEEK(KB)=LMTHENIFPEEK(E%-L)<>WTHENE
    %=E%-L:POKEE%,D:YG=-2:GOTO850:rem 107
820 IFPEEK(KB)=RMTHENIFPEEK(E%+L)<>WTHENE
    %=E%+L:POKEE%,D:YG=2:GOTO850        :rem 65
830 IFPEEK(KB)=BMTHENIFPEEK(E%-I%)=WTHENP
    OKEE%-I%,C:SC=SC-5:P=P-10           :rem 66
840 IFPEEK(KB)=UPTHENIFPEEK(E%-I%)<>WTHEN
    GOSUB130:SC=SC+L+L:P=P+3:POKEE%,D
                                        :rem 166
850 IFP>ZTHENP=128                      :rem 107
860 POKEE%,D:POKES+4,17:POKES+1,40
                                        :rem 254
870 IF E%<SRTHENE%=J:TT%=TT%+L:GOSUB370:I
    FSC>35THENZ%=RND(X)*4+1             :rem 172
880 IFTT%>TH%THENGOTO600                :rem 249
890 GOTO730                             :rem 116
900 REM LOSE MAN                        :rem 140
910 POKES+4,129:N=15:POKES,39:POKES+1,09:
    FORQ=1TO4:POKEB%(Q),32:NEXT         :rem 162
920 POKEE%,38:POKEE%-40,34:GOSUB1190:POKE
    E%,37:GOSUB1190:POKEE%,39:GOSUB1190
                                        :rem 37
930 POKEE%-40,194:POKEE%,35:GOSUB1190:POK
    EE%+1,173:POKEE%-1,173:POKEE%+40,194
                                        :rem 150
940 POKEE%-39,206:POKEE%-41,205         :rem 187
950 POKEE%+41,205:POKEE%+39,206:GOSUB1190
                                        :rem 61
960 POKEE%-120,174:POKEE%-2,174:POKEE%+2,
    174:POKEE%+120,174:GOSUB1190        :rem 144
970 POKEE%-78,174                       :rem 157
980 POKEE%-82,174:POKEE%+78,174:POKEE%+82
    ,174:GOSUB1190                      :rem 131
990 SC=SC-8:POKES+4,128:A=A+1:GOSUB370
                                        :rem 251
1000 TT%=1:IFA<3GOTO600                 :rem 68
1010 PRINT"{CLR}{9 RIGHT}{9 DOWN}";
                                        :rem 255
1015 POKE198,0:PRINT"{RVS}{6 SPACES}GAME
     {SPACE}OVER"                       :rem 103
1018 PRINT"{DOWN}{RVS}{14 SPACES}SCORE= "
     ;SC                                :rem 72
1020 PRINT"{RVS}{DOWN}{6 SPACES}HIT SPACE
      BAR TO PLAY AGAIN"                :rem 22
1030 PRINT"{RVS}{DOWN}{10 SPACES}ANY OTHE
     R KEY TO END"                      :rem 133
1040 GETYY$:IFYY$=""THEN1040            :rem 143
1050 IF YY$<>" "THEN PRINT"{CLR}":END
                                        :rem 141
1060 GOTO580                            :rem 157
1070 REM                                :rem 172
1080 PRINT"{CLR}{3 DOWN}":PRINTTAB(11)"*I
     NSTRUCTIONS*":PRINT                :rem 197
1090 PRINTTAB(9)"{RVS}A{OFF} MOVES MAN LE
     FT":PRINTTAB(9)"{RVS}D{OFF} MOVES MA
     N RIGHT"                           :rem 170
1100 PRINT:PRINTTAB(9)"{RVS}F3{OFF} MOVES
      MAN UP 1 LEVEL"                   :rem 140
```

```
1110 PRINTTAB(9)"{RVS}F1{OFF} WILL BLAST
     {SPACE}SPACE ABOVE"          :rem 151
1120 PRINT:PRINTTAB(9)"{3 SPACES}**POINTS
     **"                          :rem 66
1130 PRINT:PRINTTAB(9)"2 FOR EACH UP":PRI
     NTTAB(8)"-5 FOR USING BLAST" :rem 61
1140 PRINTTAB(8)"-8 FOR GETTING HIT"
                                  :rem 58
1150 PRINT"{3 DOWN}":PRINTTAB(8)"
     {3 SPACES}PRESS SPACE BAR{CYN}"
                                  :rem 163
1160 GETXX$:IFXX$=""THEN1160      :rem 145
1170 RETURN                       :rem 169
1180 REM TIME AND SOUND FOR LOSE MAN ROUT
     INE                          :rem 85
1190 N=N-2:IFN<0THENN=0           :rem 112
1200 IFPEEK(E%)=38ORPEEK(E%)=39THENFORQ=1
     TO50:NEXT                    :rem 90
1210 FORQ=1TO25:NEXT:RETURN       :rem 7
```

Space Patrol

See article on page 52.

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:

Space Patrol—VIC Version, Loader Program

```
5 PRINT"{CLR}{7 DOWN}{2 SPACES}{RVS}** SP
  ACE PATROL **{8 DOWN}"          :rem 144
11 PRINT"PLEASE WAIT WHILE ":PRINT"PROGRA
   M LOADS..."                    :rem 187
15 POKE 52,27:POKE56,27           :rem 250
20 FORI=7168TO7679:POKEI,PEEK(I+25600):NE
   XT                             :rem 99
30 READX:IFX<0THEN45              :rem 252
35 FORI=XTOX+7:READJ:POKEI,J:NEXT:GOTO30
                                  :rem 161
45 S$="LO"+CHR$(34)+"SP"+CHR$(34)+",8:"+C
   HR$(131)                       :rem 117
50 FOR I=1 TO LEN(S$):POKE630+I,ASC(MID$(
   S$,I)):NEXT:POKE 198,I:END     :rem 92
800 PRINT"{HOME}":NEW:CLR:END     :rem 204
1000 DATA7384,0,0,0,63,95,255,0,0:rem 116
1001 DATA7392,2,6,14,254,254,255,240,60
                                  :rem 172
1002 DATA7400,64,96,112,127,127,255,15,60
                                  :rem 17
1003 DATA7408,0,0,0,252,250,255,0,0
                                  :rem 205
1004 DATA7416,0,0,60,126,171,126,60,0
                                  :rem 56
1005 DATA7432,0,16,84,16,254,16,84,16
                                  :rem 79
1006 DATA7440,0,146,16,56,254,56,16,146
                                  :rem 181
1007 DATA7448,64,96,112,95,64,255,19,62
                                  :rem 204
1008 DATA7456,0,0,0,248,12,255,0,0
                                  :rem 166
1009 DATA7464,0,0,0,31,48,255,0,0:rem 117
1010 DATA7472,2,6,14,250,2,255,200,124
                                  :rem 107
1011 DATA7480,0,0,0,0,0,126,0,0   :rem 249
1012 DATA7488,127,204,200,126,6,14,28,255
                                  :rem 21
1013 DATA7496,254,51,19,126,96,112,56,255
                                  :rem 39
1014 DATA7504,0,0,2,3,15,31,63,255
                                  :rem 164
1015 DATA7512,1,3,7,15,159,255,255,255
                                  :rem 132
1016 DATA7520,0,128,192,192,224,227,247,2
     55                           :rem 127
1017 DATA7528,0,4,14,63,255,255,255,255
                                  :rem 187
1018 DATA7536,4,6,15,191,255,255,255,255
                                  :rem 244
1019 DATA7544,0,48,242,255,255,255,255,25
     5                            :rem 89
1020 DATA7632,0,0,0,129,195,231,255,255
                                  :rem 168
1021 DATA7640,32,112,248,252,254,255,255,
     255                          :rem 174
1022 DATA7648,0,0,0,32,112,248,252,255
                                  :rem 116
1023 DATA7168,255,255,255,255,255,255,255
     ,255                         :rem 246
1024 DATA7656,0,4,14,14,14,4,14,10
                                  :rem 166
1025 DATA7664,0,0,0,0,0,0,0,0     :rem 153
1026 DATA6913,166,47,134,95,166,48,134,96
                                  :rem 52
1027 DATA6921,160,0,177,95,201,204,240,12
                                  :rem 12
1028 DATA6929,24,101,95,133,95,144,2,230
                                  :rem 234
1029 DATA6937,96,76,9,27,96,0,0,0:rem 156
1030 DATA6945,0,0,0,0,32,148,224,165
                                  :rem 12
1031 DATA6953,142,56,229,11,176,252,101,1
     1                            :rem 66
1032 DATA6961,133,11,96,0,24,32,148,209
                                  :rem 177
1033 DATA6969,160,0,177,88,208,58,162,22
                                  :rem 249
1034 DATA6977,134,93,162,30,134,94,166,14
     2                            :rem 88
1035 DATA6985,16,16,162,9,134,11,32,37
                                  :rem 136
1036 DATA6993,27,169,12,24,101,11,133,11
                                  :rem 222
1037 DATA7001,144,7,162,10,134,11,32,37
                                  :rem 160
1038 DATA7009,27,165,93,24,101,11,133,93
                                  :rem 228
1039 DATA7017,160,1,145,88,144,2,230,94
                                  :rem 181
1040 DATA7025,165,94,136,145,88,76,163,27
                                  :rem 42
1041 DATA7033,177,88,133,94,200,177,88,13
     3                            :rem 88
1042 DATA7041,93,160,22,177,93,201,32,208
                                  :rem 17
1043 DATA7049,41,169,32,160,0,145,93,165
                                  :rem 233
1044 DATA7057,93,24,105,22,133,93,200,145
                                  :rem 17
1045 DATA7065,88,136,144,6,230,94,165,94
                                  :rem 250
1046 DATA7073,145,88,169,31,145,93,165,94
                                  :rem 51
```

```
1047 DATA7081,24,105,120,133,94,169,7,145
                                        :rem 26
1048 DATA7089,93,96,48,253,169,32,160,0
                                        :rem 204
1049 DATA7097,145,93,169,0,145,88,200,145
                                        :rem 43
1050 DATA7105,88,141,0,27,76,178,27,0
                                        :rem 82
1051 DATA7113,0,32,1,27,160,1,177,95
                                        :rem 17
1052 DATA7121,201,128,240,6,32,9,27,76
                                        :rem 123
1053 DATA7129,205,27,173,0,27,133,11,32
                                        :rem 168
1054 DATA7137,37,27,230,11,165,11,141,0
                                        :rem 164
1055 DATA7145,27,32,53,27,173,0,27,240
                                        :rem 127
1056 DATA7153,7,133,11,230,11,32,53,27
                                        :rem 115
1057 DATA7161,96,0,0,0,0,0,255,255
                                        :rem 173
1058 DATA -1                            :rem 70
```

Program 2:

Space Patrol—VIC Main Program

```
30 C=30720:V=36878:S1=36875:S2=S1+1:S3=S1
   +2:HS=7664:HI=PEEK(HS)*256+PEEK(HS+1)
                                        :rem 223
40 DD=37154:DIML%(13):POKE36879,9:GOSUB70
   0:POKE36869,255                      :rem 216
50 PRINT"{CLR}{WHT}{RVS}SC=0{6 SPACES}HI=
   0{5 SPACES}#{HOME}"                  :rem 23
55 P=7976:D=0:DI=1:BA=20:OP=5:SC=0:BO=0:G
   =7723:C$="{HOME}{21 DOWN}"           :rem 59
60 FORI=38410TO38850STEP22:POKEI,1:POKEI+
   1,1:NEXT:FORI=38443TO38861STEP22:POKEI
   ,5:NEXT                              :rem 151
65 GOSUB380:GOSUB850:PRINTC$;A$;"{CYN}@@@
   @@@@@@@@@@@@@@@@{HOME}":POKEP,27:POK
   EP+1,28                              :rem 21
70 POKE38905,3:POKE8185,0:GOSUB850:FORT=1
   TO6:GOSUB900:NEXT                    :rem 120
75 FORH=1TO5-U                          :rem 99
76 POKE7420,204:POKE37139,0:POKEDD,127:J=
   PEEK(37152)AND128:J3=-(J=.):POKEDD,255
                                        :rem 53
90 J=PEEK(37137):J1=-((JAND8)=.):J2=-((JA
   ND16)=.):J0=-((JAND4)=.):FR=-((JAND32)
   =.)                                  :rem 228
95 IFFRANDDI=1THEN400                   :rem 38
99 POKE7420,179                         :rem 59
100 IFJ1THEN265                         :rem 103
105 IFJ2THEN220                         :rem 100
110 IFJ3THEN200                         :rem 95
115 IFJ0THEN255                         :rem 107
120 NEXT:GOSUB900:GOTO75                :rem 4
200 D=1:POKEP,32:POKEP+1,32:GOSUB300:A$=M
   ID$(A$,2,26)+MID$(A$,1,1):GOTO250
                                        :rem 166
220 D=0:POKEP,32:POKEP+1,32:GOSUB300:A$=M
   ID$(A$,27,1)+MID$(A$,1,26)           :rem 212
250 PRINTC$;A$"{HOME}":GOTO120          :rem 136
255 POKEP,32:POKEP+1,32:P=P-22:IFP<7712TH
   ENP=P+22                             :rem 187
260 GOSUB300:GOTO120                    :rem 177
265 POKEP,32:POKEP+1,32:P=P+22:IFP>8130TH
   ENP=P-22                             :rem 185
275 GOSUB300                            :rem 177
280 IF(P=8130)ANDPEEK(P+22)=40ANDPEEK(P+2
   3)=41ANDDI=0THENGOSUB350:DI=1:GOSUB30
   0                                    :rem 253
285 GOTO250                             :rem 111
300 IFD=.THEN315                        :rem 153
305 IFDI=.THENPOKEP,35:POKEP+1,36:RETURN
                                        :rem 37
310 POKEP,29:POKEP+1,30:RETURN          :rem 104
315 IFDI=.THENPOKEP,37:POKEP+1,38:RETURN
                                        :rem 42
320 POKEP,27:POKEP+1,28:RETURN          :rem 110
350 POKEV,15:GOSUB855:FORT=150TO250:POKES
   1,T:POKES2,T:NEXT                    :rem 225
370 BA=BA-6+U:IFBA<5THENBA=5            :rem 161
380 B=BA:G=8163-BA*22:FORT=GTO8141STEP22:
   POKET,61:NEXT                        :rem 62
385 POKEV,0:POKES1,0:POKES2,0:RETURN
                                        :rem 46
400 POKEV,15:POKES2,245:IFD=.THEN430
                                        :rem 158
410 F=P+2                               :rem 197
415 IFF=P+11THEN493                     :rem 88
420 IFPEEK(F)=31THEN460                 :rem 75
425 POKEF+C,1:POKEF,39:FORT=1TO1:NEXT:POK
   EF,32:F=F+1:GOTO415                  :rem 254
430 F=P-1                               :rem 200
435 IFF=P-11THEN493                     :rem 92
440 IFPEEK(F)=31THEN460                 :rem 77
445 POKEF+C,1:POKEF,39:FORT=1TO1:NEXT:POK
   EF,32:F=F-1:GOTO435                  :rem 4
460 X=0:FORT=0TO13:IFL%(T)=FTHENX=T:L%(T)
   =0                                   :rem 127
470 NEXT:POKES3,200:FORT=15TOOSTEP-5:POKE
   F+C,7:POKEF,33:POKEV,T:POKEF+C,2:POKE
   F,34:NEXT                            :rem 101
480 POKES3,0:POKEF,32:SC=SC+2+U*4:BO=BO+2
   +U*4                                 :rem 159
485 IFBO>500THEN:GOSUB500:OP=OP+1:BO=BO-5
   00:GOSUB855                          :rem 218
490 PRINT"{HOME}{WHT}{RVS}"TAB(3)SC
                                        :rem 137
493 B=B-1:IFB=0THENDI=0:GOSUB300        :rem 173
498 POKEG,32:G=G+22:POKEV,0:POKES2,0:GOTO
   120                                  :rem 171
500 POKEV,15:FORT=1TO28:PRINT"{HOME}
   {DOWN}{WHT}";MID$("{18 SPACES}[£''BO
   NUS ",T,22):POKES2,230               :rem 233
510 FORY=1TO85:NEXTY,T:POKEV,0:POKES2,0:R
   ETURN                                :rem 37
700 POKE36869,240:PRINT"{CLR}{5 DOWN}
   {2 RIGHT}{RVS}{YEL}** SPACE PATROL **
   "                                    :rem 91
710 PRINT"{GRN}{6 DOWN}{4 RIGHT}EASY
   {5 RIGHT}HARD"                       :rem 67
715 PRINT"{DOWN}{4 RIGHT}{RVS}{WHT}1{OFF}
   ..{RVS}2{OFF}..{RVS}3{OFF}..{RVS}4
   {OFF}..{RVS}5{OFF}"                  :rem 155
720 GETW$:IFW$=""THEN720                :rem 129
725 U=VAL(W$):IFU<1ORU>5THEN720         :rem 155
730 Q%=14-2*U:RETURN                    :rem 170
850 PRINT"{HOME}{WHT}{RVS}"TAB(13)HI
                                        :rem 181
855 A$="{CYN}/:;<{WHT}(){CYN}*+,-./:;<
   {WHT}(){CYN}*+,-."                   :rem 137
860 PRINT"{HOME}{WHT}{RVS}"TAB(20)OP:RETU
   RN                                   :rem 220
900 POKE6912,Q%:SYS7114:IFPEEK(6912)=0THE
   N950                                 :rem 141
910 RETURN                              :rem 122
950 OP=OP-1:GOSUB860                    :rem 207
960 POKEV,15:FORT=1TO15:POKES2,250:FORY=1
   TO5:NEXT:POKES2,0:NEXT:POKEV,0:rem 106
```

```
985 IFOP<>0THENRETURN                    :rem 141
990 POKEV,15:FORT=8TO15:POKE36879,25:FORY
    =200TO220+T*2:POKES2,Y:NEXT:POKE36879
    ,9:NEXT                              :rem 88
1000 POKES0,0:POKES2,0:PRINT"{HOME}
    {5 DOWN}"TAB(4)"{WHT}←←←←←←←←←←←←"
                                         :rem 4
1005 PRINTTAB(4)"← GAME OVER ←":PRINTTAB(
    4)"←←←←←←←←←←←←"                     :rem 65
1006 PRINT"{3 DOWN}"TAB(5)"{WHT}HIT ANY K
    EY"                                  :rem 39
1010 IFSC>HITHENHI=SC:GOSUB850           :rem 160
1020 POKE7420,204:FORY=1TO80:NEXT:POKE742
    0,51:FORY=1TO80:NEXT                 :rem 33
1025 GETW$:IFW$=""THEN1020               :rem 218
1040 POKEHS,INT(HI/256):POKEHS+1,(HI/256-
    INT(HI/256))*256                     :rem 27
1050 RUN                                 :rem 187
```

Program 3:

Space Patrol—64 Version

```
10 DIM L%(12)                            :rem 96
20 BA=20:SC=0:BI=20:BL=500               :rem 191
30 SO=32768:VO=53248:MF=49530:S=54272
                                         :rem 75
40 GOSUB1570                             :rem 177
50 OP=5:POKEMF+0,1                       :rem 42
60 IFPEEK(MF+6)=.THEN90                  :rem 98
70 POKEMF+6,0:IFDITHENPOKESO+1016,PEEK(SO
   +1016)-2:DI=0:GOTO90                  :rem 213
80 A$=RIGHT$(A$,1)+LEFT$(A$,49):POKE214,2
   2:PRINT:PRINT"{WHT}"LEFT$(A$,40);
                                         :rem 191
90 IFPEEK(MF+7)=.THEN120                 :rem 144
100 POKEMF+7,0:IFDI=.THENPOKESO+1016,PEEK
    (SO+1016)+2:DI=1:GOTO120             :rem 148
110 A$=RIGHT$(A$,49)+LEFT$(A$,1):POKE214,
    22:PRINT:PRINT"{WHT}"LEFT$(A$,40);
                                         :rem 233
120 IFPEEK(MF+4)=.THENPOKEMF+4,HF:SYS4938
    9                                    :rem 73
130 IFPEEK(MF+8)THENGOSUB940:POKEMF+8,0
                                         :rem 150
140 IFBA=.ANDPEEK(SO+939)=45ANDPEEK(56320
    )=125ANDPEEK(VO+1)=227THENGOSUB640
                                         :rem 252
150 IFPEEK(MF)THEN170                    :rem 245
160 POKEMF,1:OP=OP-1:POKESO+30+2*OP,30:PO
    KESO+31+2*OP,31:GOSUB430             :rem 184
170 GETR$:IFR$<>""THENGOSUB390:GOTO170
                                         :rem 65
180 IFOP>0THEN60                         :rem 202
190 IFSC>HITHENHI=SC                     :rem 33
200 H1=INT(HI/65536):POKE820,H1          :rem 245
210 HI=HI-H1*65536:H1=INT(HI/256):POKE821
    ,H1                                  :rem 253
220 H1=HI-H1*256:POKE822,H1              :rem 242
230 PRINT"{HOME}{DOWN}{YEL}":I$="{UP}↑←↑←
    ↑←↑←↑←"                              :rem 146
240 FORI=29TO0STEP-1:GOSUB430            :rem 44
250 IFPEEK(MF+4)THEN250                  :rem 84
260 POKEMF+4,5                           :rem 26
270 I$=I$+LEFT$(RIGHT$(I$,2),1)          :rem 150
280 PRINTSPC(I)I$;:NEXT                  :rem 200
290 FORI=1TO21:PRINT:PRINTI$;:GOSUB430
                                         :rem 198
300 IFPEEK(MF+4)THEN300                  :rem 76
310 POKEMF+4,10:NEXT                     :rem 187
320 PRINT"{HOME}{9 DOWN}{CYN}"SPC(8)"
    {3 SPACES}PLAY AGAIN?(Y OR N)
    {2 SPACES}"                          :rem 210
330 GETA$:IFA$=""THEN330                 :rem 79
340 IFA$="N"THEN360                      :rem 34
350 RUN                                  :rem 141
360 POKE56576,199:POKE53272,20:POKE648,4:
    POKEVO+21,0:END                      :rem 18
370 PRINT "DATA ERROR FOUND- CHECK FOR TY
    PO'S":GOTO360                        :rem 207
380 :{4 SPACES}PAUSE FUNCTION            :rem 185
390 IFASC(R$)<>133THENRETURN             :rem 233
400 GETR$:IFR$=""THEN400                 :rem 109
410 RETURN                               :rem 117
420 :{4 SPACES}SOUNDS                    :rem 172
430 POKES+15,20:POKES+17,4:POKES+19,9
                                         :rem 157
440 POKES+18,64:POKES+18,65:RETURN:rem 69
450 POKES+15,10:POKES+19,9               :rem 239
460 POKES+18,128:POKES+18,129:RETURN
                                         :rem 169
470 POKES+18,32:POKES+15,10              :rem 28
480 POKES+19,30                          :rem 66
490 POKES+18,33:RETURN                   :rem 95
500 POKES+15,35:POKES+19,1:FORJ=1TO5:NEXT
                                         :rem 20
510 POKES+18,32:POKES+18,33:RETURN:rem 57
520 :{4 SPACES}SET DIFFICULTY LEVEL
                                         :rem 40
530 PRINT "{HOME}{4 DOWN}{6 RIGHT}SELECT
    {SPACE}DIFFICULTY LEVEL(1TO5)"
                                         :rem 243
540 GETR$:IFR$=""THEN540                 :rem 119
550 IF ASC(R$)<49 THEN 540               :rem 56
560 IF ASC(R$)>53 THEN 540               :rem 54
570 DL=ASC(R$)-48                        :rem 160
580 HF=20-4*DL                           :rem 229
590 NL=13-DL                             :rem 150
600 POKEMF+1,NL                          :rem 122
610 SI=1+2*DL+INT(DL↑2/2)                :rem 160
620 PRINT "{UP}{37 SPACES}":RETURN:rem 20
630 :{5 SPACES}GET MORE BOMBS            :rem 89
640 BI=BI-2-INT(DL/2):IFBI<5THENBI=5
                                         :rem 151
650 BA=BI:HF=HF-1:IFHF<0THENHF=0         :rem 196
660 FORI=SO+919TOSO+959-BA*40STEP-40:POKE
    I,27:GOSUB500:NEXT                   :rem 170
670 POKESO+1016,PEEK(SO+1016)-1          :rem 246
680 RETURN                               :rem 126
690 :{4 SPACES}BONUS                     :rem 96
700 BL=300:POKEVO+3,250                  :rem 36
710 GOSUB470:K=10                        :rem 214
720 FORI=250TO50STEP-1:POKEVO+5,I:POKES+1
    5,K:K=K+1:NEXT                       :rem 200
730 IFOP=5THENSC=SC+SI:BO=BO+SI:GOTO870
                                         :rem 11
740 POKEVO+2,241:POKEVO+3,53             :rem 143
750 T=SO+30-2*OP                         :rem 133
760 I=241:II=0:POKEVO+16,PEEK(VO+16)AND25
    3                                    :rem 217
770 GOSUB450:K=10                        :rem 218
780 IFI=257THENII=256:POKEVO+16,PEEK(VO+1
    6)OR2                                :rem 57
790 POKEVO+2,I-II:IFI>266+16*OPTHEN820
                                         :rem 23
800 K=K+8:IFK<40THENPOKES+15,K           :rem 59
810 I=I+8:GOTO780                        :rem 221
820 POKEVO+2,250:SL=SO+30+2*OP:OP=OP+1
                                         :rem 11
830 POKESL,28:POKESL+1,29                :rem 250
840 GOSUB450:FORK=10TO18STEP4:POKES+15,K:
    NEXTK                                :rem 195
850 POKEVO+2,0:POKEVO+3,0                :rem 242
860 POKESL,32:POKESL+1,32                :rem 242
870 SC=SC+SI:BO=BO+SI                    :rem 47
```

```
880 SS$=STR$(SC):PRINT"{HOME}{4}"SPC(31-L
    EN(SS$))RIGHT$(SS$,LEN(SS$)-1):rem 99
890 FORI=1TO200:NEXT                :rem 237
900 GOSUB470:K=211                  :rem 10
910 FORI=50TO250:POKEVO+5,I:POKES+15,K:K=
    K-1:NEXT                        :rem 49
920 POKES+15,0:POKEVO+16,0:RETURN   :rem 32
930 :{4 SPACES}TRACK SHOT           :rem 137
940 IFBA=.THENRETURN                :rem 41
950 RW=32768+INT((PEEK(VO+1)-46)/8)*40
                                    :rem 73
960 POKEVO+3,PEEK(VO+1)+3:PX=173+DI*28
                                    :rem 216
970 GOSUB450:K=10:IFDITHEN1040      :rem 38
980 FORI=RW+18TORWSTEP-2:IFPEEK(I)=31THEN
    SL=I-1:I=RW-2                   :rem 213
990 IFPEEK(I)=30THENSL=I:I=RW-2     :rem 172
1000 POKEVO+2,PX:PX=PX-16:IFK<40THENK=K+4
    :POKES+15,K                     :rem 250
1010 NEXT                           :rem 1
1020 IFI=RW-2THEN1180               :rem 217
1030 GOTO1100                       :rem 191
1040 IX=0:FORI=RW+21TORW+37STEP2:IFPEEK(I
    )=30THENSL=I:I=RW+39            :rem 124
1050 IFPEEK(I)=31THENSL=I-1:I=RW+39
                                    :rem 103
1060 POKEVO+2,PX-IX:PX=PX+16:IFPX=265THEN
    IX=256:POKEVO+16,2              :rem 230
1070 IFK<40THENK=K+4:POKES+15,K     :rem 103
1080 NEXT                           :rem 8
1090 IFI=RW+39THEN1180              :rem 24
1100 POKESL,28:POKESL+1,29:GOSUB450
                                    :rem 116
1110 TL=SL-65536:FORI=0TONL:IFL%(I)=TLTHE
    NL%(I)=0:I=NL+1                 :rem 151
1120 NEXT                           :rem 3
1130 FORK=10TO18STEP4:POKES+15,K:NEXT
                                    :rem 78
1140 SC=SC+SI:SS$=STR$(SC):PRINT"{HOME}
    {4}"SPC(31-LEN(SS$))RIGHT$(SS$,LEN(S
    S$)-1)                          :rem 243
1150 POKESL,32:POKESL+1,32          :rem 27
1160 BO=BO+SI                       :rem 238
1170 IFBO>BLTHENBO=BO-BL:GOSUB700   :rem 79
1180 POKEVO+2,0:POKEVO+3,0:POKEVO+16,0
                                    :rem 27
1190 POKE(SO+159+(20-BA)*40),32:BA=BA-1:I
    FBA=.THENPOKESO+1016,PEEK(SO+1016)+1
                                    :rem 124
1200 RETURN                         :rem 163
1210 :{4 SPACES}INITIALIZE SCREEN:rem 176
1220 A$=",,,,,,,,,,-./,,,,,,,,,,,,,,,-./
    ,,,,,,,,,,,,,-./,,"             :rem 85
1230 B$="{GRN}++++++++++++++++++++++++++
    ++++++++++"                     :rem 88
1240 FORI=STOS+23:POKEI,0:NEXT      :rem 148
1250 POKES+23,0:POKES+22,18         :rem 22
1260 POKES,32:POKES+1,2:POKES+24,31:POKES
    +3,14                           :rem 218
1270 POKES+5,40:POKES+6,0           :rem 181
1280 POKEVO+33,0:POKEVO+32,12       :rem 185
1290 PRINT"{2 CLR}{4} HIGH:00000"SPC(8)"S
    CORE:00000"                     :rem 253
1300 SS$=STR$(HI):PRINT"{HOME}{4}"SPC(12-
    LEN(SS$))RIGHT$(SS$,LEN(SS$)-1)
                                    :rem 129
1310 POKE214,22:PRINT:PRINT"{WHT}"LEFT$(A
    $,40)B$;                        :rem 40
1320 SYS49420:POKEVO+39,1:POKEVO+40,1
                                    :rem 196
1330 POKEVO+41,1:POKEVO+21,7:POKEMF+8,0
                                    :rem 66
1340 POKESO+999,43:POKESO+23527,5:rem 156
1350 FORI=SO+159TOSO+919STEP40:POKEI,27
                                    :rem 156
1360 POKEI+22528,2:NEXT             :rem 71
1370 POKESO+1016,16:POKEVO+1,100:POKEVO,1
    76:POKEVO+16,0:DI=0             :rem 81
1380 POKESO+1017,20:POKESO+1018,18
                                    :rem 180
1390 POKEVO+2,0:POKEVO+3,0          :rem 34
1400 POKEVO+5,250:POKEVO+4,216      :rem 238
1410 FORI=55326TO55335:POKEI,7:NEXT
                                    :rem 110
1420 RETURN                         :rem 167
1430 :{4 SPACES}LOAD DATA           :rem 60
1440 PRINT"{CLR}{DOWN}{7 SPACES}LOADING D
    ATA{3 SPACES}PLEASE WAIT"       :rem 69
1450 POKE56334,PEEK(56334)AND254    :rem 19
1460 POKE1,PEEK(1)AND251            :rem 105
1470 FORI=34816TO35327:POKEI,PEEK(I+18432
    ):NEXT                          :rem 41
1480 POKE1,PEEK(1)OR4               :rem 213
1490 POKE56334,PEEK(56334)OR1       :rem 123
1500 FORI=SO+1024TOSO+1343:POKEI,0:NEXT:P
    OKESO+1280,255                  :rem 82
1510 READL:IFL=0THENGOTO1530        :rem 226
1520 FORI=0TO7:READA:POKEL+I,A:CK=CK+A:NE
    XT:GOTO1510                     :rem 30
1530 IFCK<>63389THEN370             :rem 54
1540 POKE251,111                    :rem 80
1550 RETURN                         :rem 171
1560 ;{4 SPACES}MOVE SCREEN TO TOP OF BAS
    IC                              :rem 139
1570 POKE648,128                    :rem 101
1580 POKE56576,PEEK(56576)AND252OR1
                                    :rem 247
1590 POKEVO+24,2                    :rem 146
1600 IFPEEK(251)<>111THENGOSUB1440
                                    :rem 233
1610 POKEVO+32,11:POKEVO+33,11      :rem 231
1620 POKE56,127:POKE52,127          :rem 191
1630 HI=PEEK(820)*65536+PEEK(821)*256+PEE
    K(822)                          :rem 27
1640 GOSUB 1220:GOSUB530            :rem 98
1650 RETURN                         :rem 172
1660 :{3 SPACES}CHARACTER DATA      :rem 174
1670 DATA35032,0,0,0,125,255,125,0,0
                                    :rem 1
1680 DATA35040,16,70,185,134,93,82,195,16
                                    :rem 37
1690 DATA35048,8,82,131,212,43,117,66,4
                                    :rem 186
1700 DATA35056,0,63,96,255,204,127,0,0
                                    :rem 121
1710 DATA35064,0,252,6,255,51,254,0,0
                                    :rem 65
1720 DATA35160,255,255,255,255,255,255,25
    5,255                           :rem 35
1730 DATA35168,0,0,0,0,0,0,0,0      :rem 204
1740 DATA35176,255,63,15,15,15,15,15,15
                                    :rem 181
1750 DATA35184,255,255,255,195,0,0,0,0
                                    :rem 127
1760 DATA35192,255,252,240,240,240,240,24
    0,240                           :rem 5
1770 :{3 SPACES}SPRITE DATA         :rem 250
1780 DATA33792,7,255,254,30,0,7,127,254
                                    :rem 191
1790 DATA33800,63,0,0,63,127,254,63,0
                                    :rem 73
1800 DATA33808,240,63,0,63,254,0,0,0
                                    :rem 12
```

```
1810 DATA33856,7,255,254,30,0,7,127,254
                                        :rem 186
1820 DATA33864,1,0,0,1,127,254,1,0
                                        :rem 165
1830 DATA33872,240,1,0,63,254,0,0,0
                                        :rem 216
1840 DATA33920,127,255,224,224,0,120,128,
     127                                :rem 166
1850 DATA33928,254,252,0,0,252,127,254,25
     2                                  :rem 76
1860 DATA33936,15,0,127,252,0,0,0,0
                                        :rem 218
1870 DATA33984,127,255,224,224,0,120,128,
     127                                :rem 179
1880 DATA33992,254,128,0,0,128,127,254,12
     8                                  :rem 86
1890 DATA34000,15,0,127,252,0,0,0,0
                                        :rem 204
1900 :{3 SPACES}ML ROUTINES             :rem 22
1910 DATA49152,166,47,134,95,166,48,134,9
     6                                  :rem 104
1920 DATA49160,160,0,177,95,201,204,240,1
     6                                  :rem 68
1930 DATA49168,160,2,177,95,24,101,95,133
                                        :rem 38
1940 DATA49176,95,144,0,230,96,76,8,192
                                        :rem 207
1950 DATA49184,96,165,143,162,3,10,202,20
     8                                  :rem 81
1960 DATA49192,252,24,101,143,10,24,101,1
     43                                 :rem 109
1970 DATA49200,24,105,59,133,143,74,74,56
                                        :rem 36
1980 DATA49208,229,11,176,252,101,11,133,
     11                                 :rem 118
1990 DATA49216,96,24,32,148,177,160,0,177
                                        :rem 47
2000 DATA49224,88,208,63,162,40,134,93,16
     2                                  :rem 78
2010 DATA49232,128,134,94,162,34,134,11,3
     2                                  :rem 64
2020 DATA49240,33,192,165,11,201,18,48,7
                                        :rem 224
2030 DATA49248,201,21,16,3,24,105,17,101
                                        :rem 208
2040 DATA49256,93,133,93,144,2,230,94,160
                                        :rem 29
2050 DATA49264,0,177,93,201,32,208,93,200
                                        :rem 21
2060 DATA49272,177,93,201,32,208,86,165,9
     3                                  :rem 93
2070 DATA49280,145,88,165,94,136,145,88,7
     6                                  :rem 110
2080 DATA49288,189,192,177,88,133,94,200,
     177                                :rem 213
2090 DATA49296,88,133,93,160,40,177,93,20
     1                                  :rem 95
2100 DATA49304,32,208,58,200,177,93,201,3
     2                                  :rem 66
2110 DATA49312,208,51,169,32,160,0,145,93
                                        :rem 19
2120 DATA49320,200,145,93,165,93,24,105,4
     0                                  :rem 65
2130 DATA49328,133,93,145,88,136,144,6,23
     0                                  :rem 85
2140 DATA49336,94,165,94,145,88,169,30,14
     5                                  :rem 103
2150 DATA49344,93,200,169,31,145,93,165,9
     4                                  :rem 91
2160 DATA49352,24,105,88,133,94,169,7,145
                                        :rem 43
2170 DATA49360,93,136,145,93,96,48,253,16
     9                                  :rem 107
2180 DATA49368,32,160,0,145,93,200,145,93
                                        :rem 29
2190 DATA49376,136,152,145,88,200,145,88,
     141                                :rem 193
2200 DATA49384,122,193,76,212,192,32,0,19
     2                                  :rem 75
2210 DATA49392,160,1,177,95,201,128,240,6
                                        :rem 25
2220 DATA49400,32,8,192,76,240,192,173,12
     3                                  :rem 74
2230 DATA49408,193,133,11,32,33,192,230,1
     1                                  :rem 62
2240 DATA49416,32,65,192,96,120,162,25,14
     2                                  :rem 79
2250 DATA49424,20,3,162,193,142,21,3,88
                                        :rem 178
2260 DATA49432,96,72,138,72,162,3,189,124
                                        :rem 44
2270 DATA49440,193,240,3,222,124,193,202,
     16                                 :rem 117
2280 DATA49448,245,173,0,220,74,176,11,17
     4                                  :rem 84
2290 DATA49456,1,208,224,59,144,4,202,142
                                        :rem 28
2300 DATA49464,1,208,74,176,11,174,1,208
                                        :rem 231
2310 DATA49472,224,227,176,4,232,142,1,20
     8                                  :rem 72
2320 DATA49480,74,176,3,238,128,193,74,17
     6                                  :rem 100
2330 DATA49488,3,238,129,193,74,176,3,238
                                        :rem 53
2340 DATA49496,130,193,173,127,193,208,21
     ,169                               :rem 243
2350 DATA49504,12,141,127,193,169,63,77,2
     44                                 :rem 139
2360 DATA49512,136,141,244,136,169,252,77
     ,252                               :rem 238
2370 DATA49520,136,141,252,136,104,170,10
     4,76                               :rem 220
2380 DATA49528,49,234,0,0,0,0,0,0:rem 121
2390 DATA0                              :rem 24
```

The Beginner's Corner

See article on page 83.

Program 1: Quilt Squares For VIC

```
1 REM QUILT SQUARES VIC                 :rem 170
2 GOTO15                                :rem 209
3 POKE198,0:POKEV,231:FORD=1TO50:NEXT:POK
  EV,0:RETURN                           :rem 106
4 POKEA,32:POKEA+1,32:POKEA+22,32:POKEA+2
  3,32:P=1:RETURN                       :rem 115
5 POKEA,160:POKEA+1,160:POKEA+22,160:POKE
  A+23,160:RETURN                       :rem 68
6 POKEA,32:POKEA+1,233:POKEA+22,233:POKEA
  +23,160:RETURN                        :rem 21
7 POKEA,160:POKEA+1,105:POKEA+22,105:POKE
  A+23,32:RETURN                        :rem 18
8 POKEA,223:POKEA+1,32:POKEA+22,160:POKEA
  +23,223:RETURN                        :rem 21
```

```
9 POKEA,95:POKEA+1,160:POKEA+22,32:POKEA+
  23,95:RETURN                          :rem 196
10 A1=PEEK(A):A2=PEEK(A+1):A3=PEEK(A+22):
   A4=PEEK(A+23):RETURN                 :rem 66
11 POKEA,A1:POKEA+1,A2:POKEA+22,A3:POKEA+
   23,A4:RETURN                         :rem 227
12 POKEA,79:POKEA+1,80:POKEA+22,76:POKEA+
   23,122:RETURN                        :rem 240
13 POKEA+M,P:POKEA+1+M,P:POKEA+22+M,P:POK
   EA+23+M,P:RETURN                     :rem 55
14 FORI=8054TO8118:POKEI,32:NEXTI:RETURN
                                        :rem 55
15 PRINT"{CLR}{DOWN}{4 RIGHT}QUILT SQUARE
   S"                                   :rem 2
16 DIMS(16),Q(16),R(16),QQ(3)           :rem 9
17 PRINT"{DOWN}USE F1 TO MOVE TO THE QUIL
   T SQUARE DESIRED, THEN PRESS <RETURN>.
   "                                    :rem 220
18 PRINT"{DOWN}FILL IN THE SAMPLE
   {4 SPACES}SQUARES."                  :rem 254
19 FORI=1TO16:READS(I),Q(I),R(I):NEXT
                                        :rem 253
20 DATA7864,7703,79,7866,7705,160,7868,77
   07,233,7870,7751,105                 :rem 73
21 DATA7908,7709,223,7910,7753,95,7912,77
   11,160,7914,7713,233                 :rem 53
22 DATA7952,7757,105,7954,7715,223,7956,7
   759,95,7958,7717,160                 :rem 92
23 DATA7996,7719,233,7998,7763,105,8000,7
   721,223,8002,7765,95                 :rem 65
24 QQ(1)=6:QQ(2)=11:QQ(3)=16:POKE36878,15
   :V=36876:M=30720                     :rem 165
25 A$="OPOPOPOP":B$="L@L@L@L@"          :rem 2
26 PRINT"{DOWN}ONE COLOR IS WHITE.
   {3 SPACES}HOW MANY OTHER COLORS, 1, 2,
    OR 3?":GOSUB3                       :rem 84
27 GETE$:IFE$=""THEN27                  :rem 253
28 IFE$<"1"ORE$>"3"THEN27               :rem 106
29 N=VAL(E$)                            :rem 147
30 FORI=1TON                            :rem 242
31 PRINT"{2 DOWN}CHOOSE COLOR";I        :rem 27
32 PRINT" {RVS}{BLK} {OFF}{2 SPACES}{RVS}
   {RED} {OFF}{2 SPACES}{RVS}{CYN} {OFF}
   {2 SPACES}{RVS}{PUR} {OFF}{2 SPACES}
   {RVS}{GRN} {OFF}{2 SPACES}{RVS}{BLU}
   {OFF}{2 SPACES}{RVS}{YEL} {OFF}{BLU}"
                                        :rem 147
33 PRINT" 1{2 SPACES}2{2 SPACES}3
   {2 SPACES}4{2 SPACES}5{2 SPACES}6
   {2 SPACES}7":GOSUB3                  :rem 144
34 GETE$:IFE$=""THEN34                  :rem 249
35 IFE$<"1"ORE$>"7"THEN34               :rem 106
36 C(I)=VAL(E$):PRINTTAB(3*C(I)-2)"↑"
                                        :rem 74
37 IFC(I)=1THENC(I)=0                   :rem 189
38 NEXTI                                :rem 243
39 PRINT"{CLR}{7 DOWN}"                 :rem 71
40 FORI=1TO4:PRINTTAB(8)A$:PRINTTAB(8)B$:
   NEXTI                                :rem 180
41 POKE7703,79:POKE7704,101:POKE7725,99:P
   OKE7703+M,0:POKE7704+M,0:POKE7725+M,0
                                        :rem 5
42 FORI=2TOQQ(N):POKEQ(I),R(I):NEXTI
                                        :rem 23
43 FORI=2TO6:POKEQ(I)+M,C(1):NEXTI:rem 94
44 FORI=7TO11:POKEQ(I)+M,C(2):NEXTI
                                        :rem 145
45 FORI=12TO16:POKEQ(I)+M,C(3):NEXTI
                                        :rem 196
46 FORT=1TO16:GOSUB3:POKES(T),63    :rem 0
47 FORI=1TOQQ(N):GOSUB3:P=PEEK(Q(I)+M):IF
   I=1THENP=1                           :rem 231
48 POKEQ(I),32:POKEQ(I),R(I)            :rem 131
49 GETE$:IFE$=""THEN48                  :rem 4
50 IFASC(E$)=13THEN53                   :rem 189
51 IFE$<>"{F1}"THEN48                   :rem 60
52 NEXTI:GOTO47                         :rem 205
53 A=S(T):ONI GOSUB4,5,6,7,8,9,5,6,7,8,9,
   5,6,7,8,9                            :rem 127
54 GOSUB13:NEXTT                        :rem 26
55 PRINT"{2 DOWN}PRESS F1--CHANGE":PRINTT
   AB(6)"F7--PRINT QUILT":GOSUB3        :rem 170
56 GETE$:IFE$="{F7}"THEN74              :rem 137
57 IFE$<>"{F1}"THEN56                   :rem 65
58 GOSUB14                              :rem 82
59 PRINT"{2 UP}PRESS F7--NO CHANGE":PRINT
   TAB(6)"F1--CHANGE":GOSUB3            :rem 213
60 FORT=1TO16:A=S(T):GOSUB10:GOSUB3:P=PEE
   K(A+M):IFP=1THENP=0:GOSUB13          :rem 88
61 GOSUB12:GOSUB11                      :rem 102
62 GETE$:IFE$="{F7}"THENGOSUB11:GOTO73
                                        :rem 218
63 IFE$<>"{F1}"THEN61                   :rem 58
64 GOSUB12                              :rem 77
65 FORI=1TOQQ(N):GOSUB3:P=PEEK(Q(I)+M)
                                        :rem 122
66 POKEQ(I),32:POKEQ(I),R(I)            :rem 131
67 GETE$:IFE$=""THEN66                  :rem 4
68 IFASC(E$)=13THEN71                   :rem 198
69 IFE$<>"{F1}"THEN66                   :rem 69
70 NEXTI:GOTO65                         :rem 205
71 A=S(T):ONI GOSUB4,5,6,7,8,9,5,6,7,8,9,
   5,6,7,8,9                            :rem 127
72 GOSUB13                              :rem 77
73 NEXTT:GOSUB14:PRINT"{5 UP}":GOTO55
                                        :rem 217
74 FORT=1TO16:A=S(T):P=PEEK(A+M):GOSUB10
                                        :rem 223
75 A=S(T)-184:GOSUB11:GOSUB13:A=S(T)-176:
   GOSUB11:GOSUB13                      :rem 155
76 A=S(T)-8:GOSUB11:GOSUB13             :rem 130
77 NEXTT                                :rem 1
78 FORT=1TO12:A=S(T):P=PEEK(A+M):GOSUB10
                                        :rem 223
79 A=S(T)+168:GOSUB11:GOSUB13:A=S(T)+176:
   GOSUB11:GOSUB13                      :rem 157
80 NEXTT                                :rem 251
81 FORT=1TO16:A=S(T):P=PEEK(A+M):GOSUB10
                                        :rem 221
82 IFT/4=INT(T/4)THEN86                 :rem 125
83 A=S(T)-168:GOSUB11:GOSUB13:A=S(T)+8:GO
   SUB11:GOSUB13                        :rem 52
84 A=S(T)+184:IFT>12THEN86              :rem 5
85 GOSUB11:GOSUB13                      :rem 109
86 NEXTT                                :rem 1
87 PRINT"{HOME}{19 DOWN}"               :rem 150
88 END                                  :rem 71
```

Program 2: Quilt Squares For 64

Note: Before typing in or loading the program, enter the following line to clear memory for custom characters.

```
POKE 8192,0: POKE 44,32: NEW
```

```
10 REM QUILT SQUARES                    :rem 248
20 GOTO 310                             :rem 47
30 POKE A,103:POKE A+1,103:POKE A+40,103:
   POKE A+41,103:CC=C(1):RETURN         :rem 40
40 POKE A,103:POKE A+1,100:POKE A+40,100:
   POKE A+41,96:CC=C(1):RETURN          :rem 254
```

```
50 POKE A,102:POKE A+1,103:POKE A+40,96:P
   OKE A+41,102:CC=C(1):RETURN      :rem 3
60 POKE A,101:POKE A+1,96:POKE A+40,103:P
   OKE A+41,101:CC=C(1):RETURN      :rem 2
70 POKE A,96:POKE A+1,99:POKE A+40,99:POK
   E A+41,103:CC=C(1):RETURN        :rem 195
80 POKE A,96:POKE A+1,96:POKE A+40,96:POK
   E A+41,96:CC=C(1):RETURN         :rem 153
90 POKE A,160:POKE A+1,160:POKE A+40,160:
   POKE A+41,160:CC=C(1):RETURN     :rem 58
100 POKE A,160:POKE A+1,163:POKE A+40,163
    :POKE A+41,103:CC=C(1):RETURN:rem 101
110 POKE A,160:POKE A+1,163:POKE A+40,163
    :POKE A+41,103:CC=C(2):RETURN:rem 103
120 POKE A,165:POKE A+1,160:POKE A+40,167
    :POKE A+41,165:CC=C(1):RETURN:rem 117
130 POKE A,167:POKE A+1,164:POKE A+40,164
    :POKE A+41,160:CC=C(2):RETURN:rem 117
140 POKE A,167:POKE A+1,164:POKE A+40,164
    :POKE A+41,160:CC=C(1):RETURN:rem 117
150 POKE A,166:POKE A+1,167:POKE A+40,160
    :POKE A+41,166:CC=C(1):RETURN:rem 122
160 POKE A,166:POKE A+1,167:POKE A+40,160
    :POKE A+41,166:CC=C(2):RETURN:rem 124
170 POKE A,165:POKE A+1,160:POKE A+40,167
    :POKE A+41,165:CC=C(2):RETURN:rem 123
180 POKE A,224:POKE A+1,224:POKE A+40,224
    :POKE A+41,224:CC=C(1):RETURN:rem 110
190 POKE A,224:POKE A+1,227:POKE A+40,227
    :POKE A+41,231:CC=C(1):RETURN:rem 115
200 POKE A,224:POKE A+1,227:POKE A+40,227
    :POKE A+41,231:CC=C(2):RETURN:rem 108
210 POKE A,224:POKE A+1,227:POKE A+40,227
    :POKE A+41,231:CC=C(3):RETURN:rem 110
220 POKE A,229:POKE A+1,224:POKE A+40,231
    :POKE A+41,229:CC=C(1):RETURN:rem 113
230 POKE A,229:POKE A+1,224:POKE A+40,231
    :POKE A+41,229:CC=C(2):RETURN:rem 115
240 POKE A,229:POKE A+1,224:POKE A+40,231
    :POKE A+41,229:CC=C(3):RETURN:rem 117
250 POKE A,231:POKE A+1,228:POKE A+40,228
    :POKE A+41,224:CC=C(3):RETURN:rem 116
260 POKE A,231:POKE A+1,228:POKE A+40,228
    :POKE A+41,224:CC=C(2):RETURN:rem 116
270 POKE A,231:POKE A+1,228:POKE A+40,228
    :POKE A+41,224:CC=C(1):RETURN:rem 116
280 POKE A,230:POKE A+1,231:POKE A+40,224
    :POKE A+41,230:CC=C(1):RETURN:rem 103
290 POKE A,230:POKE A+1,231:POKE A+40,224
    :POKE A+41,230:CC=C(2):RETURN:rem 105
300 POKE A,230:POKE A+1,231:POKE A+40,224
    :POKE A+41,230:CC=C(3):RETURN :rem 98
310 PRINT "{CLR}":PRINT "TRANSFERRING CHA
    RACTER SET"                      :rem 17
320 PRINT "PLEASE WAIT..."           :rem 223
330 REM TRANSFER CHARACTER SET TO RAM
                                     :rem 219
340 POKE 56334,0:POKE 1,51           :rem 83
350 FOR C=2048 TO 6143               :rem 62
360 POKE C,PEEK(C+51200)             :rem 19
370 NEXT C                           :rem 28
380 POKE 1,55:POKE 56334,129:POKE 53272,1
    9                                :rem 201
390 REM                              :rem 128
400 POKE 53281,15                    :rem 88
410 PRINT "{CLR}{4}":PRINT TAB(13)"QUIL
    T SQUARES"                       :rem 218
420 DIM S(16),Q(28),R(28)            :rem 236
430 PRINT "{DOWN}USE F1 TO MOVE TO THE QU
    ILT SQUARE"                      :rem 155
440 PRINT "DESIRED, THEN PRESS <RETURN>."
                                     :rem 217
450 PRINT "{DOWN}FILL IN THE SAMPLE SQUAR
    ES."                             :rem 46
460 REM REDEFINE CHARACTERS          :rem 160
470 FOR C=2048+8*35 TO 2048+8*42-1
                                     :rem 135
480 READ G:POKE C,G                  :rem 30
490 NEXT C                           :rem 31
500 FOR C=2048+27*8 TO 2048+30*8-1:READ G
    :POKE C,G:NEXT C                 :rem 247
510 DATA 0,1,3,7,15,31,63,127        :rem 124
520 DATA 255,254,252,248,240,224,192,128
                                     :rem 184
530 DATA 128,192,224,240,248,252,254,255
                                     :rem 185
540 DATA 127,63,31,15,7,3,1,0        :rem 127
550 DATA 255,255,255,255,255,255,255,255
                                     :rem 200
560 DATA 255,128,128,128,128,128,128,128
                                     :rem 194
570 DATA 255,1,1,1,1,1,1,1           :rem 221
580 DATA 128,128,128,128,128,128,128,255
                                     :rem 196
590 DATA 255,129,129,129,129,129,129,255
                                     :rem 204
600 DATA 1,1,1,1,1,1,1,255           :rem 215
610 REM SOUND                         :rem 4
620 FOR C=54272 TO 54296:POKE C,0:NEXT
                                     :rem 49
630 POKE 54296,15:POKE 54277,190:POKE 542
    78,248                           :rem 217
640 HF=54273:LF=54272:W=54276         :rem 77
650 A$="()()()()"                    :rem 197
660 B$="[][][][]"                     :rem 99
670 FOR I=1 TO 16:READ S(I):NEXT I:rem 74
680 DATA 1352,1354,1356,1358,1432,1434,14
    36,1438                          :rem 88
690 DATA 1512,1514,1516,1518,1592,1594,15
    96,1598                          :rem 109
700 FOR I=1 TO 28:READ Q(I),R(I):NEXT I
                                     :rem 93
710 DATA 1142,103,1262,100,1382,102,1379,
    101,1259,99,1139,96              :rem 149
720 DATA 1136,160,1256,163,1376,163,1496,
    165,1499,164,1502,164             :rem 17
730 DATA 1622,166,1619,166,1616,165
                                     :rem 208
740 DATA 1133,224,1253,227,1373,227,1493,
    227,1613,229,1733,229,1853,229
                                     :rem 205
750 DATA 1856,228,1859,228,1862,228,1742,
    230,1739,230,1736,230            :rem 28
760 QQ(2)=6:QQ(3)=15:QQ(4)=28        :rem 64
770 PRINT "{2 DOWN}CHOOSE NUMBER OF COLOR
    S:{2 SPACES}2, 3, OR 4"          :rem 78
780 GOSUB 2020                       :rem 227
790 GET E$:IF E$<"2" OR E$>"4" THEN 790
                                     :rem 92
800 N=VAL(E$)                        :rem 192
810 FOR I=1 TO N                      :rem 40
820 PRINT "{3 DOWN}CHOOSE COLOR ";I
                                     :rem 98
830 PRINT "{DOWN}{RVS}{BLK} {OFF}
    {2 SPACES}{RVS}{WHT} {OFF}{2 SPACES}
    {RVS}{RED} {OFF}{2 SPACES}{RVS}{CYN}
    {SPACE}{OFF}{2 SPACES}{RVS}{PUR}
    {OFF}{2 SPACES}{RVS}{GRN} {OFF}
    {2 SPACES}{RVS}{BLU} {OFF}{2 SPACES}
    {RVS}{YEL} {OFF}{2 SPACES}{RVS}{13}
    {SPACE}{4}"                      :rem 142
840 PRINT "0{2 SPACES}1{2 SPACES}2
    {2 SPACES}3{2 SPACES}4{2 SPACES}5
```

```
    {2 SPACES}6{2 SPACES}7{2 SPACES}8"
                                        :rem 65
850 GOSUB 2020                          :rem 225
860 GET E$:IF E$<"0" OR E$>"8" THEN 860
                                        :rem 90
870 C(I)=VAL(E$)                        :rem 86
880 PRINT TAB(3*C(I))"↑"                :rem 49
890 NEXT I                              :rem 41
900 PRINT "{CLR}"                       :rem 253
910 POKE 53282,C(2):POKE 53283,C(3):POKE
    {SPACE}53284,C(4)                   :rem 131
920 POKE 53265,PEEK(53265) OR 64        :rem 129
930 REM PRINT POSSIBLE SQUARES          :rem 146
940 REM TWO COLORS                      :rem 77
950 CC=C(1)                             :rem 38
960 POKE 1139,96:POKE 1139+LF,CC        :rem 221
970 POKE 1142,103:POKE 1142+LF,CC:rem 247
980 POKE 1259,99:POKE 1259+LF,CC        :rem 232
990 POKE 1262,100:POKE 1262+LF,CC:rem 252
1000 POKE 1379,101:POKE 1379+LF,CC:rem 46
1010 POKE 1382,102:POKE 1382+LF,CC:rem 36
1020 IF N=2 THEN 1260                    :rem 7
1030 POKE 1136,160:POKE 1136+LF,CC:rem 36
1040 POKE 1256,163:POKE 1256+LF,CC:rem 46
1050 POKE 1376,163:POKE 1376+LF,C(2)
                                        :rem 117
1060 POKE 1496,165:POKE 1496+LF,CC:rem 62
1070 POKE 1499,164:POKE 1499+LF,C(2)
                                        :rem 132
1080 POKE 1502,164:POKE 1502+LF,CC:rem 39
1090 POKE 1616,165:POKE 1616+LF,C(2)
                                        :rem 117
1100 POKE 1619,166:POKE 1619+LF,C(2)
                                        :rem 116
1110 POKE 1622,166:POKE 1622+LF,CC:rem 41
1120 IF N=3 THEN 1260                    :rem 9
1130 POKE 1133,224:POKE 1133+LF,CC:rem 32
1140 POKE 1253,227:POKE 1253+LF,CC:rem 42
1150 POKE 1373,227:POKE 1373+LF,C(2)
                                        :rem 113
1160 POKE 1493,227:POKE 1493+LF,C(3)
                                        :rem 121
1170 POKE 1613,229:POKE 1613+LF,CC:rem 47
1180 POKE 1733,229:POKE 1733+LF,C(2)
                                        :rem 118
1190 POKE 1853,229:POKE 1853+LF,C(3)
                                        :rem 126
1200 POKE 1736,230:POKE 1736+LF,C(3)
                                        :rem 110
1210 POKE 1739,230:POKE 1739+LF,C(2)
                                        :rem 116
1220 POKE 1742,230:POKE 1742+LF,C(1)
                                        :rem 104
1230 POKE 1856,228:POKE 1856+LF,C(3)
                                        :rem 126
1240 POKE 1859,228:POKE 1859+LF,C(2)
                                        :rem 132
1250 POKE 1862,228:POKE 1862+LF,CC:rem 57
1260 PRINT "{6 DOWN}"                    :rem 0
1270 FOR I=1 TO 4                       :rem 63
1280 PRINT TAB(8)A$:PRINT TAB(8)B$
                                        :rem 170
1290 NEXT I                             :rem 84
1300 FOR T=1 TO 16                      :rem 119
1310 GOSUB 2020                          :rem 9
1320 POKE S(T),63                       :rem 130
1330 FOR I=1 TO QQ(N)                   :rem 73
1340 GOSUB 2020                         :rem 12
1350 PP=0                               :rem 214
1360 P=PEEK(Q(I)+LF):IF P=0 THEN PP=1
                                        :rem 56
1370 POKE Q(I),28:POKE Q(I)+LF,PP:POKE Q(
     I),R(I):POKE Q(I)+LF,P             :rem 81
1380 GET E$:IF E$="" THEN 1370          :rem 194
1390 IF ASC(E$)=13 THEN 1430            :rem 133
1400 IF E$<>"{F1}" THEN 1370            :rem 250
1410 NEXT I                             :rem 78
1420 GOTO 1330                          :rem 199
1430 A=S(T)                             :rem 62
1440 IF I>14 THEN 1470                  :rem 63
1450 ON I GOSUB 30,40,50,60,70,80,90,100,
     110,120,130,140,150,160            :rem 66
1460 GOTO 1480                          :rem 209
1470 ON I-14 GOSUB 170,180,190,200,210,22
     0,230,240,250,260,270,280,290,300
                                        :rem 63
1480 GOSUB 2040                         :rem 19
1490 NEXT T                             :rem 97
1500 PRINT "{HOME}{17 DOWN}"            :rem 203
1510 PRINT "PRESS F1 TO CHANGE"         :rem 229
1520 PRINT "{6 SPACES}F7 TO PRINT QUILT"
                                        :rem 213
1530 GOSUB 2020                         :rem 13
1540 GET E$:IF E$="{F7}" THEN 1880:rem 78
1550 IF E$<>"{F1}" THEN 1540            :rem 255
1560 FOR I=1744 TO 1766:POKE I,32:POKE I+
     40,32:NEXT I                       :rem 87
1570 PRINT "{2 UP}PRESS RETURN--NO CHANGE
     "                                  :rem 202
1580 PRINT TAB(6)"F1 TO CHANGE, THEN"
                                        :rem 24
1590 PRINT TAB(9)"PROCEED AS BEFORE"
                                        :rem 74
1600 FOR T=1 TO 16                      :rem 122
1610 PS=0:A=S(T):GOSUB 2020              :rem 6
1620 PT=PEEK(A+LF):IF PT=0 THEN PS=1
                                        :rem 56
1630 GOSUB 2050                         :rem 17
1640 POKE A,40:POKE A+1,41:POKE A+40,27:P
     OKE A+41,29                         :rem 1
1650 GOSUB 2060                         :rem 20
1660 GET E$:IF E$="" THEN 1640          :rem 195
1670 IF ASC(E$)=13 THEN 1850            :rem 140
1680 IF E$<>"{F1}" THEN 1640             :rem 4
1690 POKE A,40:POKE A+1,41:POKE A+40,27:P
     OKE A+41,29                         :rem 6
1700 FOR I=1 TO QQ(N)                   :rem 74
1710 GOSUB 2020                         :rem 13
1720 PP=0                               :rem 215
1730 P=PEEK(Q(I)+LF):IF P=0 THEN PP=1
                                        :rem 57
1740 POKE Q(I),28:POKE Q(I)+LF,PP:POKE Q(
     I),R(I):POKE Q(I)+LF,P             :rem 82
1750 GET E$:IF E$="" THEN 1740          :rem 196
1760 IF ASC(E$)=13 THEN 1800            :rem 135
1770 IF E$<>"{F1}" THEN 1740             :rem 5
1780 NEXT I                             :rem 88
1790 GOTO 1700                          :rem 210
1800 IF I>14 THEN 1830                  :rem 63
1810 ON I GOSUB 30,40,50,60,70,80,90,100,
     110,120,130,140,150,160            :rem 66
1820 GOTO 1840                          :rem 209
1830 ON I-14 GOSUB 170,180,190,200,210,22
     0,230,240,250,260,270,280,290,300
                                        :rem 63
1840 GOSUB 2040                         :rem 19
1850 NEXT T                             :rem 97
1860 FOR I=1744 TO 1769:POKE I,32:POKE I+
     40,32:POKE I+80,32:NEXT I          :rem 51
1870 GOTO 1500                          :rem 207
1880 FOR T=1 TO 16                      :rem 132
1890 A=S(T)                             :rem 72
```

```
1900 GOSUB 2050                          :rem 17
1910 CC=PEEK(A+LF)                       :rem 2
1920 FOR B=S(T)-320 TO S(T)+320 STEP 320
                                         :rem 24
1930 FOR A=B-8 TO B+24 STEP 8            :rem 195
1940 GOSUB 2060:GOSUB 2040               :rem 150
1950 NEXT A,B                            :rem 189
1960 NEXT T                              :rem 99
1970 PRINT "{4 DOWN}PRESS F7 TO END PROGR
     AM.";                               :rem 235
1980 GOSUB 2020                          :rem 22
1990 GET E$:IF E$="{F7}" THEN 2070:rem 79
2000 GOTO 1990                           :rem 206
2010 STOP                                :rem 9
2020 POKE 198,0:POKE HF,84:POKE LF,125
                                         :rem 63
2030 POKE W,17:FOR D=1 TO 60:NEXT D:POKE
     {SPACE}W,0:RETURN                   :rem 174
2040 L=A+LF:POKE L,CC:POKE L+1,CC:POKE L+
     40,CC:POKE L+41,CC:RETURN           :rem 126
2050 A1=PEEK(A):A2=PEEK(A+1):A3=PEEK(A+40
     ):A4=PEEK(A+41):RETURN              :rem 168
2060 POKE A,A1:POKE A+1,A2:POKE A+40,A3:P
     OKE A+41,A4:RETURN                  :rem 73
2070 PRINT"{CLR}"                        :rem 45
2080 PRINT "PRESS RUN/STOP - RESTORE"
                                         :rem 227
2090 PRINT "BEFORE RUNNING PROGRAM AGAIN.
     {2 DOWN}"                           :rem 56
2100 END                                 :rem 154
```

Robot Math

See article on page 90.

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: Robot Math—VIC Version

```
1 Q=108:READP1$,P2,P3$,P4:GOTO62   :rem 95
2 POKEB,0:POKEB+1,0:POKEB+2,0:POKEB+4,0:R
  ETURN                            :rem 70
3 POKEB+4,15:POKEB+1,180:FORM=1TOC:NEXT:G
  OTO2                             :rem 65
4 F=INT(RND(1)*9):RETURN           :rem 214
5 POKED,3:POKED+1,3:POKEB+4,15:FORL=99TO2
  55:POKEB+2,L:POKEB,L:NEXT:GOSUB2:GOTO92
                                   :rem 76
6 POKEB+4,15:POKEB+2,160:FORM=1TO400:NEXT
  :GOTO2                           :rem 148
7 PRINTM$;" {RVS}{GRN}PRESS RTN TO CHANGE
  ":RETURN                         :rem 200
8 PRINTM$;"{5 SPACES}{RVS}{RED}ENTER DIGI
  T{OFF}{4 SPACES}":RETURN         :rem 126
9 IFF>5THENL=126:IFF>9THENL=Q:M=1:IFF>30T
  HENL=90:M=0:IFF>50THENL=Q:M=1:IFF>99THE
  NF=0                             :rem 108
10 POKEE,L:POKEE+1,L:POKED+21,M:POKED+24,
   M:GOTO36                        :rem 22
11 PRINT"{HOME}":PRINTTAB(L);M$;O$;:GOSUB
   3:PRINTN$:RETURN                :rem 86
12 M$=" N{2 T}M {DOWN}{6 LEFT} MZZN
   {DOWN}{6 LEFT} {F}L@{D} {DOWN}{6 LEFT}
    £{Q}{W}{*} {DOWN}{6 LEFT}{2 SPACES}O
   P{2 SPACES}{DOWN}{6 LEFT} {RVS}£
   {2 SPACES}{*}{OFF} "            :rem 222
13 N$="{6 LEFT} WWWW ":O$="{DOWN}{6 LEFT}
    ZZZZ ":B=36874:READP1$,P2,P3$,P4:POKE
   B+5,30:POKE649,1                :rem 170
14 PRINT"{CLR}{9 DOWN}{RVS}{GRN} M=MENU--
   ---DEL=ERASE {HOME}":L8=48:POKE143,PEE
   K(162):GOTO92                   :rem 72
15 POKE651,255:PRINT"{HOME}":R=38649:S=79
   29:Y=P2:D=38446:E=7726:IFP1$="-"ORP2=1
   ORP2>3THENY=2                   :rem 115
16 FORI=P2TO1STEP-1:A(I)=0:S(I)=0:FORK=1T
   OY:GOSUB3:GOSUB4                :rem 139
17 A(I)=A(I)+F:S(I)=S(I)-F         :rem 144
18 M=(K*22)+I:POKER+M,4:POKES+M,F+48:IFK=
   1THENL1=F                       :rem 18
19 NEXT:IFP1$="-"THENGOSUB57:GOTO21
                                   :rem 182
20 L=A(I):N=9:GOTO22               :rem 155
21 S(I)=S(I)+2*L1:L=S(I):N=0       :rem 35
22 GOSUB49:A(I)=L                  :rem 239
23 NEXTI                           :rem 237
24 FORK=1TOY:FORI=1TOP2            :rem 255
25 M=K*22+I:IFPEEK(S+M)>48THEN28   :rem 139
26 IFI=P2THENV=1                   :rem 242
27 POKES+M,32:NEXT                 :rem 109
28 NEXTK:IFV=1THENV=0:GOTO15       :rem 76
29 A=0:U=-1:FORI=P2TO1STEP-1:U=U+1:IFP1$=
   "-"THENA=A+S(I)*10↑U:GOTO31     :rem 142
30 A=A+A(I)*10↑U                   :rem 102
31 NEXT                            :rem 163
32 A=INT(A):L2=0:IFA<0THEN15       :rem 144
33 PRINT"{11 DOWN}":FORI=2TOY:PRINTTAB(7)
   ;P1$:NEXT:POKE160,0:POKE161,0:POKE162,
   0                               :rem 93
34 PRINT"{4 UP}":FORK=0TOP2:PRINTTAB(7+K)
   ;"{3 DOWN}C{DOWN}{LEFT} {5 UP}":NEXT
                                   :rem 135
35 PRINT"{3 DOWN}":U=LEN(STR$(A))-2:I=0:L
   1=0:FORK=P2TOP2-USTEP-1         :rem 246
36 FORM=6TO8:POKEM+E+154,ASC(MID$(TI$,M-2
   ))+128:NEXT:IFTI$="000400"THENGOSUB6:G
   OTO46                           :rem 66
37 GETA$:IFA$=""THENL=124:F=F+1:GOTO9
                                   :rem 184
38 IFASC(A$)=20THENPRINTTAB(7);"
   {7 SPACES}";"{5 UP}":GOTO35     :rem 239
39 IFA$="M"THENPOKEB+5,27:GOTO62   :rem 158
40 IFA$<"0"ORA$>"9"THEN37          :rem 98
41 L1=INT(L1+VAL(A$)*10↑I):I=I+1:PRINTTAB
   (7+K);A$:PRINT"{2 UP}":NEXT     :rem 21
42 IFL1=ATHENGOSUB5                :rem 212
43 IFL1<>ATHENGOSUB6               :rem 19
44 L2=L2+1:IFL2>2THEN46            :rem 77
45 PRINT"{2 UP}":GOTO34            :rem 54
46 V=0:AN$=STR$(A):L=LEN(AN$):IFL>P2+1THE
   NV=1                            :rem 208
47 IFL-1<P2THENV=L-1-P2            :rem 125
48 PRINTTAB(8-V);"{RVS}";MID$(AN$,2,8):FO
   RK=1TO3500:NEXT:GOTO15          :rem 3
49 IFP3$="N"ANDP1$="+"THEN52       :rem 44
50 GOSUB4:X=1:IFL<N+FTHEN54        :rem 193
51 RETURN                          :rem 70
52 IFL>NTHENX=-1:GOTO54            :rem 204
53 RETURN                          :rem 72
54 L=0:FORK=1TOY:M=(K*22)+I:F=PEEK(S+M)+X
   :IFF<48THENF=48                 :rem 176
55 IFF>57THENF=57                  :rem 7
```

```
56 POKES+M,F:L=L+(F-48):NEXT:GOTO49
                                   :rem 154
57 IFP3$="N"THEN60                  :rem 6
58 IFI=1ORF>=L1THENRETURN          :rem 161
59 GOTO61                           :rem 14
60 IFF<L1THENRETURN                 :rem 3
61 POKES+M,L1+48:POKES+M-22,F+48:S(I)=(-F
   )-L1:L1=F:RETURN                :rem 149
62 M$="{HOME}{16 DOWN}":PRINT"{CLR}
   {3 DOWN}OPERATION (+/-).... ";P1$
                                   :rem 111
63 PRINT"{DOWN}# DIGITS (MAX=6)...";P2:PR
   INT"CARRY/BORROW..... ";P3$ :rem 221
64 PRINT"{DOWN}# PROBLEMS (MAX=9).";P4:PR
   INT"{DOWN}{4 SPACES}EEEEEEEEEEEEEE
                                   :rem 45
65 GOSUB7:PRINT"{3 DOWN}{5 SPACES}{CYN}
   {RVS}(B TO BEGIN)";"{GRN}{HOME}SELECT:
   USE CRSR(UP/DN){BLK}            :rem 162
66 M=7746                          :rem 206
67 IFM1=7878THENM=7746              :rem 38
68 FORI=MTOM+20:POKEI,PEEK(I)+128:NEXT
                                   :rem 129
69 GETA$:IFA$=""THEN69:A=A+128:POKEI,A:NE
   XT                              :rem 88
70 IFVAL(A$)<10ANDVAL(A$)>0THEN82:rem 159
71 IFA$="+"ORA$="-"ORA$="Y"ORA$="N"ORPEEK
   (197)=15THENA$="1":GOTO82       :rem 239
72 IFA$="{DOWN}"THEN77             :rem 140
73 IFA$="{UP}"THEN81                :rem 8
74 IFA$<>"B"THEN69                 :rem 253
75 PRINT"{WHT}{CLR}{3 DOWN}91 DATA";P1$;"
   ,";P2;",";P3$;",";P4:PRINT"RUN12";"
   {HOME}                          :rem 158
76 :POKE198,3:POKE631,13:POKE632,13:POKE6
   33,13:END                       :rem 164
77 M2=M2-1:K=44                     :rem 49
78 M=M+K:FORI=M-KTOM-K+20:POKEI,PEEK(I)-1
   28:NEXT:IFM>7878THENM=7746      :rem 190
79 IFM<7746THENM=7878              :rem 247
80 ON(M-7702)/44GOSUB7,8,7,8:GOTO67
                                   :rem 249
81 M2=M2-1:K=-44:GOTO78             :rem 59
82 ON(M-7746)/44GOTO85,87,90:IFP1$="+"THE
   NP1$="-":GOTO84                 :rem 146
83 P1$="+"                         :rem 188
84 POKEM+20,ASC(P1$)+128:GOTO69     :rem 22
85 IFVAL(A$)>6THENA$="6"            :rem 84
86 P2=VAL(A$):POKEM+20,P2+176:GOTO69
                                   :rem 98
87 IFP3$="N"THENP3$="Y":GOTO89      :rem 8
88 P3$="N"                         :rem 230
89 POKEM+20,64+ASC(P3$):GOTO69     :rem 236
90 P4=VAL(A$):POKEM+20,P4+176:GOTO69
                                   :rem 97
91 DATA+, 1 ,Y, 2                  :rem 239
92 C=0:PRINT"{HOME}{BLK}":FORL=0TO15:GOSU
   B11:NEXT                        :rem 65
93 FORL=14TO0STEP-1:GOSUB11:NEXT:PRINT"
   {BLK}":C=40                     :rem 252
94 L8=L8+1:M=34816+8*L8:PRINT"{HOME}":IFL
   8-48>P4THEN98                   :rem 79
95 FORM1=MTOM+6:X=PEEK(M1):FORL=1TO7:C=32
   :X=X*2:IFX>255THENX=X-256:C=L8:rem 231
96 PRINTTAB(13)"{CYN}";CHR$(C);:NEXT:PRIN
   T"{BLK}":NEXT:IFL8-48>P4THEN98 :rem 75
97 GOTO15                          :rem 15
98 POKEB+5,27:PRINT"{CLR}";SPC(176);"
   {RVS}PLAY ANOTHER GAME(Y/N){OFF}"
                                   :rem 48
99 GETZ$:IFZ$=""OR(Z$<>"Y"ANDZ$<>"N")THEN
   99                              :rem 29
100 IFZ$="N"THENEND                :rem 115
101 RUN1                           :rem 184
```

Program 2: Robot Math—64 Version

```
10 PRINT"{CLR}":POKE53281,1:POKE53280,5:R
   EADP1$,P2,P3$,P4:GOTO710        :rem 145
12 READP1$,P2,P3$,P4:B=54272        :rem 18
15 FORI=BTOB+24:POKEI,0:NEXT:VO=B+24:AD=B
   +5:SR=AD+1:HF=B+1:LF=B:POKEAD,20:rem 6
16 POKESR,200:SO=B+4:GOTO130        :rem 186
20 POKESO,32:RETURN                 :rem 222
30 POKEHF,50:POKELF,40:POKESO,33:FORM=1TO
   C:NEXT:GOTO20                   :rem 233
40 F=INT(RND(1)*9):RETURN            :rem 6
50 POKESO,33:FORL=99TO255:POKEHF,L:POKELF
   ,50:NEXT:GOSUB20:GOTO1050       :rem 236
60 POKESO,33:POKEHF,60:POKELF,50:FORM=1TO
   400:NEXT:GOTO20                 :rem 63
70 PRINTM$;"{8 SPACES}{RVS}{BLU}PRESS RET
   URN TO CHANGE":RETURN           :rem 229
80 PRINTM$;"{13 SPACES}{RVS}{BLU}ENTER DI
   GIT{OFF}{8 SPACES}":RETURN      :rem 177
90 IFF>5THENL=126:IFF>9THENL=108:M=1:IFF>
   30THENL=90:M=0:IFF>50THENL=108:M=1
                                   :rem 139
100 IFF>99THENF=0                  :rem 248
110 POKEE,L:POKEE+1,L:GOTO430       :rem 10
120 PRINT"{HOME}":PRINTTAB(L);M$;O$;:POKE
    VO,15:GOSUB30:POKEVO,0:PRINTN$:RETURN
                                   :rem 224
130 POKE649,1:M$=" N[2 T]M {DOWN}{6 LEFT}
     MZZN {DOWN}{6 LEFT} [F]L@[D] {DOWN}
    {6 LEFT} £[Q][W][*] {DOWN}{6 LEFT}
    {2 SPACES]OP"                  :rem 141
140 M$=M$+"{2 SPACES}{DOWN}{6 LEFT} {RVS}
    £{2 SPACES}[*]{OFF} "          :rem 14
150 N$="{6 LEFT} WWWW ":O$="{DOWN}
    {6 LEFT} ZZZZ "                :rem 232
160 PRINT"{CLR}{9 DOWN}{RVS}{GRN} M=MENU-
    ----------DEL=ERASE {OFF}
    {HOME}"                        :rem 181
165 POKE214,23:PRINT:POKE211,15    :rem 73
170 L8=48:POKE143,PEEK(162):GOTO1050
                                   :rem 58
180 POKE651,255:PRINT"{HOME}":R=54272:S=1
    561:Y=P2:E=1106:RW=16:WR=RW-4:POKEVO,
    15                             :rem 213
190 IFP2=3THENRW=17:WR=RW-5        :rem 195
200 IFP1$="-"ORP2=1ORP2>3THENY=2   :rem 144
210 POKE214,RW:PRINT:POKE211,17:PRINT"
    {7 SPACES}"                    :rem 145
220 FORI=P2TO1STEP-1:A(I)=0:S(I)=0:FORK=1
    TOY:GOSUB30:GOSUB40             :rem 24
230 A(I)=A(I)+F:S(I)=S(I)-F        :rem 189
240 M=(K*40)+I:POKER+S+M,0:POKES+M,F+48:I
    FK=1THENL1=F                   :rem 185
250 NEXT:IFP1$="-"THENGOSUB660:GOTO270
                                   :rem 73
260 L=A(I):N=9:GOTO280              :rem 7
270 S(I)=S(I)+2*L1:L=S(I):N=0      :rem 89
280 GOSUB580:A(I)=L                :rem 85
290 NEXTI                          :rem 35
300 FORK=1TOY:FORI=1TOP2           :rem 44
310 M=K*40+I:IFPEEK(S+M)>48THEN340
                                   :rem 229
320 IFI=P2THENV=1                   :rem 31
330 POKES+M,32:NEXT                :rem 154
340 NEXTK:IFV=1THENV=0:GOTO180     :rem 172
```

```
350 A=Ø:U=-1:FORI=P2TO1STEP-1:U=U+1:IFP1$
    ="-"THENA=A+S(I)*10↑U:GOTO370:rem 241
360 A=A+A(I)*10↑U                  :rem 156
370 NEXT                           :rem 217
380 A=INT(A):L2=Ø:IFA<ØTHEN180     :rem 249
390 FORI=2TOY:POKE214,WR+I:PRINT:POKE211,
    17:PRINT"{BLK}"P1$:NEXT        :rem 145
400 POKE160,Ø:POKE161,Ø:POKE162,Ø:rem 113
410 FORK=ØTOP2:POKE214,RW-1:PRINT:POKE211
    ,17+K:PRINT"C":NEXT            :rem 161
420 U=LEN(STR$(A))-2:I=Ø:L1=Ø:FORK=P2TOP2
    -USTEP-1                       :rem 230
430 FORM=15TO17:POKEM+E+R+28Ø,Ø:POKEM+E+2
    8Ø,ASC(MID$(TI$,M-11))+128:NEXT
                                   :rem 211
440 IFTI$="ØØØ4ØØ"THENGOSUB60:GOTO550
                                   :rem 175
450 GETA$:IFA$=""THENL=124:F=F+1:GOTO9Ø
                                   :rem 23
460 IFASC(A$)=2ØTHENPOKE214,RW:PRINT:POKE
    211,17:PRINT"{7 SPACES}":GOTO42Ø
                                   :rem 139
470 IFA$="M"THEN71Ø                :rem 36
480 IFA$<"Ø"ORA$>"9"THEN44Ø        :rem 200
490 PRINT"{DOWN}":L1=INT(L1+VAL(A$)*10↑I)
    :I=I+1                         :rem 135
500 POKE214,RW:PRINT:POKE211,17+K:PRINTA$
    :NEXT                          :rem 163
510 IFL1=ATHENGOTO5Ø               :rem 237
520 IFL1<>ATHENGOSUB6Ø             :rem 115
530 L2=L2+1:IFL2>2THEN55Ø          :rem 173
540 GOTO41Ø                        :rem 103
550 V=Ø:AN$=STR$(A):L=LEN(AN$):IFL>P2+1TH
    ENV=1                           :rem Ø
560 IFL-1<P2THENV=L-1-P2           :rem 173
570 POKE214,RW:PRINT:POKE211,18-V:PRINT"
    {RVS}";MID$(AN$,2,8)           :rem 47
575 FORK=1TO35ØØ:NEXT:GOTO18Ø      :rem 49
580 IFP3$="N"ANDP1$="+"THEN61Ø     :rem 140
590 GOSUB4Ø:X=1:IFL<N+FTHEN63Ø     :rem 90
600 RETURN                         :rem 118
610 IFL>NTHENX=-1:GOTO63Ø          :rem 44
620 RETURN                         :rem 120
630 L=Ø:FORK=1TOY:M=(K*4Ø)+I:F=PEEK(S+M)+
    X:IFF<48THENF=48               :rem 224
640 IFF>57THENF=57                 :rem 55
650 POKES+M,F:L=L+(F-48):NEXT:GOTO58Ø
                                   :rem 250
660 IFP3$="N"THEN69Ø               :rem 111
670 IFI=1ORF>=L1THENRETURN         :rem 209
680 GOTO7ØØ                        :rem 110
690 IFF<L1THENRETURN               :rem 60
700 POKES+M,L1+48:POKES+M-4Ø,F+48:S(I)=(-
    F)-L1:L1=F:RETURN              :rem 197

710 M$="{HOME}{16 DOWN}"           :rem 173
720 PRINT"{CLR}{BLK}{3 DOWN}{8 RIGHT}OPER
    ATION (+/-).... ";P1$          :rem 201
730 PRINT"{DOWN}{8 RIGHT}# DIGITS (MAX=6)
    ...";P2                        :rem 60
740 PRINT"{DOWN}{8 RIGHT}CARRY/BORROW....
    ... ";P3$                      :rem 20
750 PRINT"{DOWN}{8 RIGHT}# PROBLEMS (MAX=
    9).";P4                        :rem 135
760 PRINT"{DOWN}{8 RIGHT}{4 SPACES}EEEEEE
    EEEEEEEE                       :rem 11
770 GOSUB7Ø:PRINT"{3 DOWN} {8 RIGHT}
    {4 SPACES}{BLU}{RVS}(B TO BEGIN)";
                                   :rem 178
780 PRINT"{GRN}{HOME}{8 RIGHT}SELECT:USE
    {SPACE}CRSR(UP/DN){BLK}"       :rem 241
790 M=1152                         :rem 24?
800 IFM1=1392THENM=1152            :rem 51
810 FORI=MTOM+2Ø:X=PEEK(I):POKEI,X+128:NE
    XT                             :rem 211
820 GETA$:IFA$=""THEN82Ø           :rem 87
830 IFVAL(A$)<1ØANDVAL(A$)>ØTHEN95Ø:rem 7
840 IFA$="+"ORA$="-"ORA$="Y"ORA$="N"ORPEE
    K(197)=1THENA$="1":GOTO95Ø     :rem 34
850 IFA$="{DOWN}"THEN9ØØ           :rem 235
860 IFA$="{UP}"THEN94Ø             :rem 112
870 IFA$<>"B"THEN82Ø               :rem 92
880 PRINT"{WHT}{CLR}{3 DOWN}1Ø4Ø DATA";P1
    $;",";P2;",";P3$;",";P4:PRINT"RUN12";
    "{HOME}"                       :rem 79
890 :POKE198,3:POKE631,13:POKE632,13:POKE
    633,13:END                     :rem 216
900 M2=M2-1:K=8Ø                   :rem 92
910 M=M+K:FORI=M-KTOM-K+20:X=PEEK(I):POKE
    I,X-128:NEXT:IFM>1392THENM=1152
                                   :rem 242
920 IFM<1152THENM=1392              :rem 4
930 ON(M-1Ø64)/8ØGOSUB7Ø,8Ø,7Ø,8Ø:GOTO8ØØ
                                   :rem 19
940 M2=M2-1:K=-8Ø:GOTO91Ø          :rem 154
950 ON(M-1152)/8ØGOTO98Ø,1ØØØ,1Ø3Ø:IFP1$=
    "+"THENP1$="-":GOTO97Ø         :rem 204
960 P1$="+"                        :rem 240
970 POKEM+2Ø,ASC(P1$)+128:GOTO82Ø:rem 117
980 IFVAL(A$)>6THENA$="6"          :rem 136
990 P2=VAL(A$):POKEM+2Ø,P2+176:GOTO82Ø
                                   :rem 193
1ØØØ IFP3$="N"THENP3$="Y":GOTO1Ø2Ø
                                   :rem 172
1Ø1Ø P3$="N"                        :rem 56
1Ø2Ø POKEM+2Ø,64+ASC(P3$):GOTO82Ø:rem 105
1Ø3Ø P4=VAL(A$):POKEM+2Ø,P4+176:GOTO82Ø
                                   :rem 231
1Ø4Ø DATA+, 1 ,Y, 1                 :rem 73
1Ø5Ø C=Ø:PRINT"{HOME}{BLK}":FORL=ØTO34:GO
    SUB12Ø:NEXT                    :rem 206
1Ø6Ø FORL=33TOØSTEP-1:GOSUB12Ø:NEXT:PRINT
    "{BLU}":C=4Ø                   :rem 24
1Ø7Ø POKE56334,PEEK(56334)AND254:POKE1,PE
    EK(1)AND251                    :rem 233
1Ø8Ø L8=L8+1:IFL8-48>P4THEN112Ø     :rem 5
1Ø9Ø M=53247+8*L8:PRINT"{HOME}"; :rem 195
11ØØ FORM1=MTOM+7:X=PEEK(M1):FORL=1TO7:C=
    32:X=X*2:IFX>255THENX=X-256:C=2Ø9
                                   :rem 83
1111Ø PRINTTAB(3Ø)"{BLK}"CHR$(C);:NEXT:PRI
    NT"{7 LEFT}{DOWN}";:NEXT       :rem 19
1120 POKE1,PEEK(1)OR4:POKE56334,PEEK(5633
    4)OR1                          :rem 179
1130 IFL8-48>P4THEN115Ø             :rem 41
1140 GOTO18Ø                       :rem 152
1150 PRINT"{CLR}":POKE214,12:PRINT:POKE21
    1,4                            :rem 222
1155 PRINT"{RVS}{BLK}HOW ABOUT ANOTHER GA
    ME (Y/N)?{OFF}"                :rem 203
1160 GETZ$:IFZ$=""OR(Z$<>"Y"ANDZ$<>"N")TH
    EN116Ø                         :rem 201
1170 IFZ$="Y"THENRESTORE:CLR:GOTO1Ø
                                   :rem 242
1180 END                          :rem 161
```

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

PRODUCT MART

VIC-20 & 64

What *is* a Cheatsheet, anyway?

Leroy's Cheatsheet™ Keyboard overlays are durable plastic-coated templates. When simply laid on your VIC-20 and 64 keyboards, the **Leroy's Cheatsheet**™ surrounds the keys with essential information, placing your most valuable programming tool at your fingertips.

At **Cheatsheet Products**™ we take the time to learn and use each program before designing a keyboard overlay. Not only are our overlays designed using easy to follow instructions and illustrations, but all commands are available and many extras are added to make programming easy and fun.

Our BASIC **Leroy's Cheatsheet**™ not only has all commands and functions, but also has device numbers, program list printing commands, disk commands, and many illustrative examles of the actual BASIC commands in the cutout.

Leroy's Cheatsheet™ overlays make it all easy for only **$3.95.**

Dealer inquiries welcome

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

ONLY $3.95 each

WORD PROCESSORS

20	64	
	☐	EASY SCRIPT[1]
☐	☐	HES WRITER
	☐	PAPER CLIP
☐	☐	QUICK BROWN FOX
	☐	SCRIPT 64
☐		UMI WORDCRAFT 20
☐		VIC TYPEWRITER
	☐	WORDPRO 3/PLUS

LANGUAGES & UTILITIES

20	64	
☐	☐	BASIC
	☐	HESMON 64
☐		HES VIC FORTH
	☐	PILOT (COMMODORE)
☐		PROGRAMMER'S AID CART[1]
☐		VICMON

SPREADSHEETS

20	64	
	☐	CALC RESULT (ADVANCED)
	☐	CALC RESULT (EASY)
	☐	EASY CALC
	☐	HES/MICROSOFT MULTIPLAN
☐	☐	PRACTICALC 64/PLUS

MISCELLANEOUS

☐	PRINTER (CBM) 1525, MPS-801[1]
☐	PRINTER (CBM) 1526[1]
☐	PRINTER EPSON RX-80

GAZ

Send check or money order plus **$1.00** (postage and handling)
PA residents add 6% sales tax. C.O.D. — add $3.00.

Name _____
Address _____
City _____ State _____ Zip _____

(1) Products of Commodore Business Machines, Inc.

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

YOUR VOICE IN - YOUR VOICE OUT
Digital Recording on C-64/VIC20

Up to 64 numbered words or phrases. Then store as a named file on disk or tape. Words or phrases out in any order from your own BASIC program. New BASIC Commands added. The Voice Master is not needed for response—only for recording. Talking games, clocks, calculators, file data, machine response, advisories—applications too numerous to list. Wherever you want a talking computer with your own natural sounding voice and your own custom vocabulary. Even sing and play music. Many applications in education too. Software for word recognition soon available.

ONLY $89.95

WE CAN DEMONSTRATE OVER THE TELEPHONE!!

COVOX INC.

675-D Conger St. Eugene, OR 97402
Tel: (503) 342-1271, Telex 706017
Check, money order, or VISA/MC
(Add $4.00 Shipping and Handling)

FLY Your Computer

FLIGHT SIMULATOR GAMES

NEW COCKPIT 64
For the Commodore 64
- 100% Machine Language
- Windshield View
- 7 Airports

$30.

Runway 64 (Commodore 64)	$25
Runway 20 (VIC-20	$25
Sky Pilot (VIC-20)	$18

ADD $2.00 FOR DISK VERSION

BAR GRAPH 64
FOR THE COMMODORE 64
- 100% Machine Language
- Full Color Display
- Saves To Disk
- Supports All Parallel Printers

$30.

COD ORDER PHONE
WE SHIP WITHIN 48 HOURS
(312) 394-5165

SUSIE SOFTWARE
709 Wilshire Dr.
Mt. Prospect, IL 60056

PRODUCT MART

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

ARE YOU STILL HAVING PROBLEMS KEYING IN THOSE FREE MAGAZINE PROGRAMS ??

Get the NEW Magazine/Copy Holder **MAG-RAK**

- Holds your ENTIRE magazine
- Line-guides your work for fast, easy & less tiresome copying
- No magnets
- Sturdy, compact & attractive

Start saving valuable time & effort now!

Order your MAG-RAK today!

R&L PRODUCTS
P.O. BOX 26A89
LOS ANGELES, CA 90026

Only 34.95 POSTPAID

*Dealer inquiries invited Calif. add 6.5% tax Canada add $2

VISIT LAS VEGAS

on your COMMODORE 64

ADVANCED MICROWARE Introduces:

-------- 64 CASINO PAC ----------

Includes:

* SLOT MACHINE * POKER
* BLACKJACK * KENO

All for only $39
Each program is a graphic simulation of their respective Vegas video gaming machine. These are colorful and exciting games you will enjoy over and over. Practice your "system" or just play for fun.

Also available: 64TOUR
This is a Tour and demo of the many features of your 64 including a version of BASIC with new graphics commands.
Only $15

Send Check or Money Order to:
ADVANCED MICROWARE
P.O. BOX 6143 Dept. CG-4G
SANTA ANA, CA. 92706
Specify DISK or TAPE, CA. orders-add 6%
Dealer Inquiries Invited (714)554-6470

NEW AND DIFFERENT FOR YOUR C-64

First offering: for C-64, disk & printer. Ten programs and **understandable text** to create & maintain a **home inventory system** using random access relative files. Step-by-step instruction teaches new **fill-in format.** R.F.M.E. — relative files made easy. Each program explained line-by-line. Nothing to look for elsewhere. Relative files are the best part of your system. Learn to use them well. Programs can be modified later for other usage. Disk of ten programs and full text, for a limited time only, at a special introductory price of $35. Also includes text on advanced relative files. (PA add 6%). Check or money order to: W.J. Hurrell, 1291 Cedar Blvd., Pittsburgh, PA 15228. Sorry disk only, no tape.

WJH

COMMODORE® IN THE CLOSET?

Stuck way in the back because you never figured it out? You need this **guaranteed**, audio course. Instead of digging through manuals, you and your family will learn by listening. Faster. Easier. And better. Money-back if not delighted — return in 15 days for a full refund. Specify VIC-20 **($19.95)** or Commodore 64 **($29.95)**. (+ $2 shipping). Credit card holders call

800-222-3547

In IL, call 312-790-0517.

FlipTrack® training tapes from

EFFECTIVE SOLUTIONS
Dept. 100, 15 E. Madison
Lombard, IL 60148

AT LAST!

EASY BAESIC™ DISK

For Commodore 64™ Users

FEATURES

- *Three Dimensional Graphics*
- *Sound & Music Commands*
- *Sprite & Game Programming*
- *48 Additional Basic Commands*

ONLY $29.95

ADD $2.50 SHIPPING & HANDLING
UTAH — ADD SALES TAX
• CHECK, MO, VISA or MC
(SEND CREDIT CARD EXP. DATE)

CLONEWARE

CLONEWARE
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

#%@#* THAT DRIVE!

DOES THE COMMODORE 1541 DISK DRIVE MANUAL LEAVE YOU TALKING TO YOURSELF?

Let us show you, on Video Tape, how to utilize your 1541 with the Commodore 64. Step by step we take you through the various commands, including setting up files, DOS, and much more.

As a bonus, we include how to connect your 64 to your VCR, as well as a few handy programming tips.

To order, send $29.95 + $1.05 S+H, by check, MO, or credit card to:
(Canadian customers send $44.50 for all costs, including duty)

Maine Residents add 5% sales tax
Specify VHS or BETA format

THE FUTURE STORE
216A MAIN ST.
CALAIS, ME. 04619

PROTECT YOUR EXPENSIVE EQUIPMENT FROM DUST, LIQUIDS WITH A CROWN PROTECTIVE

COVER

- CUSTOM MADE TO FIT
- HEAVY 32 oz VINYL
- ANTI-STATIC
- SOFT LINED
- CHOICE of COLOR, TAN or BROWN

Covers for:

VIC20/C-64	7.00
C-1541 D/DRIVE	8.00
C-1525 PRINTER	10.00
DATASETTE (New)	5.00
DATASETTE (Old)	5.00
GEMINI 10/10X PRINTER	13.00
GEMINI 15/15X PRINTER	16.00
EPSON MX80 PRINTER	11.00
EPSON MX100 PRINTER	14.00
APPLE IIe KEYBOARD	7.00

Order by stating name and model of equipment for cover desired. Choice of color: TAN or BROWN. Enclose check or M.O.+1.50 shipping. Calif. Res. include 6.5% State Tax.

COVERS NOT NAMED ABOVE WILL BE FABRICATED TO YOUR SPECS. SEND YOUR REQUIREMENTS FOR LOW PRICE QUOTES.

CROWN CUSTOM COVERS
9606 SHELLYFIELD RD.,
DOWNEY, CA 90240

Want to get more out of your Commodore 64?

We can help!

Reviews - Ratings
Low Prices

Write for Free Catalog:

General Software Store
1146 Eastwood Drive
Fayetteville, AR 72701

ATTENTION C-64 DISK USERS
ORGANIZE NOW!
WITH THE
MASTER-DIRECTORY SUPPORT SYSTEM

With MDSS you can organize your disk files onto 1 master disk. Maintain sorted master-directories of your files categorized by business, education, recreation or any other category you choose. Print single or multiple copies of master-directory listings, disk jacket indexes or individual disk labels. MDSS can locate your "lost" disk files too!

MDSS is fully menu driven and very user-friendly. Includes an easy to follow instruction manual.

Requires C-64 and 1540 or 1541 disk drive. Printer optional.

Send check or money order for $16.95 to:

SUNSHINE SOFTWARE
P.O. BOX 831
DEARBORN, MICHIGAN 48120
MICHIGAN RESIDENTS ADD 4% SALES TAX

SIGN OF THE SPHINX

Something new from **Werewolf Software** — A world of enigmatic artifacts, disturbing visions, peculiar people. From an abandoned subway station to the Eternal Black Mass and beyond, you discover traces of forgotten knowledge which may lead you to the Crimson Altar and its final secret.

A morbid, suprising adventure, Sign of the Sphinx is disk-based to use memory more efficiently. If you appreciate detail, if you have a taste for the bizarre, then you should investigate the activity at the Sign of the Sphinx.

Text adventure on disk for the Commodore 64. $15 plus $1 shipping. Calif. residents add tax.

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

Successful Delivery!
Chromazone's New Arrival!

BRINK JINK

Maneuver through the mazes to make a million . . . if you dare!

Graphics for C-64, disk only, joystick required.

Send $29.95 in check or money order to:

Chromazone Software
P.O. Box 7325
San Jose, CA 95150-7325

CA residents add $1.95 tax

The Intelligent Software Package
For $35, you get all this on one disk:

DATA BASE: A complete fixed-record-length data base. Sort on any key, select using full logical operators on any key or keys, perform numeric manipulation on fields. All fields in a record fully customizable. Screen editing for records. Can be used for accounts-receivable, inventory control, or as an electronic rolodex. If you use your Commodore for nothing else, this program will justify its expense.

WORD PROCESSOR: A full-featured word processor: very fast file commands (including disk file catalog), screen editing, string searches, full control over margins, spacing, paging, and justification (all commands imbedded in text). A very powerful, easy-to-learn program. Includes a program interfacing W/P with DATA BASE to create custom form letters.

SPREADSHEET: Turns your Commodore into a visible balance sheet. Screen editing. Great for financial forecasting.

BASEBALL MANAGER: Compiles batting statistics for a baseball or softball league. Generates reports on a player, team, or the entire league (including standings).

All programs will load and run on any and every Commodore computer having a minimum of 10k RAM; all programs fully support tape, disk, and printer. Any two programs on cassette, $20. Price includes shipping within USA and Canada; Calif. residents add 6%. For orders over 10 in quantity, deduct 35%.

Since this ad is the catalog, no response to inquiries will be made; however, documentation for any one program may be purchased separately for $2 postpaid (deductible from later order). Thank you.

William Robbins, Box 3745, San Rafael, CA 94912

D AND L COMPUTERS

COMMODORE-64™ PUBLIC DOMAIN SOFTWARE

28 DISKS TO CHOOSE FROM

- GAMES
- EDUCATIONAL
- ART & MUSIC
- UTILITIES
- PERSONAL FINANCE
- HOME APPLICATIONS

MUCH MORE

Order Demo Disk For $8.95 and Receive "Disk-Locate" FREE

Disk-Locate Stores and Retrieves Directory Info From Over 150 Disks

Discounts on All Popular Software

Catalog Sent on Request

California Residents Add 6½% Sales Tax on Orders

D AND L COMPUTERS
7166 GATESHED WAY
CANOGA PARK, CA 91307
(818) 710-9874

SPECIAL

$15.00 ARITHMETIC TAC TOE

2 LEVELS OF DIFFICULTY

4 MATH FUNCTIONS

LEARN AND REINFORCE MATH SKILLS

Joy of PROGRAMMING
51 BORO E. STBG., PA 18301
Coding In Vic-20 Basic

VIC 20 / COMMODORE 64

CRAZY CONVEYORS-an exciting action-packed game with multi-color sprites, custom characters in 11 different colors for building blocks, ladders, fire poles, rotating pulleys, moving conveyors and bonus boxes; entertaining sound; high score history, with full names of 10 champions; action pause; start play at screen of your choice; joystick or keyboard; machine language. Also **Screen Creator** to expand the game disk and extra disks to virtually unlimited screens, & **CRAZY CONVEYOR action** to entertain and challenge the most skillful player.
List Price: $39.95 Commodore 64 (Disk) Our Price: $29.95

DUNGEONS and DEMONS-create your own characters and explore a dungeon with 12 levels and 1200 rooms. Do battle with any of over forty types of monsters. Find and trade treasure chests containing gold for superior weapons and armor. The ultimate goal is to reach the deepest dungeon level and find the Golden Chalice. As your character increases in strength and wisdom, you are allowed the option of saving, to disk, your character and his position in the dungeon. Game contains 3 dimensional graphics, multi-color sprites and excellent sound.
List Price: $29.95 Commodore 64 (Disk) Our Price: $21.95

GOOMBAHTZ-a humorous strategy game of 6 dice with 46,656 outcomes. 1-6 players. Match your skill with friends and/or COMPU the 64, who expresses feelings in phrases and sounds.
List Price: $22.95 Commodore 64 (T) or (D) Our Price: $16.95

ADVENTURE ISLAND-Intriguing adventure game. You are shipwrecked on a vast island of riches with an underground complex. Escape a host of terrifying pitfalls with great wealth, your skin only, or remain forevermore!
List Price: $22.95 C64 (T) (D); VIC + 8K (T) (D) Our Price: $16.95

Also Available:

		List	Our Price
DISK DIRECTORY MANAGER	C64 (D), VIC + 16K (D)	$26.95	$19.95
INVESTMENT PORTFOLIO MGR.	C64 (T) or (D)	$19.95	$14.95
DUNGEONS (adventure)	VIC + 16K (T) or (D)	$19.95	$14.95
RIDGE RUNNER (space) 100% ML	VIC unxp. (T) or (D)	$19.95	$14.95
PAK ALIEN (maze) 100% ML	VIC unxp. (T) or (D)	$19.95	$14.95

BYTES and BITS
524 E. Canterbury Ln.
Phoenix, AZ 85022
(602) 942-1475

Specify tape or disk, VIC or C-64
Check, money order or C.O.D.
Add $2.00 for postage & handling
Additional $3.00 for C.O.D.

SOFTWARE PROTECTION HANDBOOK

This book "BLOWS THE LOCKS OFF" protected DISKS, CARTRIDGES, and TAPES! Protection "secrets" are clearly explained along with essential information and procedures to follow for breaking protected software. An arsenal of protection breaking software is included with all listings, providing you with the tools needed! Programs include high speed error check/logging disk duplicator... Disk picker... Disk editor... Cartridge to disk/tape saver and several others for error handling and advanced disk breaking. The cartridge methods allow you to save and run cartridges from disk or tape! The tape duplicator has never been beaten! This manual is an invaluable reference aid including computer and disk maps, as well as useful tables and charts.

C64 Book only.....$16.95 US
Book & Disk of all programs.....$29.95 US
Vic 20 book.....Cart. & Tapes only....$9.95 US

THIS MANUAL DOES NOT CONDONE PIRACY
DISTRIBUTING COPIED SOFTWARE IS ILLEGAL
PSIDAC, 7326 N. ATLANTIC, PORTLAND, OR 97217
-CHECK OR MONEY ORDER ONLY-

C-64™ & VIC-20™

SUPER TYPEWRITER
The mini word processor you've wanted . . .

FEATURES:
- ★ Changeable line width up to 80 characters
- ★ Automatic margin setting
- ★ Automatically centers each additional copy
- ★ Upper and Lower Letters
- ★ No more broken words with use of automatic carriage return
- ★ Edit Text

All Poorhaus Programs user accessible for learning or adding personal touch. Simple to use. Load and follow instructions within programs.

Super Typewriter **$24.95**
Home Inventory **12.95**
Check Register **19.95**
Black Jack **9.95**
Loan Analyzer **9.95**

Some VIC-20 Programs may need memory expansion.

POORHAUS SOFTWARE
P.O. Box 10782, Yakima, WA 98909
(509) 966-8461
SPECIFY TAPE OR DISK
MC, VISA, AND CHECKS ACCEPTED

KEEP THE DUST OFF & PUT THE ELEGANCE ON WITH GENUINE LEATHER DUST COVERS

Enjoy the look of soft elegance, along with durability that only real leather can offer. Don't settle for less than the best. Order singly or as a matched set, custom fitted to your Commodore computers.

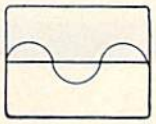
ORDER TODAY

QTY.		AMT.
_____	Computer Cover/14.95	_____
_____	1541 Disk Cover/13.95	_____
_____	Dataset Cover/9.95	_____
	TOTAL $	_____

□ Check or Money Order enclosed.
□ Visa; □ Mastercard Exp. Date _____
Card No. _____
Signature _____
SHIP TO:
Name _____
Address _____
City _____ St./Zip _____

S & S ENTERPRISES
P. O. BOX 111
HOT SPRINGS, S.D. 57747
Dealer Inquiries Invited

C-64
Color Digital Oscilloscope *$159*



- Add-on peripheral converts C-64 to 4-channel digital oscilloscope.
- 5" x 7" box plugs into user port.
- 2 MHZ sampling rate.
- 500 KHZ analog bandwidth.
- Simple menu driven operation.
- Stores/retrieves waveforms on disk or cassette.
- Gains and timebase controlled from keyboard.
- Spectrum analyzer software available.
- Specify disk or cassette software.

Send: Rapid Systems Inc.
5415 136th Pl. S.E.
Bellevue, WA 98006

call: 206-641-2141

$159

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).

J&H COMPUTERS DEPT. 123G
5056 NORTH 41st STREET
MILWAUKEE, WISCONSIN 53209
PHONE (414) 461-9941

• 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 (802) 453-3122

Dealer Inquiries Invited

CONVERSE WITH YOUR COMPUTER

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

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

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

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

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

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

1. Protected Version $25
(Protected Version can be run but not listed or modified)
2. Un-protected Commodore 64 BASIC Source Version $45
(Source Version can be listed and modified as well as run)

Both versions include a six page user manual.

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

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

COMMODORE 64-

DISKMIMIC 5™ @ $49.95

- Backs up virtually all existing disks for Commodore 64™, including COPY PROTECTED versions. ALL AUTOMATICALLY.
- Supports one/two 1541™ Drives.
- Don't be without back-up.

DISKMIMIC™ @ $24.95

- Back-up your Commodore 64™ programs with SAVE YOUR DRIVE disk formatter.
- Hi-speed, Hi-buffer (190 Blocks).
- Extends life of 1541™ Drive.
- Single drive back-up.
- Selects tracks or backs up entire disk.

FAST • FAST • FAST

SPECIAL PACKAGE

Diskmimic™ & Diskmimic 5™ @$64.95

A.I.D. Corp.
4020 Hempstead Turnpike
Bethpage, New York 11714
(516) 731-7100

Diskmimic™ & Diskmimic 5™ is a trademark of A.I.D. Corporation
Commodore 64™ & 1541™ is a trademark of Commodore Electronics Ltd.

EPROM PROGRAMMER
FOR
PET[1] • COMMODORE-64[1] • VIC-20[1]

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 32 K BYTES (incl. some E² 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 - LIF incl. $59.50 •

[1] PET, COMMODORE 64, & VIC 20 are trademarks of CBM, INC.

(215) 256-6933 **DAZCO** (215) 256-6933
Box 267 Lederach, Pa. 19450
• +$2. ship. & Pa. res. add 6%

ELECTRONICS
Circuit Design and Analysis

TEST CIRCUITS BEFORE YOU BUILD THEM!

ANALYSIS PACKAGE INCLUDES:
Two powerful programs to analyze the frequency and phase response of almost any circuit configuration of Resistors, Capacitors, Inductors, Op-Amps, FET's and NPN Transistors.

CIRCUIT ANALYSIS PACKAGE...$29.95

DESIGN PACKAGE INCLUDES:
Programs to design Active Filters, Passive Filters and Attenuators, plus...Ohms-Law, Resonance, Wire Gauge, Standard Resistor Value Solutions & more...

CIRCUIT DESIGN PACKAGE...$19.95

EACH ON DISK FOR THE COMMODORE 64 FROM:

Nth DIGIT SOLUTIONS
3243 Arlington Avenue, No. 195
Riverside, CA 92506

NAME
ADDRESS
CITY
STATE ZIP

• CALIFORNIA RESIDENTS ADD 6 PER CENT SALES TAX

ADVERTISERS INDEX

CARDCO Letter Quality Printers

"Commodore-ready", PLUS . . .

An excellent line of Letter Quality PRINTERS are ready for you and "Commodore-ready", PLUS . . . compatible input for PC, PC jr., TRS-80 and a host of other personal computers.

All three CARDCO PRINTERS are true letter quality printers, bi-directional with continuous printing capability, for computers with centronics parallel output, of excellent quality, standard friction-feed, designed in attractive neutral to blend with any decor in your office, home or business. The fine letter quality is suited for personal or business use, ideal for word processing applications including subscript, supersubscript and bold-typing.

Priced attractively and much lower than you expect.

- **Model LQ/1** . . . 13" carriage, daisy wheel type, printing speed 14 cps.
- **Model LQ/2** . . . 8½" carriage, drumhead design, printing speed 12 cps.
- **Model LQ/3** . . . 11" carriage, daisy wheel type, printing speed 13 cps.

Write for illustrated literature and prices or see CARDCO Computer Accessories and Software wherever Computers are sold.

cardco, inc.

300 S. Topeka Wichita, Kansas 67202 (316) 267-6525

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

Commodore Software– The Best Game in Town.

...Take on the world, toughen up your trigger finger and fire away...

Commodore is the best computer value in town...at home, at school and at work... with our exciting, easy to use, inexpensive VIC 20 and C64 computers.

We're fast becoming the best game in town when it comes to entertainment for the whole family...and at affordable prices.

THE BEST ARCADE IN TOWN can be in your own home with our exciting, faithful reproductions of the best of Bally Midway arcade games. Our **Kickman,** (which just received a coveted "Electronic Games" award for an arcade translation) lets you steer the unicycle to catch the falling objects, as they fall quicker and quicker!!

Gorf, Lazarian, and Omega Race give you the best in classic space action against the one-eyed leviathon, the droids or the evil Empire.

In **The Wizard of Wor** you attempt to defeat the Wizard and the Warriors, fighting your way through to the end. With the new Commodore "MAGIC VOICE"... It talks back to you too!!

You commandeer the fleet at sea with our version of **Seawolf,** and become the master tactician as you battle "it out" with enemy fleet.

Clowns and **Blueprint** round out our arcade entertainment package to keep your fingers nimble and your mind in gear.

commodore COMPUTERS

First In Quality Software

See your local dealer now... He's got the best game in town. Just for you.

www.commodore.ca