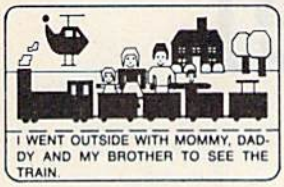
THOUGHTFUL EDUCATIONAL PROGRAMS

BECi

C-64 VIC 20 ATARI

CREATIVITY SERIES



MY BOOK Ages: 4 and up

Your child makes his or her own book!

Using only a joystick, your youngster picks objects from the library (princesses, pirates, houses, vehicles, people, etc.), colors them and arranges them in the picture!

Older youngsters can then type in a story-line or caption, and the page is saved to disk.

Page after page can be saved. But the fun isn't over yet! Your child can PRINT the book on almost any printer capable of producing Commodore graphics!

The first program in BECi's new Creativity Series. Available soon for the Atari.

C-64 DISK: $34.95

CHILD DEVELOPMENT SERIES

"The goal of Boston Educational Computing is to provide owners of the most elementary computer systems with educational software that can be used easily by those with little knowledge of computing.

"In its Child Development Series, BECi (pronounced Becky) meets this goal."*

*Computes!'s Gazette, January 84

ALPHA-BECi Ages: 2 and up

An alphabet program with 26 screens, each featuring a capital and small letter and an object.

"For a child, watching the colorful objects appear on screen is like opening a present."*

(VIC-20 only).

NUMER-BECi Ages: 2 and up

Number identification and color and shape grouping. Five levels and adjustable timing.

ADD/SUB Ages: 5 and up

Addition and subtraction. Up to four digits. Optional objects, carries and borrows, decimal points and hints. Answers entered from right to left, one digit at a time.

MULT-BECi Ages: 7 and up

Multiplication. Up to four digits in multiplier and multiplicand. Answers entered from right to left, one digit at a time.

TAPE: $19.95 **DISK:** $24.95

If ordering directly from BECi, add 5% (or $2.00 minimum) for shipping.

BOSTON EDUCATIONAL COMPUTING, INC.

Dept. G
78 Dartmouth Street
Boston, MA 02116
(617) 536-5116

Dealer Inquiries Invited

MasterCard VISA

Charge cards and phone orders accepted.

Hats off to a machine language teaching system that works.

Machine language programming isn't easy, but you don't have to be a genius to learn it. Despite what you may think after getting lost in umpteen "How to program the 6502" books. Let your Commodore 64 teach it to you.

The Visible Computer: 6502 is an award-winning blend of text and software that thousands have used to master the elusive skills of machine language.

It's an animated simulation of the 6502 microprocessor that lets you see with your own eyes how the 6502 works. You'll be using it as a debugging tool for years to come.

It's a tutorial. The 150 page manual is more than just instructions on running the simulator — it may just be the best book on machine language ever written.

It's 30 demonstration programs you'll work through with the 6502 simulator, from simple register loads to advanced graphics programs.

The Visible Computer: 6502

Send me The Visible Computer: 6502 for Commodore 64 (requires disk drive). I've enclosed $39.95 plus $2.50 postage and handling.

☐ Check or Money Order ☐ Visa ☐ Mastercard

Name

Address

City/State/Zip

Credit Card No. Exp.

CG

Software Masters™
3330 Hillcroft, Suite BB
Houston, Texas 77057
(713) 266-5771

File Copier

Martin Engert

"File Copier" is a BASIC utility that lets you transfer files from one disk to another using a single drive without worrying about starting addresses or machine language. For the VIC-20 and Commodore 64.

"File Copier" can help those who want to copy sequential or program files from one disk to another, but have only a single disk drive and no machine language monitor. Since the program is written in BASIC, it's a bit slow. But one advantage of this program over a machine language monitor when transferring machine language programs is that you don't have to know the initial address or length of the program to be transferred.

File Copier works on both the VIC-20 and Commodore 64. The program first resets the top of BASIC pointers to reserve 1K of memory for itself. The remaining memory is used to store your file temporarily. VIC users should make sure enough memory is available for this purpose before running the program. Any amount of expansion memory can be added if necessary. Each byte of your file is then read from disk using the GET# command and POKEd into free memory. Then you insert the new disk and the program writes these bytes onto it using PRINT#. After the file is copied, the top of BASIC pointers are restored to normal.

Screen instructions are provided within the program for easier use.

File Copier

```
10 POKE251,PEEK(52)                    :rem 49
20 POKE52,PEEK(44)+4:POKE56,PEEK(52):CLR
                                       :rem 89
30 PRINT"{CLR}RUN THIS PROGRAM TO"
                                       :rem 175
40 PRINT"COPY A PROGRAM OR"            :rem 106
50 PRINT"SEQUENTIAL FILE FROM"         :rem 133
60 PRINT"ONE DISK (THE SOURCE"          :rem 30
70 PRINT"DISK) TO ANOTHER (THE"        :rem 73
80 PRINT"DESTINATION DISK)."           :rem 253
90 PRINT"INSERT SOURCE DISK."           :rem 57
100 M=256*PEEK(52)                     :rem 191
110 OPEN15,8,15                        :rem 32
120 PRINT"WHAT IS THE NAME OF"         :rem 203
130 PRINT"THE FILE OR PROGRAM":INPUTF$
                                       :rem 83
140 T$="P":PRINT"WHAT IS THE FILE TYPE"
                                       :rem 252
150 PRINT"(P FOR PROGRAM, S FOR"       :rem 68
160 PRINT"FILE)"                       :rem 177
170 INPUTT$                            :rem 160
180 OPEN2,8,2,F$+","+T$+",R"           :rem 128
190 INPUT#15,E,E$,X,X:IFE<>0THENPRINTE$:C
    LOSE2:GOTO120                      :rem 134
200 GET#2,A$:IFA$=""THENA$=CHR$(0):rem 90
210 POKEM+J,ASC(A$):J=J+1:IFST=0THEN200
                                       :rem 66
220 CLOSE2                             :rem 60
230 PRINT"INSERT DESTINATION"          :rem 125
240 PRINT"DISK AND PRESS {RVS}RETURN"
                                       :rem 228
250 PRINT"TO COPY."                    :rem 116
260 GETC$:IFC$<>CHR$(13)THEN260         :rem 6
270 PRINT"PRESS {RVS}RETURN{OFF} IF YOU"
                                        :rem 7
280 PRINT"WANT TO KEEP THE NAME"       :rem 111
290 PRINTF$                            :rem 146
300 INPUT"FILE NAME ";F$                :rem 77
310 OPEN2,8,2,F$+","+T$+",W"           :rem 128
320 INPUT#15,E,E$,X,X:IFE<>0THENPRINTE$:C
    LOSE2:GOTO300                      :rem 129
330 FORK=0TOJ-1:PRINT#2,CHR$(PEEK(M+K));:
    NEXT                                :rem 7
340 CLOSE2:CLOSE15                     :rem 85
350 POKE52,PEEK(251):POKE56,PEEK(251):CLR
                                       :rem 145
```

Food for Thought.

Real computing at appetizing prices.

Inventory 64.™ An easy-to-use menu-driven program that lets you track complete and complex inventories. Pinpoint locations, vendors, units of measure. Update prices, reorders, sales, entire stocks. A powerful tool for your home computer at only $29.95*, on disk.

64 PAK.™ A self-teaching sampler package of 10 menu-driven programs. Ideal as an introduction to computing. Or compute your life expectancy, your mortgage payments, even learn the language of the deaf, plus a lot more. Only $19.95*, on tape; $24.95*, on disk.

PractiFile.™ A fully professional file manager for your Commodore. PractiFile can handle mailing list entries by the thousands. You can change records, numbers, methods of filing, and a great deal more. Plus, it's easy to use. Only $54.95*, on disk.

PractiCalc™ 64. The most powerful spreadsheet available for a home computer—at the least cost. Track investments, inventories, expenses. Project profits. Make charts. Keep mailing lists. Sort alphabetically or numerically, instantly. An incredible value at $49.95*, tape; $54.95*, disk.

Total Health.™ The fun way to keep fit. Stay on top of your daily intake and outgo of calories. Balance your daily diet and graph your progress—all tailor-made to your body, past, present, and future. It's like having a health spa in your home for only $24.95*, on tape; $29.95* on disk.

Now you can do real, honest-to-goodness computing at computer-game prices. Computer Software Associates brings you hard-working software that makes your work easy. (While it's easy on your budget.) All programs are easy to work with right on the screen, from self-teaching instant software that tracks your vital statistics to potent programs to track inventories and profits.

Of course, if it's games you want, we've got them too. But if you're looking for home computer power with real brainpower, look no more. Now you can play for keeps.

COMPUTER SOFTWARE ASSOCIATES, INC.

Exclusively distributed by Micro Software International, Inc. • The Silk Mill, 44 Oak St., Newton Upper Falls, MA 02164 • (617) 527-7510

Appending Sequential Disk Files

John S. Winn

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.

Sequential files are lists of things—phone numbers, addresses, names, or other data—kept on a disk. They are similar to tape files (which are also sequential) because they keep the information in the same order it is entered. The first item written into the file will be the first one to come out when a program reads the file.

It's easy to write BASIC programs to construct such lists. But how do you add new information to the end of a sequential file?

The 1541 disk drive user's manual suggests two possible solutions. The first is to read the whole file, add the new data, and write the file out again. The longer the file, the more time it takes, which can be rather annoying. Or, perhaps you could switch to random access files, at a considerable cost in programming effort.

A simpler method is available, but it's not mentioned in the user's manual, and thus is not widely known. The operating system on Commodore's PET and CBM series (with BASIC 4.0) includes an APPEND command. The VIC and 64 versions of BASIC do not recognize this command, but your 1541 disk drive does. And it's simple to use.

Normally, when you want to write a sequential file, you use OPEN 1,8,8, "filename,S,W" (the S means Sequential and W means Write). To read the file, replace the W with an R. The method for appending uses a similar form. If you want to add to a file which already exists, just use OPEN 1,8,8, "filename,A" (A for Append).

The following short programs demonstrate this useful command. First we'll write a new file containing the first ten letters of the alphabet.

```
10 OPEN1,8,8,"ALPHABET,S,W"
20 A$="ABCDEFGHIJ"
30 FORJ=1TO10
40 PRINT#1,MID$(A$,J,1);
50 NEXT
60 CLOSE1
```

RUN the program. The red light on your drive should blink on, indicating the file is being written. Then, append to this file the next ten letters of the alphabet, using the A (Append) in the OPEN statement. Change lines 10 and 20:

```
10 OPEN1,8,8,"ALPHABET,A"
20 A$="KLMNOPQRST"
```

RUN the program again and the new data will be added to the file. Now read the file to doublecheck that the technique worked. Type NEW and enter this short program.

```
10 OPEN1,8,8,"ALPHABET,S,R"
20 FORI=1TO20
30 GET#1,A$
40 PRINTA$
50 NEXT
60 CLOSE1
```

UNDERWARE

Software that's priced UNDER the competitions'

COMMODORE 64

Just for You!

PERSONAL FINANCE

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

WORD PROCESSING

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

SPREADSHEETS

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

DATABASES

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

HOME APPLICATIONS

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

ART & MUSIC

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

EDUCATION

SPINNAKER LEARNING FUN

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

DESIGNWARE (DISKS)

Crypto Cube ages 8 & up	$29
Spellicopter ages 6 & up	$29
Creatr. Creator ages 4-8	$29
Math Maze ages 6-11	$29
SpellakaZam ages 7-14	$29
SpellaGraph ages 7-14	$29

SCHOLASTIC WIZWARE (DISKS)

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

COMMODORE 64 Cont'd.

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

LANGUAGES & UTILITIES

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

BUSINESS SOFTWARE

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

GAMES

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

ELECTRONIC ARTS

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

FREE DISKETTE with each purchase of electronic arts software

COMMODORE VIC 20

PERSONAL PRODUCTIVITY

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

EDUCATIONAL SOFTWARE

HESWARE SPINNAKER

KinderComp (CT)	$23
Story Machine (CT)	$23
Face Maker (CT)	$23
Kids On Keys (CT)	$23
Alphabet Zoo (CT)	$23
HesWare Turtle Graphics (CT)	$26
Creative Pipes (C)	$19
Creative Spills & Fills (C)	$19
WizWare Turtle Tracks (C)	$19
WizWare Square Pairs (C)	$19

GAMES

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

PERIPHERALS

GRAPHICS TABLET

CHALKBOARD POWERPAD (C64 & Atari)	$79

Software Packages

Logic Master	$27
Leo's Links	$27
Programmers Kit	$19
BearJam	$25
KOALAPAD ATARI C64	$67

PRINTERS

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

MODEMS

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

MONITORS

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

COMMODORE DISK DRIVES

Concord Disk Drive	$279
Commodore 1541	$269

WICO JOYSTICKS

The Boss	$14
3 Way Grip Stick	$21

CARDCO

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

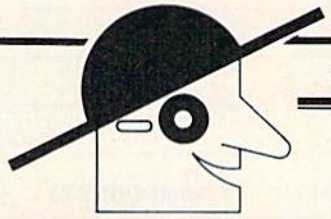
BLANK DISKETTES

MAXELL DISKS

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

VERBATIM

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



$uch A Deal

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

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

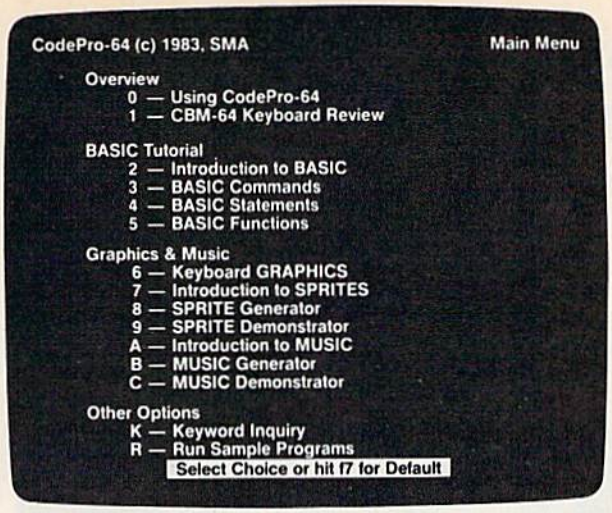
ORDERING & TERMS: Send cashier check, money order, personal/company checks allow 3 weeks bank clearance. VISA/MasterCard accepted. Provide phone number with order. SHIPPING: Software add $4.00 for first piece, add $1.00 each additional piece. Hardware add 3% or $10.00 whichever is greater. Returns must have authorization number (call 602-968-9128 for authorization number). All returned merchandise subject to restocking fee and **must** come with **all** original packaging. No returns allowed after 30 days from shipping date. Prices are for cash; VISA and MasterCard add 3%. Prices subject to change without notice. All products subject to availability from manufacturers and/or suppliers. All prices in U.S. dollars. We pay shipping on backorders.

When you RUN this program you should see all 20 letters on the screen. It works.

There are a couple of things you should be aware of when you use this new command. First, you cannot use the A command to open a file for the first time. The file must already exist—you have to use OPEN1,8,8"filename,S,W" before appending. If you want to, you can create an empty file with a CLOSE1 immediately after the OPEN statement. You can then use the A command within your BASIC program.

Second, if you use a lot of appends, disk space can be used up more quickly than normal, due to a quirk in the append command. Think about the file created by the two programs above. It is short (containing only 20 characters) and should use only one block on the disk. But if you call up the directory (LOAD"$",8 followed by LIST), you will notice that ALPHABET uses two blocks. This is because the A command puts the new data at the beginning of a brand new block. Even if you append only one item, it will use up a whole block. If you append often, you may start to lose free disk space.

The answer to the second problem is to read in the whole file, scratch the old sequential file, and write a new one, a process we originally wanted to avoid. But at least you won't have to do it every time you append.



A new concept in interactive visual learning.

CodePro-64™

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

SEE PROGRAM EXECUTION

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

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

EXTENSIVE TUTORIAL

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

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

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

Once you have practiced and mastered the BASIC language elements you move on to more advanced concepts. You learn about sprite and music programming.

SPRITE GENERATOR & DEMONSTRATOR

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

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

MUSIC GENERATOR & DEMONSTRATOR

Our *Music Generator* and *Music Demonstrator* will provide hours of instruction and creative enjoyment. From the beginning of your instruction you can *compose simple tunes* on the screen using the generator. Once you've completed a composition you can *save the tune* and its associated SID parameters to a diskette file. Our music sample program can be used alone or incorporated into your own programs to *read the saved music file* and replay your songs.

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

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

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

OUR GUARANTEE

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

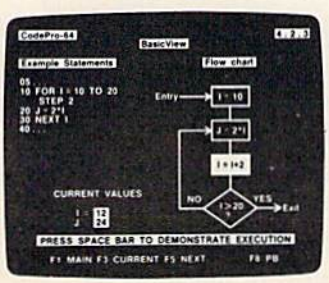
HOW TO ORDER

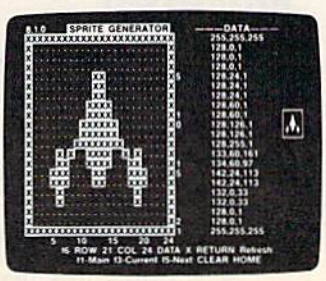
Order your copy of CodePro-64 today by mail or phone. Send only $59.95 plus $3.00 shipping and handling to:

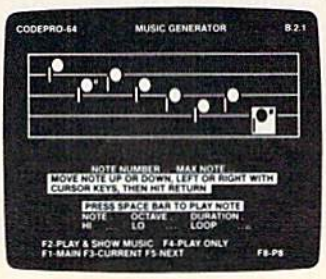
SYSTEMS MANAGEMENT ASSOCIATES
3700 Computer Drive, Dept. G-1
Raleigh, N.C. 27609

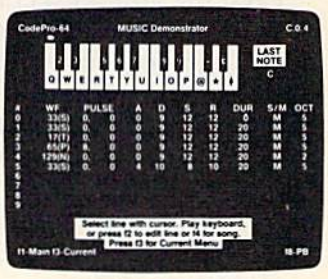
Available on diskette only. MasterCard/-VISA accepted. For faster service on credit card orders, call toll free 1-800 SMA-RUSH. (1-800-762-7874). *Dealer inquiries invited.*

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









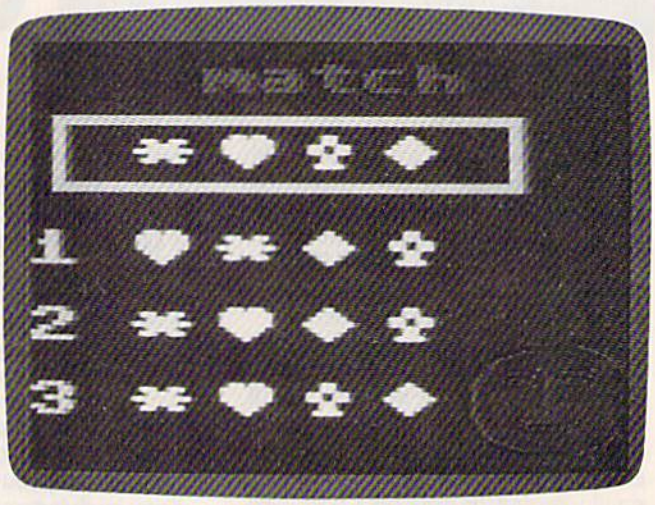
SMA SYSTEMS MANAGEMENT ASSOCIATES

Dan Carmichael, Assistant Editor

Software For The VIC

In this month's column, we'll look at some new games and educational programs for the VIC-20.

Commercial software for the VIC-20 has decreased as of late. That is not to say that the popularity of the VIC has also lessened. Nor does it indicate that support for the VIC is waning. With the introduction of the Commodore 64 and other computers, a greater area is being covered by software companies. However, there are still many good games, educational programs, and applications being offered for the VIC. We'll look at a few of the better ones here.



The Match option from Kindercomp.

Educational Programs

Type Attack, from Sirius Software, is a novel typing tutor program. Usable by all ages, it teaches letter recognition and keyboard familiarity. Because the typing speeds can be set from 1 to 99 words per minute, *Type Attack* can help improve the techniques of novice and expert typists alike.

The program contains three lesson options: Character Attack, Word Attack, and Lesson Maker. Character Attack teaches character recognition, Word Attack teaches word recognition, and the last option allows you to set up your own lessons.

Type Attack offers more than many other typing tutors in that it plays like a game. You learn and have fun at the same time. Based on a Space Invaders theme, the letters and words drop from the sky. The player blasts them (and defends the earth) by pressing the appropriate key or keys. The action can become so fast and furious that even the youngest child's attention can be held. If your software needs include a typing tutor, I strongly recommend this one.

Kindercomp, from Human Engineered Software, is a collection of learning games for children ages 3 to 8. Four of the games are Names, Sequence, Letters, Match.

Names allows you to type in the child's name, which is then displayed on the screen a number of different ways.

Sequence presents a line of numbers in sequential order followed by a question mark. The child is then asked to supply the next number in the sequence. A right answer is rewarded with a smiling face, and three correct answers in a row produce a colorful screen display with sound.

Letters displays a single letter on the screen, then asks the child to type the same letter on the keyboard. Right answers are rewarded with colors and sounds.

Match familiarizes the child with shapes and patterns. A pattern of three shapes appears in a box. The child is then asked to identify the pattern by matching it with one of three choices.

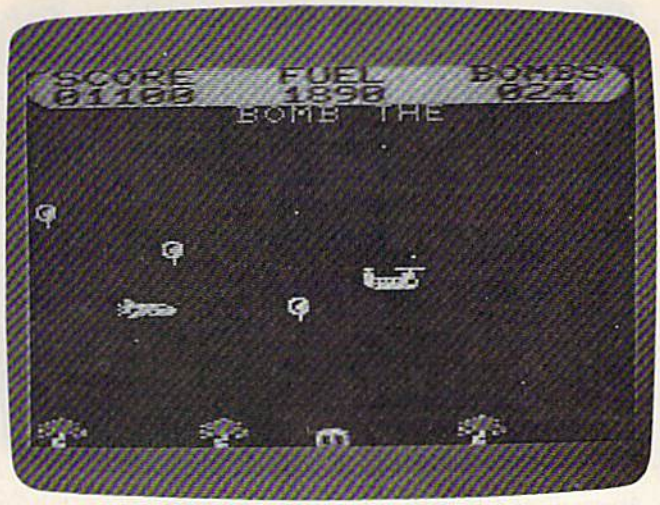
All of the games feature good color and sound to hold the child's attention.

A Full-Featured Word Processor

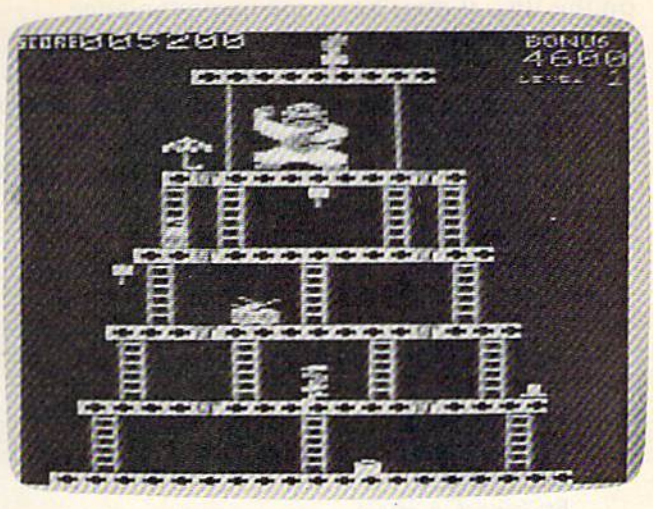
Write Now!, from Cardco, is a word processor for the unexpanded VIC. It contains many of the options you'd expect to find in word processors designed for computers more powerful than the VIC.

The program is on cartridge, so there's no handling of fragile diskettes, or lengthy tape cassette loads. Other advanced features include:

- Multiple line headers and footers.
- Ability to save text to tape or disk.



Skyblazer *offers good graphics and fast action.*



An arcade classic for the VIC, Donkey Kong.

- Options for sending special printer codes.
- The ability to recall frequently used passages and insert them into your text.
- Text scrolling up or down.
- Easy insert and delete modes.
- Block commands including copy, move, and delete.
- Global search, which allows you to find and replace any string.
- Optional page numbering at the top or bottom of the page.
- Instruction booklet and a 30-minute audio cassette instruction course.

All you need is a disk drive or cassette (if you wish to save the text).

Games

There seems to be no shortage of games on the market for the VIC. Some of them bear no mention. However, there are many that play well and offer hours of challenging fun. Here are a few of them:

Skyblazer is a "defender" type of game. It offers smooth horizontal scrolling plus good graphics. And there's plenty of action.

The object of the game is to successfully complete the five missions, or game levels. The first level puts you on a bombing run to destroy enemy radar. Once this is destroyed, your next missions include attacking the enemy tank and ICBM defenses. After you've destroyed the primary defenses, the final challenge is to destroy the enemy headquarters. And that is not easy.

Skyblazer contains good graphics and sound. Also included are options to pause the game, or adjust the positioning of the screen display. Available on cartridge, all that is needed is your VIC and a joystick.

Donkey Kong, from Atarisoft, is a relatively accurate version of the arcade classic. The object of the game is to guide Mario through the steel girders and rescue his girlfriend from the clutches of a gigantic gorilla.

The game offers four different levels of play. While climbing upward, you must avoid the rolling barrels, firefoxes, mad springs, and cement piles. Grabbing the magic hammer allows you to smash the rolling barrels.

The game is challenging, and the colors, sound, and smooth graphic animation make it a fun game for the VIC. *Donkey Kong* is available on cartridge and requires a joystick.

Lazer Zone is a fast-paced shoot-'em-up game. The object is to fend off the attacking Warfiends of Zzyzax. You're armed with two rapidly firing cannons. The cannons are located at the bottom and the side of the screen and each moves and fires independently. Maneuver each cannon to shoot and destroy the invading Zzyzaxians.

Although the game is simple in theory, it's fun to play. The action is nonstop, and the sound effects are very good.

Lazer Zone is available on cartridge.

Capture the Flag, from Sirius Software, is a unique game in which you maneuver through a maze to capture your opponent's flag. But this is not the usual maze game. Instead of viewing the field of play from the top, you see it from ground level. This produces an interesting 3-D effect as you work your way through a maze of walls towering over your head.

Game options include a pause control and a feature which allows you to toggle the labyrinth maps on or off. You can also choose to play defense (protect) or offense (capture). You may also choose your opponent—a friend or the computer.

Although the 3-D screen display and the game action make this one of the best games I've seen

on the VIC, it does have one inconvenience. Dreary music plays throughout the game, and the instruction manual offers no way to switch it off. The music gets monotonous after a while, but you can always turn your volume down.

Capture the Flag is on cartridge and requires a joystick.

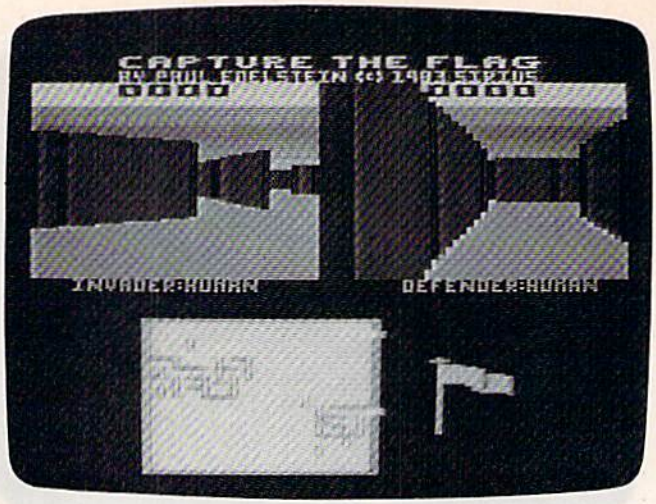
Type Attack and Capture the Flag
Sirius Software
10364 Rockingham Dr.
Sacramento, CA 95827
(916) 366-1195
$39.95 each

Kindercomp and Lazer Zone
Human Engineered Software
150 North Hill
Brisbane, CA 94005
(415) 468-4111
Kindercomp—*$34.95*
Lazer Zone—*$29.95*

Write Now!
Cardco Incorporated
313 Mathewson Ave.
Wichita, KS 67214
(316) 267-6525
$39.95

Skyblazer
Bróderbund Software
Entertainment Software Division
17 Paul Drive
San Rafael, CA 94901
(415) 479-1170
$34.95

Donkey Kong
Atarisoft
1265 Borregas Ave.
P.O. Box 427
Sunnyvale, CA 94086
(408) 745-2000
$44.95



An excellent 3-D effect in Capture The Flag.

Scroll 64

Peter Marcotty

A window can make a static screen more dynamic. This short machine language routine gives you control over screen scrolling from within BASIC programs.

Someone spots a tornado and reports it to the local weather bureau. Your television beeps and a warning moves across the bottom of the screen.

How would you create that effect on your 64? How do you make words scroll sideways?

Scroll Control And Windows

When you LIST a program, the screen fills quickly. As new lines appear, the screen scrolls from bottom to top (everything moves up a notch).

But there may be times when you want movement from top to bottom, or right to left. Or perhaps you want some information to stay in one section of the screen while everything else moves.

You need a screen window. Things in the window move, while everything else stays put. Some new computers, such as the Apple Macintosh, have built-in windowing.

"Scroll 64" won't turn your 64 into a Macintosh, but it can make your screen displays more dynamic.

Asteroid Belts And Invoices

There are many ways to creatively use screen windows and scrolling. For example, scrolling is common in certain types of video games. You drive a car on a road that moves toward you. Or your spaceship at the bottom of the screen has to shoot at descending asteroids. In addition to the action window, there is usually a section with information about your current score, remaining fuel, velocity, and so on. It would be confusing if your score moved with the asteroids, so the action of the game is put in a window. Your score goes somewhere outside the window.

Business programs can benefit from windows, as well. You might want a command line in an invoicing program, to remind the user of the various options. The window would cover all of the screen except the last line, which says "F1 = Help F3 = New F5 = Help F7 = Continue." Everything scrolls on the screen except the line at the bottom. Another possibility is a product list window in the corner of the screen. When the user of the invoice program wants to look up a product number, the window opens up and the list scrolls by.

Customizing Your Programs

Scroll 64 is a machine language program which goes into memory locations 49152–49528 ($C000–$C172). It does not use any BASIC RAM. The BASIC loader program reads the DATA statements and POKEs the numbers into memory. When the ML program is safe in memory, type NEW to get rid of the loader and clear RAM.

To use it, LOAD and RUN Scroll 64, type NEW, and then LOAD your own program. To activate it, simply SYS 49152. It scrolls once and returns to BASIC.

Or, if you prefer, you could build the BASIC loader into your program. Renumber the lines (starting at 60000, for instance), add a RETURN, and call it with a GOSUB at the beginning of your program.

Scroll 64 moves a certain section of the screen in a certain direction, along with the corresponding color memory. These memory addresses contain the pertinent information:

LOCATION	FUNCTION
49522	Direction
49523	Left Boundary
49524	Right Boundary
49525	Top Boundary
49526	Bottom Boundary
49527	Horizontal Wrap
49528	Vertical Wrap

Direction is the way in which the screen scrolls. To change it, POKE 49522 with one, two, three, or four (for left, right, up, or down respectively). The boundary values define the size of the window. Left and right boundaries can range from 0 to 39. Top and bottom must be between 0 and 24. When the program is first run, a five by five window goes in the top left corner.

The wrap values determine what happens to characters when they reach the edge of the window. You can make them disappear or wrap around to the beginning. POKE 49527 and 49528 as follows:

Number	Effect
0	Don't wrap around, leave a trail
1	Wrap around
2	Don't wrap around, erase trail

To activate the scroll window, SYS 49152. You can SYS over and over, changing the direction, boundaries, and wrap values as you wish. Note that when the ML routine is activated, whatever is in the window scrolls, but at all other times, the screen acts as it normally does.

Special Loading Instructions

Enter the program and SAVE it before you test anything. To put the ML into memory, type

```
RUN 60
```

The computer will take a few moments to complete the POKEs. As added insurance, there is a checksum routine built in. Type RUN and the values in memory are checked. If an error message appears, check the DATA statements. Block 1 includes lines 5010–5050, block 2 includes lines 5060–5100, and so on. If you find a mistake, fix it and type RUN 60 followed by RUN. Remember to SAVE the final, debugged version.

There is one thing to watch out for. If you decide to use a single line for your window, you can scroll left or right, but don't try to move up or down. For example, if you set the top boundary to five and the bottom to five, you can scroll line five to the left or to the right. But try to scroll up and the computer crashes. And you cannot escape the crash with RUN/STOP–RESTORE. You have to turn your computer off and then on again (and lose whatever you have in memory).

Smoother Scrolling

Regular scrolls move whole characters. It's like picking up a letter and dropping it down one line.

The 64 can do smoother scrolls, moving characters a pixel at a time. The key is memory locations 53270 (horizontal) and 53265 (vertical). To do smooth scrolls, use these formulas:

```
POKE 53270, (PEEK(53270)AND248)+X
POKE 53265, (PEEK(53265)AND248)+Y
```

X and Y can be any numbers from 0 to 7. Once you've gone to 7 or 0, you'll have to do a regular scroll and reset the smooth scroll to the other limit. Smooth scrolling can make an action game look more realistic—the characters don't jump around, they slide.

A minor annoyance in this method is that while the screen is doing a smooth scroll, you may see small gaps at the edges. You can get around this by turning off bit 3 of these two registers; in the POKEs above, AND with 240 instead of 248. In effect, you pull the border in a notch, resulting in a 38 column by 24 row display (instead of 40 x 25).

Because smooth scrolling affects the whole screen, it is not compatible with Scroll 64 windows. If you combined the two, you would see smooth scrolling inside the window and jittery, vibrating characters outside the window. To fix this would require a high-res screen, customized word sprites, or a raster interrupt wedge.

See program listings on page 176.

Three good reasons to try Encore™ diskettes.

1. It saves you money.

Encore diskettes meet the same high standards as the more expensive brands. In fact, Encore actually exceeds system requirements.

Yet you can buy Encore disks for ⅓ the price of the high-priced diskettes. (only $1.59 for single-sided, single-density 5¼" diskettes, when you purchase 4 or more boxes, ten diskettes to a box.)

A diskette intended for a wide range of uses.

Whether you use a lot of diskettes or only a few, you need something more than guaranteed accuracy and reliability. You need economy.

Encore was designed to meet the demand for a reliable, low-cost diskette.

At Encore's low price, it's ideal for memos, rough drafts, spreadsheets, reports, even scratchpad-type uses.

Designed for all major personal computers.

IBM® apple® Radio Shack® commodore®

Encore was specifically designed to provide optimum performance and reliability on the four leading computer systems. Inmac's Quality Assurance Department constantly monitors all Encore diskettes to ensure they meet or exceed the media specifications for these systems.

Our on-going testing program guarantees that the Encore diskette you buy tomorrow will be as good as the one you buy today.

® signifies manufacturer's registered trademark.

2. It's absolutely reliable.

Encore is guaranteed for one full year, or Inmac will replace it, free.

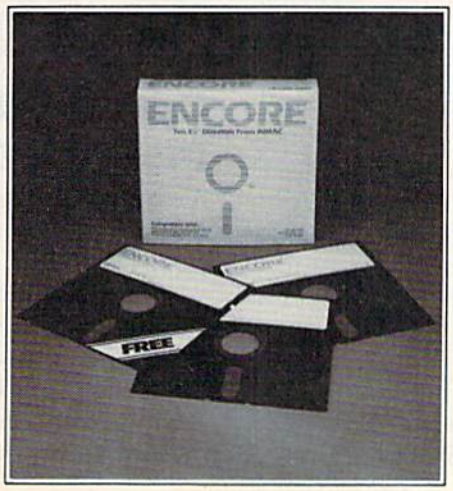
Inmac's Quality Assurance Department requires that Encore meet the media specifications for the most popular systems – Apple, Commodore, IBM, and Radio Shack – exactly. So you can rely on Encore diskettes in your system. They won't lose information or cause read/write errors.

45-day money-back trial.

We're backing our Encore diskettes with a 45-day money back trial because we're positive you'll be delighted with Encore's quality and performance.

And we're sure that once you try Encore, you'll agree that it's the most reliable economy floppy available.

If you don't agree for any reason, just return the three diskettes for a full refund.



3. And you can get one, free.

For a limited time only, we're offering a special trial pack of three 5¼" Encore diskettes. You pay the regular price for two of the diskettes, but the third one is free.

We'll send you 3 single-sided, single-density 5¼" diskettes for only $5.19. (Single-sided, double-density for $5.99. Double-sided, double-density for $8.79.) Use all three Encore diskettes for 45 days.

Then, if you're not completely satisfied, return the three diskettes for a full refund.

Here's how it works.

Simply mail the attached postage-paid card, or phone our toll-free number,

1-800-538-8157

extension 987. In California, 1-800-672-3470, extension 987.

For this special offer, please include payment with your order. You may send a check, money order, or bill it to your Mastercard or Visa account. Company PO's accepted with verification.

Offer is limited to one trial pack per customer. Good only in U.S. Customer must be 18 years or older to order.

Offer expires September 30, 1984.

Remember to ask for your free Inmac catalog. It contains over 2,500 computer supplies and accessories, many not available anywhere except through the Inmac catalog or special offers like this one.

inmac™

2465 Augustine Dr., Santa Clara, CA 95051

Tape Data Files For VIC And 64

Brian Prescott

Storing information on tape files can free up memory for BASIC. The sample programs presented here show how to set up, write to, and read from tape files.

Beginning programmers often balk at writing and reading files to the Datassette. But some serious applications require the use of the same data in more than one program, or several sets of data with the same program. These situations call for data files.

Here's a trio of simple programs that create data files, read them back, and display the contents. Not only will they help you grasp the techniques, but you can also use them to create files for your programs, or incorporate them into your own programs.

The first two programs create data files. Program 3 reads the files and prints the contents to the screen.

Program 1 prompts you for each item. It then writes the items onto a tape file. This method is convenient, but if an incorrect entry is typed in and stored on tape, the only way to correct it is to create a new file, which means entering all the data again.

Program 2 solves this problem, but is perhaps less convenient. To use it, LOAD the program and add DATA statements at lines 540–570. Running the program creates the files.

Creating A File

The programs are fairly straightforward, but a few comments are in order. The first program asks you for the number of items to be in the file, then DIMensions a string array to hold them. The filename is then requested. It's best to use a name that identifies the file. Using +1 as the filename ends the program. After the array is filled and the file written to tape, the program displays the contents on the screen. You could modify the program to allow display and possible editing before the file is created.

The second program does the same job in a slightly different way. The data lines must be organized properly to avoid problems. The first data item will be read as the filename, so be sure the filename is the first item entered. To signal the end of a file, use –1. This is included at the end of the DATA statements. You can create several files at one time, as you can see from the data included. To signal the end of data, use +1—this stops the program.

To see what's on the files, RUN Program 3. You can ask for any file, but be sure to rewind the tape to some point before the file you want.

Opening, Filling, And Closing Files

A tape file is like a desk drawer. First you open it, put something in or take something out, and then close it.

In the first two programs you will see the statement OPEN 1,1,1, "filename". The three numbers following OPEN serve three different purposes. First is the file number. You can pick any number from 1 to 127, but 1 is most commonly used. The second is the device number. Tape

PROSYS

The Professional Systems People And

MICRO WORX

Present Products From

commodore

And

The Software That Makes Them Work!

CBM PRODUCTS

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

SOFTWARE

SBSYS 8032, 8096 & B-Series CPU's
THE SMALL BUSINESS SYSTEM
Available for twin 8250 systems.
GL, AP, AR, INV & payroll modules all communicate. Custom installation recommended.

LEGISYS
8032, 8096 & B-Series CPU's
The total legal office information, accounting and tickler system. Available for 8250 & twin 8250 systems. Custom installation recommended.

C-64 STUFF

C-64 Computer	$219.00
1541 Disk Drive	249.00
1702 Monitor	249.00
1526 Printer	339.00
1600 Modem	69.00
MSD Dual Disk Drive	649.00
MSD Single Drive	349.00

PERSYS 64
For the 'checkbook hacker or the double-entry, 'serious-stuff' accounting buff. Disk: $79.00

PROSYS

PROSYS (C-64, 4000, 8000 & B-Series)
The affordable, professional modular accounting system for business.

Clear instructions that assume no prior knowledge. Computer assigned debits & credits...too many reports to list.

Turns CBM's into veritable giants!

PROSYS
General Ledger as low as $199
AR, AP, INV & Payroll as low as $129 ea.

Call Toll-Free by dialing:
Outside Texas: **1-800-221-WORX**
Inside Texas: **1-800-692-4265,** wait for beep, then dial 008-3378, wait for tone and dial 993, if no answer, dial 806/797-2623.

Send $1 for catalog, refundable on first order.
Prices for cash paid mail order only, subject to change without notice. Credit prices slightly higher.
VISA & MasterCard okay.

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

drives are always device number 1 (usually the screen is device 3, a printer is device 4, and a disk drive is device 8). The final number is the secondary address. A 1 here means "write to the file." Thus, OPEN 1,1,1 tells the computer to open file number 1 on the cassette drive for writing.

Once a file is opened, you can print to it. In the first two programs, you will see PRINT#1,data. PRINT# works like PRINT, except that if you use keyword abbreviations, a question mark (?) won't work. Use P-shift-R instead, followed by the file number. And you have to put a comma between the file number and the data you are writing. After you finish writing the file, CLOSE it.

Opening a file for reading is similar, except that the secondary address is zero. After the file is open, you can INPUT# or GET# from it. You can read and write any type of data—floating point numbers, integers, or strings.

Since the size of a data file can vary, it is advisable to indicate how long the file is, or where it ends. One method is to PRINT# the number of records as the first item in the file. This is best when you are setting up arrays. The computer reads the first number in the file, then DIMensions the array. Another way to mark the length of a file is to make up an end-of-file marker. In the sample programs, "–1" acts as the marker.

Tape Files On A Disk Drive?

Knowing the basics of tape files is helpful if you decide to buy a disk drive. There are a variety of ways to store information on a disk; one of them is very similar to tape files.

Sequential disk files store information in the order it is received (tape files are always sequential). To transfer information from tape to disk, simply open the tape file for reading, open a sequential disk file for writing, and then input the data from tape, print it to the disk, input more, print more, and so on until you reach the end of the file.

See program listing on page 176.

VIC 5K Emulator

Glen Reesor

Some programs written for the unexpanded VIC-20 do not run properly when expansion memory is installed. Often the only solution is to switch off or unplug the expander. You can save wear and tear on your memory cartridges by using the technique outlined here, which "emulates" the memory layout of an unexpanded VIC.

Have you ever encountered the problem of upgrading your computer system in such a way that most of your programs need to be modified to operate properly? I had such a problem when I bought a 16K expander for my VIC-20.

When the 8K or 16K expander is in place, screen memory moves from 7680–8185 to 4096–4601, color memory moves from 38400–38905 to 37888–38393, and the start of BASIC moves from 4096 to 4608. Because of these changes, programs written on a 5K VIC that have POKEs to the screen, POKEs to color memory, or high-resolution graphics will not operate properly with an 8K or 16K expander in place.

Some programs use the formula S=4*(PEEK(36866)AND128)+64*(PEEK(36869)AND112) to locate screen memory. Color memory can be located with the formula C=37888+4*(PEEK(36866)AND128). Using these two formulas, some programs written on a 5K VIC will work. However, depending on the length of the program, high-resolution graphics usually will not work.

Simulating An Unexpanded VIC

In order to get all of my programs (approximately 70) to operate with a 16K expander in place, I came up with two alternatives—rewrite all of my programs, or develop a sequence of commands to make my VIC operate like a 5K VIC. Naturally, I decided on the second alternative.

To make my VIC emulate a 5K VIC, I had to change screen memory, color memory, the start of BASIC, and the end of BASIC. Changing the start of BASIC and the end of BASIC was easy; changing screen memory and color memory was the hard part.

I remembered an article, "Alternate Screens" (*Home and Educational* COMPUTING!, Fall 1981), that could change screen memory to 7168–7673 and the color memory to 37888–38393 (for the 5K VIC). The article provided some commands to change the screen memory back to 7680–8185 and the color memory back to 38400–38905. This is exactly where we want the screen and color memory to be.

Now we are ready to change the VIC with an 8K or 16K expander into a 5K VIC.

1. Turn your VIC off and then on.

2. To change screen and color memory, type the following:

```
POKE36866,150:POKE648,30:FORJ=217TO228:PO
  KEJ,158:NEXT:FORJ=229TO250:POKEJ,159:NE
  XT
```

After you press RETURN, the screen will become a mess of various characters and colors. If you do not see this, you probably typed Step 2 incorrectly. Go back to Step 1. If you typed the line correctly, screen memory is now from 7680–8185, and color memory is from 38400–38905 (normal for a 5K VIC).

3. Press RUN/STOP and RESTORE simultaneously to clear the screen and get a cursor.

4. Now change the start of BASIC and the end of BASIC by typing the following:

```
POKE44,16:POKE56,30:POKE4096,0:CLR:NEW
```

The first POKE changes the start of BASIC to 4096 (16*256). The second POKE changes the end

of BASIC to 7680 (30*256). The third POKE puts a 0 at the start of BASIC. On all Commodore machines, there must be a 0 at the start of BASIC for a program to run. "CLR:NEW" clears any variables and any garbage that may have been in the program area.

Your VIC will now operate like a 5K VIC. Almost all programs written for a 5K VIC will now operate properly with an 8K or 16K expander. The only exceptions are those few programs which manipulate these pointers themselves, or which otherwise tinker directly with memory layout. Even many machine language programs will work.

User Group Update

Kathy Yakal, Editorial Assistant

The Southeast Metro Commodore User Group (SEMCUG) of Portland, Oregon, has been disbanded.

New address for the South Orange County User Group is c/o Steve Wimer, 32221 Alipaz #240, San Juan Capistrano, CA 92675.

The Commodore 64 User Group, Inc., of Glen Ellyn, Illinois, has been disbanded. (New group has been formed; see C-64 User Group, Inc., Lincolnwood, Illinois, in this listing.)

The correct address for the Commodore Preference Users Connection (C.P.U. Connection) is c/o Danni Hudak, P.O. Box 42032, Brook Park, OH 44142.

The Capitol Area Commodore Club can now be reached at P.O. Box 333, Lemoyne, PA 17043.

The VIC-20 User Group of Lincolnton, North Carolina, has expanded to include coverage of the Commodore 64. The new name is VIC-20/64 User Group.

The new phone number for the Commodore PET User Group of Gretna, Louisiana, is (504) 455-4619.

The new address for the North Country Computer Club is c/o Eleanor Cunningham, 1607 Ford St., Ogdensburg, NY 13669. (315)393-2708.

Correspondence for the 64 User Group (formerly of Midnight Circle in Plano, Texas) should be addressed to P.O. Box 801828, Dallas, TX 75380.

The new address for the Quad Cities Commodore Computer Club is c/o Mike Hoeper, P.O. Box 3994, Davenport, IA 52808. (319)242-1496.

The Central Washington Commodore User Group can be contacted at P.O. Box 10937, Yakima, WA 98909.

When writing to a user group for information, please remember to include a self-addressed, stamped envelope. Send additions, corrections, and deletions for this list to:

COMPUTE! Publications
P.O. Box 5406
Greensboro, NC 27403
attn: Commodore User Groups

Commodore Club/South
William J. Freeman
1641 3rd Pl. NW
Birmingham, AL 35215

Southern California 64 User Group
Robert Johnson
14944 Bayou Ave.
Bellflower, CA 90706
(213) 867-4398

Amateurs And Artesians Computing
Bill Alexander
P.O. Box 682
Cobb Mountain, CA 95426
c/o Alex KR6G

Santa Rosa Commodore 64 User Group
Garry Palmer
333 E. Robles Ave.
Santa Rosa, CA 95407
(707) 584-7009

The Exchange
Michael C. Joseph MD
P.O. Box 9189
Long Beach, CA 90810
(213) 595-1771

Castlegar Commodore Computer Club
SS1 S37 C7
Castlegar, B.C., Canada V1N 3H7
(604) 365-3889

The London Commodore User Club (LCUC)
Dennis Trankner
28 Barrett Cr.
London, Ontario,
Canada N6E 1T5
(519) 681-5059

Diamond State User Group
Michael Butler
Box 892 Rd 2
Felton, DE 19943
(302) 284-4495

VIC 20 Owners Club
Tom Lomax
17 Greystoke Dr.
Bilborough, Nottingham,
Nottinghamshire, England
Phone: Nottingham 289567

Chips User Group
Jerry Klimczak
UMR Box 3063
Avon Park, FL 33825

Clearwater Commodore Club
Gary Gould
1532 Lemon St.
Clearwater, FL 33516
(813)442-0770

Commodore Connection Computer Club
Craig Mihalko
P.O. Box 6684
West Palm Beach, FL 33405

Lake County Commodore Club
John Ziegler
P.O. Box 326
Tavares, FL 32778
(904) 343-4499

Commodore User Group
(forming)
David S. Weatherly
1401 Denfield Ct.
Lithonia, GA 30058

Atlanta Commodore 64 User Group
Ronald Lisoski
1767 Big Valley Ln.
Stone Mountain, GA 30083
(404) 981-4253

Commodore User Group
Grant Bervick
310 Emerald Dr.
Kellogg, ID 83837
(208) 784-8751

User Group of Lower Idaho (U.G.L.I.)
Sean Brixy
Rt. 4 Box 67
Rupert, ID 83350

Fox Valley 64 User Group
Frank Christensen
P.O. Box 28
N. Aurora, IL 60542
(312) 898-2779

Commodore Club Of Evanston (COMCOE)
Jim Salsbury
2108 Sherman Ave.
Evanston, IL 60201

Pros And Newcomers Into Commodore (P.A.N.I.C.)
David A. Davis
R.R.5 Box 243
Danville, IL 61832
(217) 443-5573

South Suburban VIC-20 User Club
Nick Forte
16409 Emerald Ave.
Harvey, IL 60426
(312) 331-4851

The C-64 User Group Inc.
Darrell Hancock
David Tamkin
P.O. Box 46464
Lincolnwood, IL 60646
(312) 588-0334 (Hancock)
(312) 583-4629 (Tamkin)

Dekalb County Computer Club
Darla McCammon
4896 C.R. 60
Saint Joe, IN 46785
(219) 337-5528 (home)
(219) 432-4300 (work)

Computer Hobbyists In Plymouth (CHIP's)
Don Myers
201 Dickson St.
Plymouth, IN 46563
(219) 936-2423

Commodore Owners Of Lafayette (COOL)
Ross Indelicato
20 Patrick Ln.
W. Lafayette, IN 47906
(317) 743-3410

Iowa City Commodore User Group
John Navitsky
616 Hawkeye Dr.
Iowa City, IA 52240

Newton Commodore User Group
David Schmidt
320 W. 9th St. S.
Newton, IA 50208

Commodore User Group Ireland
W. J. Murphy
3 Woodlands Dr.
Stillorgan, Blackrock,
County Dublin, Ireland

Randallstown C-64 User Group
Steven Zigler
8827 Sigrid Rd.
Randallstown, MD 21133
(301) 655-8402

Commodore 64 User Group
Jorge Montalvan
11209 Tack House Court
Potomac, MD 20854
(301) 983-8199

Commodore 64 User Group (forming)
Steve Lepsetz
20050 Winchester
Southfield, MI 48076
(313) 354-7224
(313) 353-1130

Slipped Disk Inc.
31044 John R
Madison Heights, MI 48071
(313) 583-9803

Heartland Area Computer Cooperative
Robert (Sam) Walz
Rt. 4 Box 204
Little Falls, MN 56345
(612) 632-5511

Commodore Computer Club
Andrew Holder
Southern Station, Box 10076
Hattiesburg, MS 39401
(601) 268-7585

Hackers Of Lake St. Louis (HLSL)
Jim Gaeschke
513 Oak Terrace
Lake St. Louis, MO 63367
(314) 625-4782
BBS (314) 625-4782

Desert Byte Computer Club
Lenny Baldini
P.O. Box 3757
Tonopah, NV 89049
(702) 482-5565

Morris Area Commodore User Group
Bob Searing
51 Ferncliff Rd.
Morris Plains, NJ 07950

Hudson Valley Commodore Club
P.O. Box 2190
Kingston, NY 12401

The Northern New York Commodore Home User Group (NORNY-CHUG)
P.O. Box 226
Norwood, NY 13668
Andrew VanDuyne
(315) 353-4591

64 Southtowns User Group
Michael Tubbert
59 Bright St.
Cheektowaga, NY 14206
(716) 893-6586

The New York City VIC-20/C-64 User Group (NYCUG)
Joycelyn Woods/Allan Hobbs
436 E. 69th St.
New York, NY 10021
(212) 787-2854
BBS (212) 534-3149

Zweibrucken Commodore Computer Club
James K. Sturgeon
Box 1044, TSA-EURCOR,
APO, NY 09052
(This group is in Germany)

Head Start Commodore User Club
Anne Murphy
5701 Main St.
Williamsville, NY 14221
(716) 631-0241

Staten Island Commodore User Group (SICOM)
Tony Isoldi
3770 Richmond Ave.
Staten Island, NY 10312
(212) 984-4256

New Zealand Commodore User Group
J. R. Walker
P.O. Box 5223,
Auckland, New Zealand

CCND
James Allen
1607 Reno Dr.
Bismark, ND 58501

Marion Ohio Commodore User Group (MOCUG)
Van Munro
775 Wolfinger Rd.
Marion, OH 43302
(614) 726-2630

Jefferson State Computer User Group
2355 Camp Baker Rd.
Medford, OR 97501

Klamath Commodore 64 User Group
Mrs. Juanita Eiteneier
P.O. Box 7654
Klamath Falls, OR 97602

Commodore User Group of South Pittsburgh
Groves TV & Computers
2407 Penna Ave.
West Mifflin, PA 15122
c/o Charles W. Groves III

The Charleston Computer
Jack A. Furr Jr.
P.O. Box 5264
N. Charleston, SC 29406
(803) 747-0310
BBS (803) 747-6981

ET 64 User Group
Walt Turner
P.O. Box 495
Knoxville, TN 37901

Corpus Christi Commodores
Bob McKelvy
P.O. Box 6541
Corpus Christi, TX 78411
(512) 852-7665

Bunch 'A Bytes
Joey Miller
4916 121st Pl. NE
Marysville, WA 98270

Ellensburg Commodore User Group
William C. Smith
Department of Anthropology
Central Washington University
Ellensburg, WA 98926
(509) 936-3489 (work)
(509) 925-2981 (home)

C-64 Diversity
Jill Johnston
18204 67th Ave. NE
Arlington, WA 98223
(206) 435-4580

Spokane Commodore User Group
S.A. White
N. 4311 Whitehouse
Spokane, WA 99205

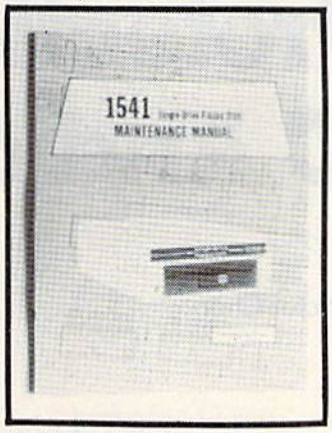
Kanawha Valley Commodore 64 User Group
Jeff McClintick
P.O. Box 922
Charleston, WV 25314

Commodore Hobbyists Involved In Personal Systems (CHIPS)
Richard M. Kohn
1017 Kilbourn Ave.
West Bend, WI 53095
(414) 338-1609 (days)
(414) 334-2494

GOSUB

How to do your own maintenance, troubleshooting, schematics, theroy 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 (Pictoral) views
Test point locators



With all these illustrations and the detailed theory for each circut 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 tempratures to safe operating levels.
Does not promote dust migration.
No added noise.
Easy to install.
Increases life expectancy and realiability of disk drive.
Increases operating time 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 dissipateded into the surrounding air. The kit will lower operating tempratures 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 temprature ratings.

Suggested list price: $24.95

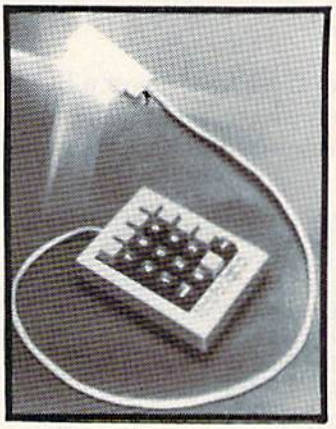
GRIDIRON STRATEGY '64

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 an a scrolling field. NOT GRIDIRON. In GRIDIRON, you coach the entire team and the colorful field and the stadium styled scoreboard are completely visiable 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. These 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 print out copy of plays and statistics?
Individuaoized teams, based on actual performances?
96 possible play combinations, infinate 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



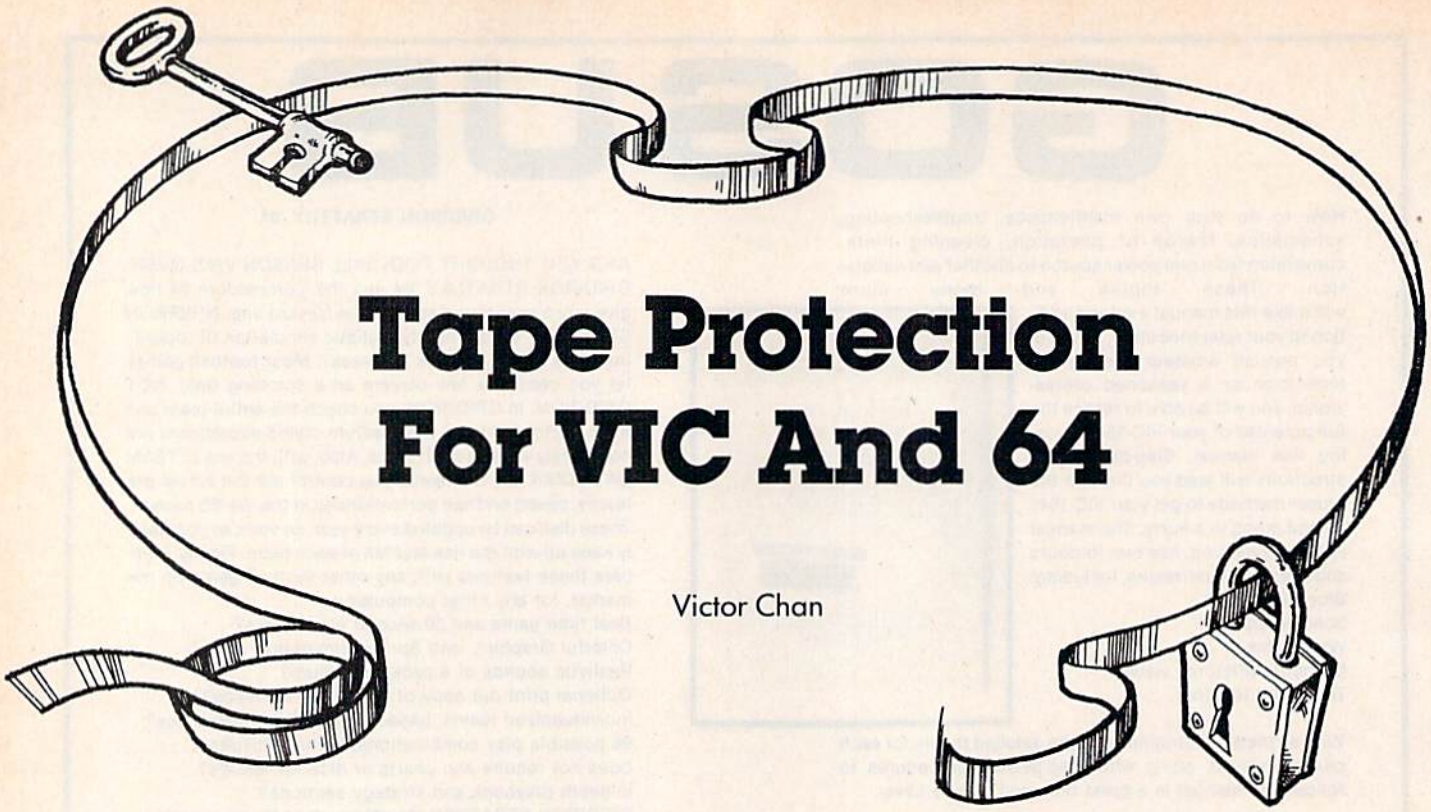
THE FLEXI-KEY SYSTEM

20 keys, 19 of which may have 3 seperate definitions! Complete documentation! Works on the VIC-20 (Expanded) and C-64 Computers! Compatible with most existing software! Great for use with business programs and electronic spread sheets! Works with COMPUTES MLX program as an aid to machine language programmers!

Suggested list price: $69.95

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



Tape Protection For VIC And 64

Victor Chan

Here are several tricky techniques that allow you to protect your tape programs from being copied.

In the VIC and 64, whenever a LOAD from tape or SAVE to tape is processed, the section of memory known as the tape buffer will be used as follows:

Location	Use
828 ($033C)	Type of tape file
829 ($033D)	Low byte of start address for LOAD/SAVE
830 ($033E)	High byte of start address for LOAD/SAVE
831 ($033F)	Low byte of end address for LOAD/SAVE
832 ($0340)	High byte of end address for LOAD/SAVE
833–1019 ($0341–$03FB)	Program name or filename (padded with spaces to fill the buffer)

When a program is being LOADed or SAVEd, the filename will be stored starting at memory location 833, and the rest of the tape buffer will be filled with spaces. The computer compares the program name with the filename on the tape. A program is *found* on tape when all of the characters of the filename in the LOAD statement are matched. Therefore, being able to find a program does not guarantee that the filename specified in the LOAD statement is the same as that of the filename on the tape.

For example, LOAD "VIC" will LOAD a tape file named VIC, or VIC20, or VICTOR, or even VIC@+=$%@$%.

All that matters is that all of the characters specified in the filename in the LOAD statement are matched. When a program is found during a LOAD, the screen displays only the first 16 characters of the filename. Thus, if a user SAVEs a program according to the name shown on the screen during a LOAD, the file may not be SAVEd with the same filename as the original one on the tape, especially if part of the filename is nonprintable characters. This property of the load statement can be used to prevent unauthorized tape reproduction.

Let's look at three methods of protecting tape files.

SAVEing With Hidden Characters

In immediate mode, the VIC can execute a line of up to 88 characters long; the 64 is limited to 80 characters. To SAVE a program with a long filename, the whole 80 or 88 characters may be used. If a program is SAVEd with a filename that is longer than 16 characters, the characters starting at the seventeenth position are not displayed.

To protect a program, first SAVE the program with a name longer than 16 characters. Then, somewhere in your program, check for one or more of the characters beyond the sixteenth position. If a match is not found, do a SYS to some arbitrary position to crash the program.

Here is an example.

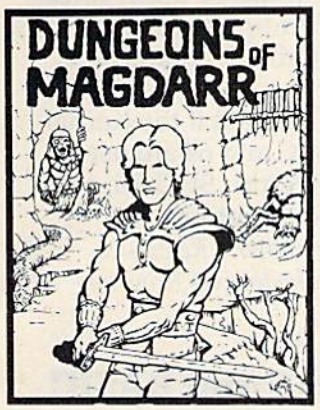
1. SAVE the program you wish to protect with a character in the seventeenth position of the filename:

```
SAVE"NAME{12 SPACES}A"
```

2. Include this subroutine in the program

AARDVARK LTD.

VIDEO ADVENTURES™



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

Available On: TRS80C, IBM PC, CMD64

TAPE $19.95 DISK $24.95

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

Available On: TRS80C 32K, CMD64

TAPE $19.95 DISK $24.95

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

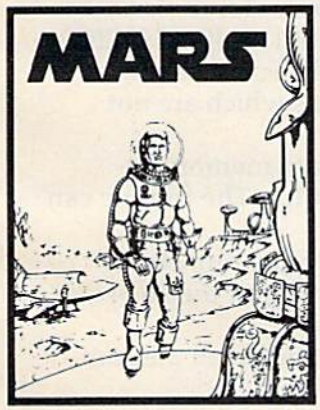
Available On: TRS80C 16K, CMD64, VIC20 13K, MC10 16K, TI99 (EXT. BASIC), IBMPC

TAPE $14.95 DISK $19.95

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

Available On: TRS80C 16K, CMD64

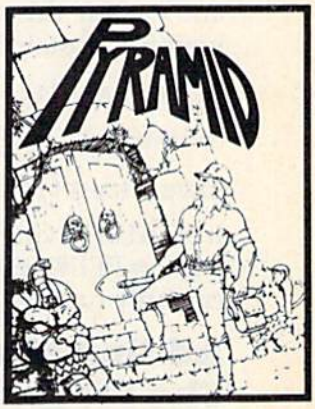
TAPE $19.95 DISK $24.95



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

Available On: TRS80C, CMD 64, IBM PC

TAPE $19.95 DISK $24.95



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

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

TAPE $19.95 DISK $24.95

NEW! GRAPHIC ADVENTURES

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

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

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

MasterCard CHARGE CARDS WELCOME VISA

1-313-669-3110

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

AARDVARK *Action Software*

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

CMD64 / TRS80C / IBMPC / VIC20 / TI99

and execute it with a GOSUB 60000 early in the program:

```
60000 A = PEEK (833+16) : IF CHR$(A)="
      A"THEN RETURN
60010 SYS 833
```

When the program is LOADed, the filename will be displayed as NAME, so if someone attempts to make an unauthorized copy, they will probably use NAME alone. Then, when the program goes to the subroutine at line 60000, it will not find the required A in the seventeenth position.

The SYS 833 (any memory address can be used) will send the computer off to execute a machine language program where no such program exists. This doesn't hurt the computer, but the resulting *lockup* will probably make it necessary to turn the computer off and back on.

SAVEing With Hidden Variables

The method described above checks the contents of a certain memory location in the tape buffer. To provide even more protection, the contents of the location may be used to initialize some variables in the protected program. If the user tries to SAVE without the full filename, the program does not initialize properly and will not run.

For example, if the value of the variable Z is set to 45 in the protected program, include in the SAVE statement the character equivalent of 45 at or after the seventeenth position in the filename. To find out the character equivalent of any number, simply use the BASIC command CHR$, and include this character between the quotes in your SAVE statement. For example, use

```
SAVE "NAME{12 SPACES}-"
```

or simply

```
SAVE "NAME{12 SPACES}"+CHR$(45)
```

Instead of having a line with Z=45 in the program, use Z=PEEK(833+17). This way the program can only run properly with the full filename.

Adding Machine Language To The Program Name

The final approach is the most complicated. It requires the use of machine language. You use part of the filename portion of the tape buffer to hold a machine language subroutine. Again, the program cannot run properly unless the program is SAVEd with the original name. To use this method, put a SYS somewhere in the protected program which calls the subroutine in the tape buffer. If the subroutine is not there (if the program was not SAVEd with the original filename, including the machine language portion), the program will probably crash when it attempts to execute the contents of the tape buffer as machine language.

It would be a tedious process to calculate the CHR$ equivalent of every byte in the ML routine and type the corresponding characters as part of the filename. Also, only a portion of the available 171 bytes (locations 849–1019) could be used, since the length of the filename (including the SAVE command and quotes) is limited to 88 characters on the VIC, 80 on the 64.

These limitations can be avoided if the Kernal ROM routines built into the VIC and 64 are used. The required routines are SETNAM, SETLFS, and SAVE. For information on these routines, see Chapter 3 of the *VIC-20 Programmer's Reference Guide* or Chapter 5 of the *Commodore 64 Programmer's Reference Guide*.

Using The Registers From BASIC

These ROM routines all require that values be placed into the microprocessor's A, X, and Y registers, which can be done directly only with ML programming. The key to using these routines without resorting to ML is knowing that values POKEd to location 780 will be loaded into the A register, location 781 will be loaded into the X register, and location 782 will go into the Y register.

As an example, the following steps illustrate one way to use the Commodore 64 Kernal routines to SAVE a BASIC program from memory to tape with a machine language subroutine as part of the filename:

1. POKE the character values for the desired program name into locations 49152–49167 ($C000–$C00F). POKE the value for the space character (32) into any of the 16 locations which are not used.

2. Load the ML routine into memory beginning at location 49168 ($C010). The routine can be up to 171 bytes long.

3. POKE location 780 with the number of bytes in the ML routine, plus 16 (for the 16 bytes in the name).

4. Use the SETNAM routine to tell the computer where to find the filename:

```
POKE 781,0:POKE 782,192:SYS 65469
```

The POKE values given are for a filename starting at location 49152 (256*192+0=49152). These values must be changed if the filename is at some other point in memory.

5. Use the SETLFS routine to specify that a SAVE is to tape:

```
POKE 780,1:POKE 781,1:POKE 782,255:SYS 65466
```

6. Use the SAVE routine to store the BASIC program on tape. The data in the section of memory defined in Step 4 will be copied into the tape buffer and used as the filename:

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.

YORK 10® CASSETTES

COMPUTER GRADE • DATA TRAC • BLANK CASSETTES

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

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

- BASF-LHD (DPS) world standard tape.
- Premium 5 screw shell with leader.
- Internationally acclaimed. Thousands of repeat users.
- Error Free • Money back guarantee.

BUY THE BEST, AT FACTORY-DIRECT PRICES

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

CASSETTE STORAGE CADDY
Holds 12 cassettes $2.95 w/lb boxes. Includes edge labels and index card

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

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

Call: 818/700-0330
FOR IMMEDIATE DELIVERY on Credit Card Orders.

ORDER NOW . . . MAIL TO –
YORK 10 9525 Vassar Ave. #G Chatsworth, CA 91311

ORDER FORM #G

ITEM	1 DOZEN	2 DOZEN	TOTAL
C-05	☐ 7.00	☐ 13.00	
C-06	☐ 7.00	☐ 13.00	
C-10	☐ 7.50	☐ 14.00	
C-12	☐ 7.50	☐ 14.00	
C-20	☐ 8.75	☐ 16.50	
C-24	☐ 9.00	☐ 17.00	
C-32	☐ 11.00	☐ 21.00	
Hard Box	☐ 2.50	☐ 4.00	
Wht. Labels	☐ 3.00/100	☐ 20.00/1000	
Color Labels	☐ 4.00/100	☐ 30.00/1000	
Color			
Storage Caddy @ 2.95 ea. Qty			
		SUB TOTAL	
Calif. residents add sales tax			
Shipping/handling			3.50
Outside 48 Continental States — Additional $1 per caddy, per doz. cassettes or boxes			
		TOTAL	

Each cassette includes 2 labels only. Boxes sold separately. In Continental U.S. shipment by U.P.S. If Parcel Post preferred, check here. ☐

Check or M.O. enclosed ☐ Send Quantity Discounts ☐
Charge to credit card: VISA ☐ MASTERCARD ☐

Card No. _____ Exp. _____
Name _____
Address _____
City _____ State/Zip _____
Signature _____ Phone _____

Ask about our DUPLICATING SERVICE

YORK 10® DISKETTES

BRAND NAME QUALITY "DOUBLE DENSITY" AT DISCOUNT PRICES!

"Plain Wrap" $1.59* ea.

• Certified 100% Error-free • 5 YEAR WARRANTY

BASF $2.19* ea. qualimetric | Dysan $2.69* ea.

*5¼" SSDD, Soft sector, price per disk, 100 pak.

TRACTOR FEED DISKETTE LABELS 1 7/16 x 5" 2¢ EACH 1000 QUANT.

FLIP "N" FILE "15" $7.95 for 5¼" Diskettes
"25," $21.95 "50," $31.95

Call: 818/700-0330
FOR IMMEDIATE DELIVERY on Credit Card Orders.

ORDER NOW . . . MAIL TO –
YORK 10 9525 Vassar Ave. #G Chatsworth, CA 91311

ORDER FORM

PLEASE INDICATE QUANTITIES DESIRED

SIZE	Plain Wrap	BASF	Dysan	TOTAL
5¼" SSDD	10 17.90 / 100 159.00	10 24.90 / 100 219.00	10 29.90 / 100 269.00	
5¼" DSDD	10 20.70 / 100 182.00	10 33.90 / 100 299.00	10 41.30 / 100 363.00	
5¼" DS96tpi		10 47.90 / 100 421.00	10 52.50 / 100 461.00	
8" SSDD		10 30.90 / 100 274.00	10 35.70 / 100 314.00	
8" DSDD		10 38.60 / 100 339.00	10 41.30 / 100 363.00	
DISKETTE LABELS: ☐ $3.00/100 ☐ $20.00/1,000				
FLIP 'N' FILE 5¼": "15" $7.95 qty. ____ "25" $21.95 qty. ____ "50" $31.95 qty. ____				
			SUB TOTAL	
Calif. residents add sales tax				
Shipping/handling (any quantity)				3.50
Outside 48 Continental States: Additional $1 per 10 pak, per file				
			TOTAL	

All Diskettes are soft sectored, unformatted. In Continental U.S., shipments by U.P.S. If Parcel Post preferred, check here ☐ #G

Check or M.O. enclosed ☐ Send Quantity Discounts ☐
Charge to credit card: VISA ☐ MASTERCARD ☐

Card No. _____ Exp. _____
Name _____
Address _____
City _____ State/Zip _____
Signature _____ Phone _____

Ask about our DUPLICATING SERVICE

```
POKE 780,43:POKE 781,PEEK(45):POKE 782,PEEK
(46):SYS 65496
```

Press play and record on the Datassette when instructed to do so, and your program will be SAVEd with the ML subroutine included.

Also Good For VIC

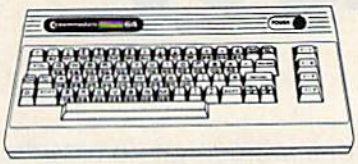
This same procedure (with the same SYS locations) can be used on the VIC. But you must change the memory locations where the program name and ML routine are stored. Locations 49152 and up cannot be used on the VIC. Some other unused area of RAM will have to be chosen. Note that the values POKEd to locations 781 and 782 in Step 4 above will have to be modified to reflect the new address of the filename.

If your protected program includes a SYS to the ML in the tape buffer, it will work properly only if the program has been SAVEd in this manner, so the chances of someone breaking this protection scheme are very slim.

Using any one or a combination of the three different levels of difficulty described should reduce unauthorized tape reproduction.

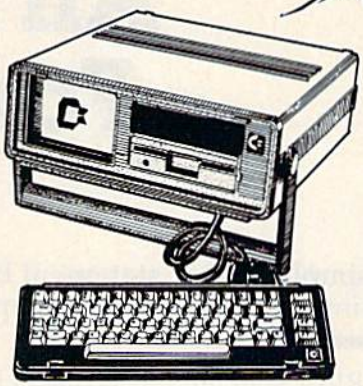
COMPUTER MAIL ORDER

commodore



COMMODORE 64.
$199

SX-64 PORTABLE
$839



VIC 20.....CALL	
M-801 Dot Matrix/Parallel	$219.00
MCS 801 Color Printer	$499.00
1520 Color Printer/Plotter	$129.00
1530 Datasette	$69.00
1541 Disk Drive	$249.00
1600 VIC Modem	$59.00
1610 VIC Term 40	$59.00
1650 Auto Modem	$89.00
1702 Color Monitor	$249.00
DPS Daisywheel Printer	$459.00
Magic Voice Speech Module	$54.00
Desk Organizer Lock	$49.00
1311 Joystick...each	$4.99
1312 Paddles	$11.99
1110 VIC 8K	$42.00
1111 VIC 16K	$69.00
1011 RS-232 Interface	$42.00
1211 Super Expander	$53.00

MSD

SD 1 Disk Drive	$349.00
SD 2 Disk Drive	$599.00

CARDCO.

Light Pen	$32.00
3 Slot VIC Expansion Interface	$32.00
6 Slot Expansion Interface	$79.00
Cassette Interface	$29.00
Parallel Printer Interface	$49.00
Parallel Interface w/Graphics	$69.00

PRINTERS

Epson	CALL
Okidata	CALL
Star Gemini 10X	$299.00
Star Gemini Delta 10	$559.00
Smith Corona TP-2	$399.00
C.Itoh Prowriter 8510P	$379.00
C.Itoh Gorilla	$209.00
BMC BX-80	$269.00

DUST COVERS

C-64/VIC 20 Cover	$9.99

SOFTWARE

commodore

CBM 64 Reference Guide	$18.00
C-64 DISKS	
EasyCalc	$65.00
EasyFinance I, II, III, IV	$19.00
EasyMail	$19.00
EasyScript	$39.00
Word/Name Machine	$19.00
EasySpell	$19.00
Accounts Receiveable	$39.00
Accounts Payable	$39.00
General Ledger	$39.00
Assembler	$19.00
Logo	$52.00
Pilot	$45.00
Pet Emulator	$19.00
Screen Editor	$19.00
Music Machine	$15.00
Music Composer	$15.00
VIC 20 CARTRIDGES & DISKS	
Gortek & the Micro Chips (C)	$19.00
Super Slot (R)	$14.00
Super Alien (R)	$14.00
Jupiter Lander (R)	$14.00
Radar Rat Race (R)	$14.00
Count Adventure (R)	$21.00
Pinball Spectacular (R)	$19.00
VIC Reference Guide	$15.00

ARTWORX
C-64/VIC 20 CASSETTES

Bridge 4.0	$15.00
Teacher's Pet	$12.00

BRODERBUND
VIC 20 CASSETTES

Martian Raid	$16.00
Shark Trap	$16.00
Multisound Synthesizer	$16.00

QUICK BROWN FOX
C-64/VIC 20 CARTRIDGE

Word Processor	$49.00

COMMERCIAL DATA
VIC 20 CASSETTES

Motor Mouse	$23.00
Centipod	$23.00
Frogee	$23.00
C-64 CASSETTES	
Road Toad	$24.00

CREATIVE SOFTWARE
C-64 CASSETTES

Home Inventory	$11.00
Household Finance	$23.00
C-64 CARTRIDGES	
Trashman	$29.00
C-64 DISKS	
Home Inventory	$11.00
Household Finance	$29.00
VIC 20 CASSETTES	
Home Inventory	$12.00
Household Finance	$14.00
VIC 20 CARTRIDGES	
Astro Blitz	$21.00
Black Hole	$32.00
Trashman	$21.00
Choplifter	$21.00

DYNATECH
C-64 DISKS

Codewriter	$75.00

EPYX
VIC 20 CASSETTES

Ricochet	$32.00
Sword of Fargoal	$24.00
Rescue at Rigel	$24.00

UMI
VIC 20 CASSETTES

Amok	$30.00
Meteor Run	$40.00
Alien Blitz	$30.00
VIC 20 CASSETTES	
Cloud Burst	$15.00
Video Verman	$19.00

ATARISOFT
C-64/VIC 20 CARTRIDGES

Centipede	$37.99
PacMan	$37.99
Donkey Kong	$37.99
Dig Dug	$37.99
Defender	$37.99
Robotron	$37.99
Stargate	$37.99

HES
VIC 20 CARTRIDGES

VIC Forth	$32.00
HES Mon	$29.00
HES Writer	$29.00
Aggressor	$29.00
Synthesound	$21.00
Shamus	$29.00
Protector	$29.00
Turtle Graphics	$29.00
C-64 CARTRIDGES	
HES Mon	$29.00
HES Writer	

NUFEKOP
VIC 20 CASSETTES

Alien Panic	$10.00
Race Fun-Drag Race	$16.00
The Catup	$10.00
Exterminator	$19.00
C-64 CASSETTE	
3—D Man	$16.00

ROMOX

Type (20/64)	$29.00

RAINBOW
C-64 DISKS

Personal Finance	$48.00
Writer's Assistant	$95.00
Spreadsheet Assistant	$95.00

QUANTUM

40/80 Column Video Board	$95.00
40/80 Column Video Board (16K)	$179.00

SIRIUS
VIC 20 CARTRIDGES

Deadly Duck	$21.00
Spider City	$29.00

MICROSPEC
VIC 20 CASSETTES

Spelling Bee-	
Grades 2, 3, 4, 5 or 6	$8.00
Math Drill	$8.00
Portfolio Manager	$16.00
Data Manager	$16.00
VIC 20 DISKS	
General Ledger	$69.00
Mailing List Manager	$35.00
Inventory Package	$69.00
Payroll	$69.00
Data Base	$49.00
C-64 CASSETTES	
Black Box	$12.00
Color Sketch	$20.00
Match Maker	$16.00
C-64 DISKS	
Mailing List Manager	$45.00
Inventory Package	$79.00
General Ledger	$79.00
Payroll	$79.00
Data Base	$69.00
CalcResult	$139.00
Black Box	$16.00
Color Sketch	$22.00
Match Maker	$20.00

VICTORY
C-64/VIC 20 CASSETTES

Adventure Pak I (3 games)	$12.00
Adventure Pak II (3 games)	$12.00
Annihilation	$16.00
Grave Robber	$11.00
Kongo Kong	$16.00
Trek	$11.00

PROFESSIONAL SOFTWARE

Word Pro 64 Plus Spell	$65.00

We carry a selection from the above manufacturer's plus...
Synapse, Thorn, InfoCom, Onslow, Practicalc, Spinnaker & Timeworks.

east
800-233-8950
In PA call (717)327-9575,Dept.40613
Order Status Number: 327-9576
Customer Service Number: 327-1450
477 E. 3rd St., Williamsport, PA 17701

canada
Ontario/Quebec
800-268-3974
Other Provinces800-268-4559
In Toronto call (416)828-0866,Dept.40613
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 3B
Mississauga, Ontario, Canada L5L1T1

west
800-648-3311
In NV call (702)588-5654,Dept.40613
Order Status Number: 588-5654
P.O.Box 6689
Stateline, NV 89449

MasterCard VISA

No risk, no deposit on C.O.D. orders and no waiting period for certified checks or money orders. Add 3% (minimum $5) shipping and handling on all orders. Larger shipments may require additional charges. NV and PA residents add sales tax. All items subject to availability and price change. Call today for our catalog. CANADIAN ORDERS: All prices are subject to shipping, tax and currency fluctuations. Call for exact pricing in Canada.

All About PRINT For VIC And 64

Julie Harris

The simple PRINT statement becomes a drawing tool in this screen graphics tutorial.

The PRINT statement was one of the first BASIC statements you learned when your computer was so new it still gleamed. You may be interested in more complicated routines now—reading joysticks, creating your own graphics characters, or writing machine language. But the humble PRINT statement still has something to offer.

You can use PRINT in a generalized subroutine that draws pictures using the keyboard graphics characters. PRINT is useful if you want to manipulate graphics characters as if they were in a variable-length array. However, this won't use the storage space required for arrays. Or, you can PRINT an unending series of pictures, each unique, and never run out of memory.

Let's begin with a review of some basic characteristics of the PRINT statement.

Commas And Semicolons In PRINT Statements

When a PRINT statement ends with a semicolon, the next character printed will appear in the next horizontal screen location. For example, PRINT "THREE": PRINT "DIFFERENT": PRINT "LINES," will print

```
THREE
DIFFERENT
LINES,
```

but PRINT "ALL ";: PRINT "ON ";: PRINT "ONE." will print

```
ALL ON ONE.
```

In the example above, each word ends with a space. If you left out the spaces, you would see something like this: ALLONONE. Using a semicolon after a PRINT tells the cursor to stay put. If you forget to include the spaces, the words run together.

The rules are slightly different when you PRINT numbers. Try PRINT1;2;3 and you will find that each number has a space on either side. Whenever you PRINT a number, a trailing space is automatically added and the space in front is reserved for a minus sign (in case the number is negative. If you enter PRINT–1;–2;–3 you see only one space between the numbers instead of two.

When you use PRINT by itself, the computer prints and then moves to the beginning of the next line; the result is a printed list of items. Using a semicolon makes the cursor stay where it is, and everything runs together.

If you want something in between the two extremes, you can use a comma to separate the variables. This is helpful when you want columns (rather than just a list). A VIC screen has two columns of eleven characters; a 64 gives you four columns of ten characters. PRINT "A","BC", "DEF","GHIJ","KLMNO","PQRSTUVW" will show you how the columns look (notice that the left edges are lined up). Using commas to make columns can be useful in a variety of applications. For example, FORX=56TO63: PRINTX,PEEK(X): NEXT prints a column of memory addresses followed by their contents.

PRINTing Characters With CHR$

The CHR$ function can be used in place of any string in the PRINT statement. For instance, PRINT CHR$(65) has the same effect as PRINT "A". A list of CHR$ codes can be found in the appendices of the owner's manual or *Programmer's Reference Guide* for your computer.

Some of the CHR$ codes are used for characters. Others are used for functions like "clear screen" or "cyan."

Positioning PRINT

PRINT causes the printing to begin in the first space of the next available unused line on the screen. This beginning position can be controlled, however, by using the equivalent of X and Y coordinates. The Y coordinate specifies the line on which to begin printing, and the X coordinate specifies the space (column) within that line.

Vertical (Y) positioning can be controlled using the LEFT$ function. LEFT$(X$,I) returns a string containing the leftmost I characters of string X$. In positioning printing, we will define a string L$ = "{HOME}{21 DOWN}" for the VIC, or L$ = "{HOME}{23 DOWN} for the 64. To begin on

The Banner Machine™

For the **Commodore 64** (3 extra fonts available). For the **VIC-20** with 24K memory (2 extra fonts available). • Use on any Gemini or Epson MX with Graftrax or the FX and RX printers. Also Commodore 1525E and Banana with the C-64. • Menu-driven program operates like a word processor. • Makes signs up to 13" tall by any length. • Makes borders of widths up to ¾". • 8 sizes of letters from ¾" to 8" high. • Proportional spacing; Automatic centering; Right and left justifying. • $49.95 Tape or Disk (Specify computer equipment)

For the Commodore 64:

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

CTRL-64 Permits listing of C-64 programs on non-Commodore printers. Lists control symbols in readable form. Tape or disk $24.95

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

Preschool Educational Programs ABC Fun and 123 Fun have bright color and action. **Each** $14.95

Formulator A scientific calculator for repetitive arithmetic computations. Save formulas. Ideal for chemistry, physics, or engineering students. $39.95

MUPOS Create a file of up to 9 basic programs from keyboard/tape/disk and run each program selectively. Combine programs. Renumber BASIC lines. Note pad, and more. Tape or disk $29.95

Grade Organizer Teachers—store grades for 6 classes, up to 40 students each, 680 grades per student. Print interim and final reports, class rosters, and more! Disk $39.95

Cardinal Software™

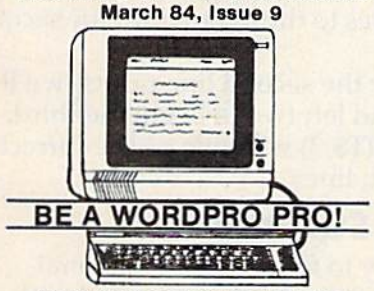
Virginia Micro Systems, 13646 Jeff Davis Hwy., Woodbridge, VA 22191

ORDERS ONLY (800) 762-5645
FOR INFORMATION (703) 491-6502

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

Professional Software Inc.

"#1 Selling Software" for Word Processing per Compute Gazette March 84, Issue 9



BE A WORDPRO PRO!

WORDPRO 3+/W SPELLRIGHT

- Numeric Mode (add columns)
- Easy directory loading!
- Built-in dictionary w/"Fast Scan" option $79.00

SPELLRIGHT (DISK) 20,00 word † 1500 word user-defined to "add to" WordPro 3+ $39.00

NEW! WORDPRO FOR IBM-PC

Word-Plus PC w/The "BOSS", On-Line "SPELL-CHECKING & CORRECTION" add professional "WINDOWING" dictionary.MS-DOS compatible. $450.00
(Merges with 1-2-3, dBase II, VISICALC, SUPERCALC, & MULTIPLAN)

SJB DISTRIBUTORS INC.
3524 Dividend Drive
Garland, Texas 75042
1-800-527-4893 (US)
1-800-442-1048 (Texas)
VISA/MC-add 3% Continental US orders only

HesWare™

EXPAND THE COMPUTER EXPERIENCE.™

$23.95

$49.95

$24.25

We Have the Software & Prices You Need!

Access
Beach-Head ..... (D)26.95
Neutral Zone ..... (D)26.95
CBS
Addition/Subtraction ..... (D)20.95
Multiplication/Division ..... (D)20.95
Linear Equations' ..... (D)20.95
Quadratic Equations ..... (D)20.95
Murder by the Dozen ..... (D)29.95
CDS
Frantic Freddy ..... (D)26.95
Pegasis ..... (D)26.95
Commodore
Assembler ..... (D)37.95
Bonus Pack ..... (D,T)16.95
CP/M2.2 System ..... (C,T)57.95
Easy Script ..... (D)44.95
Easy Mail 64 ..... (D)18.95
Easy Spell 64 ..... (D)18.95
Easy Calc 64 ..... (D)64.95
Easy Lesson/Quiz ..... (D)18.95
Easy Finance(1-5) ..... (D)18.95
Logo ..... (D)68.25
Pilot ..... (D)42.95
Pet Emulator ..... (D)18.95
Intro to Basic I ..... (T)18.95
Screen Editor ..... (D)18.95
Super Expander VSP ..... (D)37.95
The Manager ..... (D)59.95
Simons Basic ..... (C)45.95
Music Machine ..... (C)15.95
Music Composer ..... (D)18.95
Word/Name Machine ..... (D)18.25
Zortek and the Microchips ..... (T)21.95
Inventory Mgmt ..... (D)59.95
Payroll Checkwriting ..... (D) 59.95
Accts. Payable ..... (D)59.95
Accts. Receivable ..... (D)59.95
Gen. Ledger ..... (D)59.95
Nevada Cobol ..... (D)49.95
Nevada Fortran ..... (D)49.95
Magic Desk ..... (D)54.95
Zork (1-3) ..... (D)30.95
Suspended ..... (D)37.95
Starcross ..... (D)30.95
Deadline ..... (D)37.95
Continental
Home Accountant ..... (D)50.75
Tax Advantage ..... (D)40.60
F.C.M. ..... (D)40.60
Home Cataloger ..... (D)30.95
Creative Software
Pipes ..... (C)23.65
Decision Maker ..... (D)13.50
Household Finance ..... (D)16.90
Home Inventory ..... (D)13.50
Loan Analyzer ..... (D)13.50
Moondust ..... (C)23.65
Save New York ..... (C)23.65
Cyber
Cyberchess 34.95
Cymbal
English/Spanish ..... (D)34.95
Preschooler ..... (D)34.95
Science ..... (D)34.95
History/Geography ..... (D)34.95
Music Theory ..... (D)34.95
Mathematics ..... (D)34.95
Trivia ..... (D)34.95
Datasoft
Poo Yan ..... (D,T)20.30
Genesis ..... (D)27.00
O'Riley's Mine ..... (D)20.30
Dallas Quest ..... (D)27.00
Designware
Cryptocube ..... (D)27.25
Mathmaze ..... (D)27.25
Trap-A-Zoid ..... (D)27.25
Spellagraph ..... (D)27.25
Epyx
Pitstop ..... (C)27.00
Temple of Apshai ..... (D)27.00
Upper Reaches of Apshai ..... (D)13.55
Curse of Ra ..... (D)13.55
Sword of Fargoal ..... (D)20.30
Jumpman ..... (D,T)27.00
Gateway to Apshai ..... (D)27.00
Jumpman Jr. ..... (C)27.00
Oil Baron ..... (D)27.00
Lunar Outpost ..... (D)27.00
Finesse
Global Risk ..... (D)34.95
Hes Software
HES Magic Basic 42.50
The Factory ..... (D)24.25
Missing Links ..... (D)20.45
Type 'N Write ..... (C)20.45
Paint Brush ..... (C)17.00
Hes Mon 64 ..... (C)27.00
Turtle Graphics II ..... (C)40.60
Attack of the Mutant Camels ..... (C)20.45
Time Money Mgr. ..... (D)51.00
Omniwriter ..... (D)42.95
Synthesound 64 ..... (D)25.50
Omni Calc ..... (D)58.25
Multiplan ..... (D)79.95
Hes Writer 64 ..... (C)30.45
Gridrunner ..... (C)20.25
Benji Space Rescue ..... (D)30.45
The Pit ..... (C)20.25
64 Forth ..... (C)49.95
Turtle Toyland Jr. ..... (D,T)23.95
Chameleon ..... (D)17.00
Ghost Manor/Spikes Peak ..... (D)17.00
HES Games '84 ..... (D)27.25
Minnesota Fats Pool ..... (C)20.45
HES Modem I ..... (D)49.95
Orbyte
Stock Analysis ..... (D)41.95
Micro Prose
Floyd of the Jungle ..... (D)23.95
Hellcat Ace ..... (D)23.95
Solo Flight ..... (D)26.95
Nato Commander ..... (D)26.95
Sierra on Line
Frogger ..... (D,T)23.65
Homeword ..... (D)47.95
Sammy Lightfoot ..... (D)21.80
Threshold ..... (D)21.80
Lunar Leeper ..... (D)21.80
B.C. Quest for Tires ..... (D)25.50
Oil's Well ..... (D)21.80
Learning W/Leeper ..... (D)21.80
Sirus Software
Turmoil ..... (D)27.00
Snake Byte ..... (D)27.00
Squish 'Em ..... (D)23.65
Repton ..... (D)27.00
Bladc of Blackpoole ..... (D)27.00
Type Attack ..... (D)27.00
Way Out ..... (D)27.00
Critical Mass ..... (D)27.00
subLogic
Flight Simulator II ..... (D)44.95
Night Mission Pinball ..... (D)26.95
Spinnaker
Alf in the Color Caves ..... (C)27.95
Bubble Burst ..... (C)27.95
Juke Box ..... (C)27.95
Kidwriter ..... (D)23.95
Face Maker ..... (D,C)23.65
Hey Diddle Diddle ..... (D)20.30
Kindercomp ..... (C,D)20.30
Snooper Troops ..... (D)27.00
Fraction Fever ..... (C)23.65
Amazing Thing ..... (D)27.00
Alphabet Zoo ..... (D)23.65
Kids on Keys ..... (C,D)23.65
Delta Drawings ..... (C)27.25
Up for Grabs ..... (C)27.25
Cosmic Life ..... (C)23.65
Story Machine ..... (C)27.25
Trains ..... (D)27.95
Aerobics ..... (D)30.95
Scarborough
Songwriter ..... (D)35.95
Master Type ..... (C)35.95
Synapse
Ft. Apocalypse ..... (D,C)23.70
Shamus ..... (D,C)23.70
Blue Max ..... (D,C)23.70
Dimension X ..... (D,C)23.70
Zaxxon ..... (D)27.95
SSI
Combat Leader ..... (D)29.00
Knights of the Desert ..... (D)29.00
Computer Baseball ..... (D)29.00
Eagles ..... (D)29.00
Ring Side Seat ..... (D)29.00

Cx commodore

Commodore 64
Executive 64
1541 Disk Drive
1530 Datasette
MPS 801 Printer
1650 Automodem
1702 Color Monitor
1600 Modem
64 Prog. Ref. Guide

Diskettes
BASF 5¼" SSDD
2 Pak 5.50
BASF
Soft Box(10) ... 23.80

Joysticks
Pointmaster 10.95
Pointmaster Pro . 18.55
Fire Control 7.85
Wico
Command Ctrl . 19.50
Wico "Boss" ... 13.50
Wico Red Ball ... 20.75
Wico Power Grip 20.75

Cardco
Card+G 68.25
Card B 37.85
Cass. Interface .. 33.25
5 Slot Expansion . 60.25
Numeric Keypad 30.25

Koala
Touch Tablet . (D)79.95
GEMINI Printers
O'SULLIVAN Furniture
MAXTRON Datasettes

CALL FOR INFORMATION

Call for Atari, Apple & IBM-PC Software

To Order 800-527-8698 **In Texas 800-442-8717**

To order, call toll free or send by mail. We accept Visa, Master Card, American Express or Money Orders. Personal checks allow two weeks to clear. Add $2.00 for shipping with an additional $1.65 for C.O.D. orders. Other countries please include 10% with a minimum of $10.00. All prices are subject to change. For return authorization call (214) 759-8937.

SEND FOR FREE CATALOG

SoftWare Warehouse

P.O. Box 2511
Longview, Tx. 75606

any line Y, we will use PRINT LEFT$ (L$,Y).

Horizontal (X) positioning can be controlled using the TAB function. PRINT TAB(X) will begin printing in column X of the designated line. Remember that the first position on a line is column 0, the second is column 1, and so on.

Let's combine these two functions and print a red heart in the fifth column of the tenth line on the screen:

```
10 L$ = "{HOME}{21 DOWN}": PRINT "{CLR}"
                                    :rem 104
20 PRINT LEFT$(L$,10) TAB(5) CHR$(28) CHR
   $(115)                           :rem 142
```

Repeating Characters

If a character is to be printed several times, a FOR–NEXT loop is more efficient and takes less memory than repeating the character in the statement:

```
10 L$ = "{HOME}{21 DOWN}": PRINT "{CLR}"
                                    :rem 104
20 PRINT LEFT$(L$,11) TAB(3);       :rem 35
30 PRINT CHR$(30) CHR$(18);        :rem 155
40 FOR I = 1 TO 7: PRINT CHR$(32);: NEXT
                                    :rem 14
```

This program prints a green bar on the eleventh line, beginning in the third column. The CHR$ codes used are:

30 = GREEN
18 = REVERSE ON
32 = SPACE

A Simple Sun

With these four characteristics in mind, let's draw a simple picture using the PRINT statement: a child's representation of the sun.

We need to consider three elements in creating this picture: the individual characters needed, the color desired, and the location of each character. After consulting our chart of CHR$ codes, we find that the needed values are 109, 98, 110, and 113. As for color, let's be conventional and use CHR$(158)—yellow. Coordinates Y=10, X=10 should give us a beginning point roughly at the center of the VIC screen. (You'll need to adjust the TAB values if you want the sun to appear centered on a 64 screen.)

So our sun-drawing program will read:

```
10 L$ = "{HOME}{21 DOWN}": PRINT "{CLR}"
                                    :rem 104
20 PRINT LEFT$(L$,10) TAB(10);      :rem 80
30 PRINT CHR$(158) CHR$(109) CHR$(98) CHR
   $(110)                           :rem 115
40 PRINT TAB(11);                   :rem 182
50 PRINT CHR$(113)                  :rem 217
60 PRINT TAB(10);                   :rem 183
70 PRINT CHR$(110) CHR$(98) CHR$(109)
                                    :rem 135
```

In typing this and other programs, eliminate spaces between words. They add to readability, but use up memory.

Let's draw another picture, this time using a repeated character:

```
10 L$ = "{HOME}{21 DOWN}": PRINT "{CLR}"
                                    :rem 104
20 PRNT LEFT$(L$,10) TAB(7);        :rem 221
30 PRINT CHR$(156) CHR$(18) CHR$(169);: F
   OR I = 1 TO 3: PRINT CHR$(32);: NEXT
                                    :rem 166
40 PRINT CHR$(146) CHR$(169)        :rem 208
50 PRINT TAB(6);                    :rem 139
60 PRINT CHR$(156) CHR$(18) CHR$(169);: F
   OR I = 1 TO 3: PRINT CHR$(32);: NEXT
                                    :rem 169
70 PRINT CHR$(146) CHR$(169)        :rem 211
```

Voilà! A purple parallelogram!

The Basic Picture Elements

We now have all the basic elements necessary to print any picture:

- X, Y positioning of first line
- PRINTing one character
- PRINTing repeated characters
- X positioning of successive lines

Instead of using the TAB function to position all the lines, let's use a string that we'll define as T$ = "{DOWN}{21 LEFT}" for the VIC, or T$ = {DOWN}{39 LEFT} for the 64. By using the LEFT$ function with this string, we can position each new line in relationship to the preceding line. Let's look at our sun-drawing routine again and see how we would program it using T$. In this picture, when the first line is finished printing, the cursor will be in the blank space following the / (slash) character. We want to go down one space and move two spaces to the left to print the second line.

Likewise, after the second line prints, we'll move down one and left two to begin the third line. PRINT LEFT$(T$,3) will give us the correct positioning for both lines.

Writing The Subroutine

At last we are ready to formulate our general PRINT subroutine. This subroutine, used with DATA statements, will print the two pictures we have already created. By adding additional DATA statements, it can also be used to draw any other picture we might design.

```
100 REM***GENERAL PRINT SUB         :rem 104
110 READQ:PRINTLEFT$(L$,Q);         :rem 142
120 READQ:PRINTTAB(Q);              :rem 123
130 READQ:IFQ=0THEN170               :rem 79
140 IFQ<0THENHQ=ABS(Q):READQ:FORI=1TOHQ:P
    RINTCHR$(Q);:NEXT:GOTO130       :rem 62
150 IFQ>191THENPRINTLEFT$(T$,Q/100);:GOTO
    130                             :rem 162
160 PRINTCHR$(Q);:GOTO130            :rem 9
170 RETURN                          :rem 120
900 REM***DATA FOR SUN              :rem 242
```

This brand new $29.95 arcade/adventure game for the C64 is yours free when you join the 64 CLUB. In fact, this is just one of the 1st month's programs you receive when you join the 64 CLUB. Each month's diskette contains 10 programs in the areas of business, education, and recreation. Plus you also receive free consulting and software discounts. All this for only $40/year. If you would like further proof of the quality of our software, send $5 for a trial month which includes Quest for Power.

The 64 CLUB
1260 Oliver Ave.
San Diego, CA 92109

KIWISOFT PROGRAMS
FOR THE C-64

"VENUS" BY VELAZQUEZ

"There's nothing like it!"

- Paint it with PAINTPIC™ 16 colors, 160 x 200 screen, CAD — $39.95
- Print it with PRINTAPIC™ Most printers, hook rug, "Venus" — $44.95
- Do both with CADPIC™ Combines Paintpic + Printapic — $79.95
- Puzzle it with PUZZLEPIC™ Pictures, mosaics, mazes — $49.95
- More pictures with PICTUREDISKS Space, Cars, Planes, Fantasy, Animals — $24.95ea
- More Puzzlepic puzzles: PUZZLEDISKS Easy/Medium/Fiendishly Difficult! — $24.95ea
- Treasure the best of PAINTPIC ART 5 original paintings by "D.J.R." on disk — $89.00
- See our COMPUTER ART PRINTS (3 for $25) 20" x 24" "Room" "Storm" "Holy Land" — $8.95ea

SEND FOR OUR FREE COLOR BROCHURE
• DEALER ENQUIRIES WELCOME •

Add $2 p & h. Pay in advance by check or m/o or VISA, AMEX. California, add 6% sales tax.

Programs on diskette from
KIWISOFT PROGRAMS,
18003-L Skypark South, Irvine CA 92714.
ph 714 261 5114

INTRUDER ALARM
for VIC 20™ or Commodore 64™

Intruder Interface includes:

1. V100 Alarm Interface
2. Four Magnetic Switches
3. Piercing 95db Siren
4. 100 ft. twisted wire
5. Alarm software

* More accessories available

Total System Only $69.00

plus **$2.75** for postage and handling

Specify Computer and Memory
☐ VIC 20TM ☐ Commodore 64TM
☐ Disk ☐ Cassette

Name _____
Address _____
City _____ State _____ Zip _____
Phone Number ( ) _____
MC or VISA # _____
Expiration Date _____
Send Check, M.O., VISA/MC to:

VIN Systems (U.S.A.) P.O. Box 31
Boone, Iowa 50036

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

VIC - 20 / COM - 64
HOME / BUSINESS

	VIC	64
PRACTICALC PLUS (16K)(T)	43.95	
PRACTICALC 64 (T)		43.95
(disk version)		46.95
TOTL TEXT 2.5 (T)	26.95	29.95
(disk version)		32.50
CARDCO 3 SLOT EXPANSION BOARD	28.95	

16K MEMORY EXPANSION

NOW ONLY $46.50 SPECIAL SALE PRICE!

- ★ 14 Day Money Back Guarantee
- ★ Boosts VIC to 21K RAM
- ★ Top Quality, Fully Tested
- ★ 90 Day Warranty

for IMMEDIATE SHIPMENT on Credit Card Orders

Call: (303) 245-9012

9 AM - 9 PM MST Every Day

ASSEMBLY TECHNOLOGY
2692 Hwy 50 Suite 210
Grand Junction, CO 81503

Personal checks allow 3 weeks
Shipping & handling $2.50
Colorado Residents add Sales Tax
COD add $2.50

PB SYSTEMS

FOR COMMODORE 64™

MusicPlus	$19.95
GraphicAids	$19.95
UtilityPlus	$15.95
DeluxBanner	$12.95
MailLister	$12.95
Comal Proc & Func	$14.95
PUBLIC DOMAIN GAMES	
Copy Of Many	$12.95

TEXAS ADD SALES TAX
$1.50 SHIPPING ON ANY ORDER
MASTERCARD & VISA WELCOME
$1.00 FOR DETAIL OF PROGRAMS AND LIST OF PRODUCTS
Good for credit toward purchase
Prices good until August 31, 1984

PB SYSTEMS
Box 790816
Dept CM684 or CG684
Dallas, Texas 75379
(214) 991-0237

VIC-20 Protect your Investment C-64

- **Superior to cloth or vinyl.**
- **No more dirt, ashes, spilled liquid and dropped items crashing onto the keyboard.**

To order: check, money order, MC/Visa.

Card No. _____

Bank _____

Exp. date _____

Add $3.00 shipping & handling for each cover.
Kansas residents, add 3% sales tax.

Diversified Manufacturing
3517 S. Knight / Wichita, KS 67217
(316) 943-5516

```
910 DATA5,10,158,109,98,110,300,113,300,1
    10,98,109,0                          :rem 29
920 REM***DATA FOR PARALLELOGRAM         :rem 193
930 DATA10,10,156,18,169,-3,32,146,169,70
    0,156,18,169,-3,32,146,169,0         :rem 117
```

Lines 110 and 120 determine the screen coordinates. Line 130 checks to see if the picture is complete. We will use 0 to indicate the end of a picture since there is no valid CHR$(0). Line 140 handles repeated characters. Our data for a repeated character will be the negative of the number of characters we want to print (we use the negative value to avoid using a valid CHR$ code), followed by the character to be repeated. Line 150 controls tabulation of new lines.

Values desired are multiplied by 100 to take them beyond the range of valid CHR$ codes when creating the DATA. Line 160 prints one character. The whole procedure will continue until all DATA is read and printed and a 0 is encountered.

Add the following lines to complete the program:

```
10 L$="{HOME}{21 DOWN}":T$="{DOWN}
   {21 LEFT}"                            :rem 239
20 PRINT"{CLR}":FORI=1TO2:GOSUB100:NEXT:C
   LR:END                                :rem 99
```

Try running the entire program now, and see our two pictures print.

Adjustments For The 64

When this program is RUN on the 64, the shapes will be drawn off center on the screen. If you find this displeasing, change the definitions for L$ and T$ in line 10 to those mentioned earlier for the 64, then adjust the tabulation values in the DATA statements as appropriate for the 40-column screen.

What are the advantages of using this generalized subroutine rather than a straight PRINT?

The first and most immediate is a savings in memory. After you run this program, enter PRINTFRE(0). You will notice that the memory used is about 422 bytes. Beyond this initial requirement, the memory needed by additional DATA statements will be whatever is required to store the statements themselves. As the number of pictures printed increases, so does the savings in memory.

DATA Used As An Array

This subroutine could also use the DATA statements as if they were an array. An identifier could begin each set of DATA and could be used in the program to find the desired picture.

Here is a very simple program illustrating this use:

```
10 L$="{HOME}{21 DOWN}":T$="{DOWN}
   {21 LEFT}"                            :rem 239
20 PRINT"{CLR}{BLK}{2 SPACES}CHOOSE A SHA
   PE:","{2 DOWN}{PUR} TRIANGLE SQUARE","
   {2 DOWN} RECTANGLE"                   :rem 127
30 PRINT"{2 DOWN} PARALLELOGRAM"         :rem 25
40 INPUTA$:RESTORE                       :rem 183
50 READB$:IFB$="END"THENPRINT"{2 DOWN} NO
   T A VALID CHOICE":FORI=1TO5000:NEXT:GO
   TO20                                  :rem 107
60 IFB$<>A$THEN50                        :rem 206
70 PRINT"{CLR}{3 DOWN}"A$:GOSUB100:FORI=1
   TO4000:NEXT:GOTO20                    :rem 59
100 REM***GENERAL PRINT SUB              :rem 104
110 READQ:PRINTLEFT$(L$,Q);              :rem 142
120 READQ:PRINTTAB(Q);                   :rem 123
130 READQ:IFQ=0THEN170                   :rem 79
140 IFQ<0THENHQ=ABS(Q):READQ:FORI=1TOHQ:P
    RINTCHR$(Q);:NEXT:GOTO130            :rem 62
150 IFQ>191THENPRINTLEFT$(T$,Q/100);:GOTO
    130                                  :rem 162
160 PRINTCHR$(Q);:GOTO130                :rem 9
170 RETURN                               :rem 120
900 DATATRIANGLE,10,10,18,169,127,400,169
    ,-2,32,127,146,0                     :rem 219
910 DATARECTANGLE,10,10,30,18,-5,32,600,-
    5,32,146,0                           :rem 170
920 DATASQUARE,10,10,28,18,-3,32,400,-3,3
    2,400,-3,32,400,-3,32,146,0          :rem 162
930 DATAPARALLELOGRAM,10,10,159,18,169,-3
    ,32,146,169,700,18,169,-3,32,146,169,
    0                                    :rem 159
940 DATAEND                              :rem 142
```

In this example, the DATA is searched and selected as if from an array, but no extra storage space is used. As before, 64 owners may wish to adjust L$, T$, and the tabulation values to reflect their larger screens. You may also want to make the following change to properly align the menu.

```
20 PRINT"{CLR}{BLK}{2 SPACES}CHOOSE A SHA
   PE:":PRINT"{2 DOWN}{PUR} TRIANGLE":PRI
   NT"{2 DOWN}{RIGHT}SQUARE"
30 PRINT"{2 DOWN}{RIGHT}RECTANGLE":PRINT"
   {2 DOWN}{RIGHT}PARALLELOGRAM"
```

Load From Tape Or Disk

The third usage we will consider takes a slightly different approach to our subroutine. Instead of storing DATA as statements in the program, we will create a data file. Our subroutine will then use INPUT# statements instead of READ statements to retrieve the data and accomplish the desired printing.

The wonderful thing about this usage is that the length of the data file is not restricted to the amount of memory in your computer. It can be as large as your tape or disk allows. With this type of processing, you could print picture after picture without ever using any memory beyond the initial amount required to store the program.

An Adaptation For Tape

Here is an adaptation of our previous example program that illustrates the use of a data file as input from tape. Type, SAVE, and RUN the programs in the following order:

1. Type in Program 1 (the main program) and SAVE it on tape.

2. Type in Program 2 (creates the data file) and RUN it, leaving your tape positioned at the end of Program 1. The data file will then immediately follow the main program and can be used as input. Then SAVE Program 2 on tape after the data file, in case you want to create another data file.

3. Rewind tape, LOAD Program 1, and RUN it.

You will notice when you RUN this program that the amount of available memory does not appreciably decrease with each new picture. Only four pictures are included in this particular program, but 400 could have been stored in a data file and used as input without requiring any more memory in the main program.

Program 1:
PRINT Pictures From Data Files

```
10 REM***INPUT DATA FROM TAPE FILE AND DR
   AW SHAPES                          :rem 176
20 L$="{HOME}{21 DOWN}":T$="{DOWN}
   {21 LEFT}"                         :rem 240
30 OPEN1,1,0,"TAPE FILE"              :rem 57
40 INPUT#1,A$:IFA$="END"THENPRINT"{CLR}
   {3 DOWN} THAT'S ALL!":CLOSE1:CLR:END
                                      :rem 215
50 PRINT"{CLR}{3 DOWN}{BLK}"A$        :rem 241
60 GOSUB100                           :rem 119
70 PRINTLEFT$(L$,18)"AVAILABLE MEMORY:"FR
   E(0)                               :rem 207
80 FORI=1TO5000:NEXT:GOTO40           :rem 190
100 REM***GENERAL PRINT SUB USING TAPE IN
    PUT                               :rem 168
110 INPUT#1,Q:PRINTLEFT$(L$,Q)        :rem 71
120 INPUT#1,Q:PRINTTAB(Q)             :rem 52
130 INPUT#1,Q:IFQ=0THEN170            :rem 67
140 IFQ<0THENHQ=ABS(Q):INPUT#1,Q:FORI=1TO
    HQ:PRINTCHR$(Q);:NEXT:GOTO130     :rem 50
150 IFQ>191THENPRINTLEFT$(T$,Q/100);:GOTO
    130                               :rem 162
160 PRINTCHR$(Q);:GOTO130              :rem 9
170 RETURN                            :rem 120
```

Program 2: Data File Creator

```
10 REM***CREATE A TAPE FILE OF DATA
                                      :rem 177
20 OPEN1,1,1,"TAPE FILE"              :rem 57
30 R$=CHR$(13)                        :rem 204
40 READA$:PRINT#1,A$;R$               :rem 66
50 IFA$<>"END"THEN40                  :rem 129
60 CLOSE1:END                         :rem 30
900 DATATRIANGLE,10,10,18,169,127,400,169
    ,-2,32,127,146,0                  :rem 219
910 DATARECTANGLE,10,10,30,18,-5,32,600,-
    5,32,146,0                        :rem 170
920 DATASQUARE,10,10,28,18,-3,32,400,-3,3
    2,400,-3,32,400,-3,32,146,0       :rem 162
930 DATAPARALLELOGRAM,10,10,159,18,169,-3
    ,32,146,169,700,18,169,-3,32,146,169,
    0                                 :rem 159
940 DATAEND                           :rem 142
```

NEWS & PRODUCTS

Cheatsheet Products' plastic overlay for Easy Script *for the 64.*

VIC And 64 Keyboard Overlays

Cheatsheet Products has produced a series of plastic keyboard overlays for the VIC-20 and Commodore 64 which are quick reference aids for a variety of Commodore and third-party software.

The plastic templates, called *Leroy's Cheatsheets*, include program starting instructions, marked function keys, and reference tables, charts, or drawings.

Overlays are available for the following VIC-20 programs: *Programmer's Aid, Vicmon, Super Expander, VIC Typewriter, and VICTerm I* (all by Commodore), and *Quick Brown Fox* (Quick Brown Fox), *UMI Wordcraft 20* (United Microware Industries, Inc.), *HES Writer* and *HES VIC Forth* (both by Human Engineered Software), as well as *Graphic Printer* (Commodore 1515 and 1525) and a general BASIC overlay.

Overlays for the Commodore 64 programs include: *Term 64* and *Easy Script* (both by Commodore), *HES Writer* and *HESmon* (both by Human Engineered Software), *Quick Brown Fox* (Quick Brown Fox), *WordPro 3/Plus* (Professional Software), *Graphic Printer* (Commodore

1515 and 1525), *Paper Clip* (Batteries Included), and BASIC.

Each overlay sells for $3.95, plus $1 shipping and handling.

Cheatsheet Products
P.O. Box 8299
Pittsburgh, PA 15218
(412) 456-7420

VIC And 64 Cassette Backup

Creative Electronics has produced a cassette backup for the VIC-20 and Commodore 64 which works with any Commodore machine with a data cassette recorder. Written in machine language, the product allows you to protect your programs on tape by creating a reserve copy.

The cassette backup is available for $14.95.

Creative Electronics
P.O. Box 4253
Thousand Oaks, CA 91360
(805) 492-1506

Bible Study For Commodore

Seek-Bible is a group participation program for two to ten people which allows you to competitively search for particular sections of the Bible. It is available on tape for the VIC-20, PET, and Commodore 64 machines.

Using a system of challenges, search periods, solutions, and an internal timer, *Seek-Bible* controls the action of the search. A variety of menus are presented to the searchers from which to choose appropriate actions or supporting displays on the screen or printer.

Seek-Bible sells for $24.95 on tape. Two additional search tapes, *Seek-Bible 2* and *Seek-Bible 3* are available on tape for $12.95 each.

SEI Enterprises, Inc.
17 Serpi Road
Highland Mills, NY 10930
(516) 757-9783

Graphics Package For 64

Doodle!, a graphics program for the Commodore 64 designed to effectively use its color and high resolution capabilities, has been introduced by City Software.

The program works with either a track ball or joystick, and uses color-coded menus and visual cues as aids. Among its features are a "stamp" option which lets you create up to nine repeatable graphic symbols. On-screen designs may be printed on the Commodore 1525 printer and most other popular printers, including C. Itoh 8510 or NEC8023, most Epson or Star/Gemini, and most Okidata printers.

Doodle!'s copy function will duplicate, squeeze, reduce, enlarge, stretch, rotate, or reposition a drawing or part of a drawing on the screen. The save feature will store drawings on a disk for later recall or alterations. You may draw and erase with track ball or joystick in nine pen sizes at nine speeds. With the program, you can draw in 16 colors; choose drawing, fill, and background colors; change any color; or color over anything on the screen.

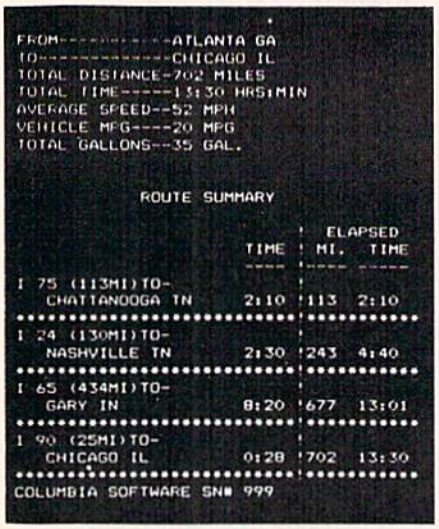
The suggested price for *Doodle!* is $39.95.

City Software Distributors, Inc.
735 West Wisconsin Avenue
Milwaukee, WI 53233
(414) 291-5125

VIC-20 Interface Board

A parallel interface board has been produced by Showtronics for the VIC-20 computer.

The circuit board has two parallel eight-bit ports. One port has buffering and the other can be used for input or output. A 6522 VIA and 2716/32 Eprom are also on the board. Several included routines allow you to use the SYS command for input or output to 2-20 pin connectors. The interface may be used for printer control, cordless telephone input, alarm input-output,



A sample screen from Columbia Software's Roadsearch.

and other applications.

The board sells for $139.00.

Showtronics
6780 Friars Road #105
San Diego, CA 92108
(619) 692-1212

Computerized Road Maps

Roadsearch and *Roadsearch-Plus* have been released by Columbia Software for the Commodore 64.

These computerized road maps simplify the calculation of driving routes, determining the shortest practical routes between cities in its data base.

Roadsearch contains a data base of 406 cities and road intersections and about 70,000 miles of interstate and major highways throughout the U.S. and Canada. *Roadsearch-Plus* offers these features as well as a road map development system that lets you customize maps. With *Roadsearch-Plus*, you can add up to 50 towns and 100 road segments anywhere in North America.

Roadsearch is available on disk for $34.95. *Roadsearch-Plus* sells for $74.95 on disk.

Columbia Software
P.O. Box 2235C
5461 March Hawk
Columbia, MD 21045
(301) 997-3100

64 Color Graphics

Flying Colors, a color graphics package designed for use with a standard joystick, has been released for the Commodore 64 by The Computer Colorworks.

With screen windows, you can pick such functions as thick and thin lines, circles and boxes of any size, erasures, and the ability to fill enclosed areas with a number of colors.

Text can be added to the screen, and a grid feature helps you align your pictures. You may save and retrieve the pictures that are created. *Flying Colors* also includes a program which allows you to produce a slide show with the graphic creations.

Flying Colors is available for $39.95 on disk.

The Computer Colorworks
3030 Bridgeway
Sausalito, CA 94965
(415) 331-3022

64 BASIC Aid

Don's Program House has developed a BASIC programming aid for the Commodore 64 called *BASIC-Plus*, which uses two-touch commands in order to

simplify entry and expand the number of BASIC commands.

Additional commands such as FIND, REF, CRUNCH, UNCRUNCH, DELETE, APPEND, ANALYZE, AUTO, and OLD are included. The two-touch commands (FSET, FLIST, FSAVE, and FLOAD) allow the programmer to define and control up to 64 functions by touching two function keys.

Written in machine language, *BASIC-Plus* is available on disk for $21.95.

Don's Program House
4817 Clipping Court
Louisville, KY 40222
(502) 228-0341

VIC, 64 Modem Adapter

The Universal Modem Adapter by Applied Systems and Products allows a VIC-20 or a Commodore 64 to be used with any type of phone for telecommunications through a VIC modem.

The adapter will support modular as well as non-modular phones, including one-piece electronic phones with built-in dialing. It does not require any modification to your phone or the modem. Nor do you need additional equipment, such as an RS-232 interface or acoustic coupler.

The modem adapter sells for $14.95, plus $2 for shipping.

Applied Systems and Products
1021-H West Bishop
Santa Ana, CA 92703
(714) 541-0233

Games For The 64

Bytes and Bits has released several new games for the Commodore 64.

Dungeons and Demons is a fantasy adventure game in which the player attempts to successfully move a character through 12 levels of a dungeon in search of a golden chalice. With a total of 1200 rooms to explore and more than 40 types of monsters to encounter, the player may choose to be a dwarf, warrior, halfling, thief, elf, or wizard. The game sells for $21.95 on disk and features three-dimensional graphics, multi-color sprite graphics, and sound.

Goombahtz is a computer version of a dice game for up to six players. The game uses six three-dimensional dice and flashes more than 100 messages of encouragement, sympathy, and rule applications as the game progresses. The computer will play against itself, against another player, or allow you to play the game while making sure all rules are followed. *Goombahtz* is available for $16.95 on disk or tape.

Crazy Conveyors is an action game that challenges you to race against time as you climb ladders, slide down fire poles, and cross unpredictable moving conveyors. The player gathers ammo while fending off enemies. A built-in screen creator allows you to expand the game up to 128 screens. Additional disks may be used for unlimited screens. The game uses custom characters, multi-color sprites, three-part harmony music, and works by joystick or keyboard.

Crazy Conveyors sells for $29.95 on disk.

Bytes and Bits
524 East Canterbury Lane
Phoenix, AZ 85022
(602) 942-1475

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

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

A Beginner's Guide To Typing In Programs

What Is A Program?

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

BASIC Programs

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

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

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	

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
```

Castle Dungeon

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:

Castle Dungeon—VIC Version, Loader

```
1 PRINTCHR$(147):POKE52,29:POKE56,29:CLR
                                    :rem 153
2 SV=36873:S1=SV+1:S2=SV+2:S3=SV+3:V=SV+5
  :SB=SV+6                          :rem 21
3 READL,N:IFN=-1THEN5               :rem 43
4 FORJ=0TON:READS:POKEL+J,S:NEXT:GOTO3
                                    :rem 52
5 POKESB,110:FORT=1TO500:NEXT       :rem 75
6 FORM=1TO3:READA,B,C,D,E           :rem 3
7 POKES1,A:POKES2,B:POKES3,C:FORJ=15TOEST
  EP-1:POKEV,J:FORT=1TOD:NEXT:NEXT:NEXT
                                    :rem 11
10 FORT=1TO2000:NEXT                :rem 232
11 PRINTCHR$(147):FORT=1TO500:NEXT
                                    :rem 113
12 POKESB,59:POKESV-4,242           :rem 248
13 FORT=1TO500:NEXT                 :rem 190
14 PRINTCHR$(144)"{UP}{RIGHT}FIND AND DEF
  USE THE"                          :rem 224
15 PRINT:PRINT" BOMBS HIDDEN IN THE"
                                    :rem 149
16 PRINT:PRINT" DUNGEON. DON'T FALL"
                                    :rem 56
17 PRINT:PRINT" INTO A PIT OR GET"
                                    :rem 233
18 PRINT"{DOWN}{RIGHT}EATEN BY A BEAST."
                                    :rem 78
19 PRINT"{DOWN}{RIGHT}PRESS THE 'L' KEY F
  OR"                               :rem 65
20 PRINT"{RIGHT}A LEVITATION SPELL."
                                    :rem 62
21 PRINT:PRINT"{RIGHT}YOU HAVE 5 MINUTES"
                                    :rem 19
22 PRINT"{DOWN}{RIGHT}TO COMPLETE YOUR":P
  RINT"{DOWN}{RIGHT}QUEST."         :rem 167
23 POKE143,VAL(MID$(TI$,5,2))       :rem 91
24 READCL:IFCL=-1THEN26             :rem 155
25 FORJ=CLTOCL+7:READCC:POKEJ,CC:NEXT:GOT
  O24                               :rem 139
26 PRINT:PRINT"(HIT ANY KEY TO BEGIN)"
                                    :rem 143
27 GETA$:IFA$=""THEN27              :rem 245
28 PRINT"{CLR}":FORJ=1TO3:POKESV+J,0:NEXT
  :S$="LO"+CHR$(34)+"D"+CHR$(34)+",8:"+C
  HR$(131)                          :rem 192
29 FOR I=1TOLEN(S$):POKE630+I,ASC(MID$(S$
  ,I)):NEXT:POKE198,I:END           :rem 98
30 DATA7878,20,114,64,73,73,32,110,85,73,
  110,85,64,73,112,64,75,85,64,73,85,73,
  110                               :rem 14
31 DATA7900,20,93,32,93,93,32,93,93,93,93
  ,93,64,73,107,64,32,93,32,93,93,93,93
                                    :rem 94
32 DATA7922,20,113,64,75,74,64,115,125,74
  ,75,74,64,75,109,64,73,74,64,75,125,74
  ,75                               :rem 34
33 DATA7792,17,85,64,73,112,64,110,85,64,
  75,64,114,64,112,32,32,112,64,75
                                    :rem 75
34 DATA7814,16,93,32,32,107,64,115,74,64,
  73,32,93,32,93,32,32,107,64       :rem 80
35 DATA7836,17,74,64,75,75,32,125,85,64,7
  5,32,75,32,74,64,75,109,64,73     :rem 216
40 DATA -1,-1                       :rem 102
41 DATA0,0,219,36,5,0,236,231,36,5,237,23
  1,226,100,0                       :rem 13
44 DATA7632,247,227,246,193,215,247,235,2
  35,7640,255,34,34,34,255,68,68,68
                                    :rem 162
45 DATA7648,255,231,195,129,129,131,199,2
  55,7656,255,191,95,64,90,186,255,255
                                    :rem 76
46 DATA7664,191,121,112,1,0,135,55,115,76
  72,255,239,247,231,195,195,231,255
                                    :rem 184
47 DATA7424,255,255,255,255,255,255,255,2
  55,7440,255,255,255,129,129,255,255,25
  5,7432                            :rem 205
48 DATA255,255,255,255,255,255,255,255,74
  48,255,253,251,247,143,207,175,255
                                    :rem 210
49 DATA7456,191,121,112,1,0,135,55,115,74
  64,255,255,231,0,0,231,255,255,-1
                                    :rem 98
```

Program 2:

Castle Dungeon—VIC Version, Main Program

```
1 PRINTCHR$(147);CHR$(144):S1=36874:S2=S1
  +1:S3=S1+2:S4=S1+3:V=S1+4:SB=S1+5:CL=S1
  -5                                :rem 255
2 C=30720:L=7680:MW=59:FC=0:PC=33:BT=8182
  :FV=15:FP=0:CS=0:POKESB,8:R=37154:AF=0:
  KF=0                              :rem 215
3 POKEV,15:POKECL,255:FORJ=LTOL+505:POKEJ
  +C,0:POKEJ,32:NEXT                :rem 236
4 READD:IFD=-1THEN6                 :rem 161
5 POKEL+D,MW:L=L+D:GOTO4            :rem 20
6 L=7680:FORJ=1TO9:READD:POKEL+D,34:NEXT
                                    :rem 19
7 FORJ=1TO46:READD:POKEL+D,33:NEXT
                                    :rem 172
8 FORJ=1TO3:GOSUB67:POKEB+L,63:NEXT:rem 4
9 FORJ=1TO9:GOSUB67:POKEB+L,36:NEXT
                                    :rem 11
10 PC=32:GOSUB67:POKEB+L,61         :rem 114
11 GOSUB67:POKEB+L,35               :rem 5
12 FORJ=1TO3:GOSUB67:POKEB+L,60:NEXT
                                    :rem 44
13 GOSUB67:M=B+L:TI$="000000"       :rem 106
14 POKER,127:JS=(PEEK(37137)AND28)OR(PEEK
  (37152)AND128):JS=ABS(JS-100)/4-7:POKE
  R,255                             :rem 129
15 IFTI$>"000500"THEN48             :rem 248
16 IFJS=6THENDR=-22                 :rem 153
17 IFJS=5THENDR=22                  :rem 108
18 IFJS=3THENDR=-1                  :rem 101
19 IFJS=11THENDR=1                  :rem 104
20 IFJS=7THENDR=0                   :rem 52
21 P=PEEK(DR+M):IFP=59THENDR=0      :rem 106
22 IFP=35THENCS=1:POKEBT,35:POKEBT+C,5:BT
  =BT+1                             :rem 123
23 IFP=36THEN78                     :rem 136
```

```
24 IFP=34ANDKF=ØTHENGOSUB73          :rem 211
25 IFP=61THENKF=1:POKE8181,61:POKE8181+C,
   5                                 :rem 240
26 GETL$:IFL$="L"THENLS=1:GOSUB82    :rem 30
27 IFP=6ØANDLS<>1THENFP=1            :rem 59
28 IFP=6ØANDLS=1THENFP=2:PIT=M+DR:rem 114
29 IFP=63THENAF=AF+1:POKEBT,63:POKEBT+C,5
   :BT=BT+1:GOSUB75                  :rem 77
3Ø IFDR=ØANDFC=1THEN44               :rem 86
31 POKEM,32:POKEM+C,7:POKEM+DR+C,7:POKEM+
   DR,58                             :rem 8Ø
32 POKEM+DR+C-22,7:POKEM+DR+C+22,7:POKEM+
   DR+C+1,7:POKEM+DR+C-1,7           :rem 37
33 POKEM+DR+C-23,7:POKEM+DR+C+23,7:POKEM+
   DR+C+21,7:POKEM+DR+C-21,7:FC=1:rem 189
34 IFP=35ORP=61THENGOSUB7Ø           :rem 15Ø
35 IFFP=1THENPOKEM+DR,6Ø:GOTO76      :rem 217
36 IFPS=1THENPOKEPIT,6Ø:PS=Ø         :rem 48
37 IFFP=2THENPS=1:FP=Ø               :rem 123
38 LS=Ø:IFDR=ØTHEN44                 :rem 218
39 POKES3,24Ø:FORT=1TO2:NEXT:POKES3,Ø
                                     :rem 95
4Ø IFDR=-22THENPOKEM+C+23,Ø:POKEM+C+22,Ø:
   POKEM+C+21,Ø:GOTO44               :rem 84
41 IFDR=1THENPOKEM+C-23,Ø:POKEM+C-1,Ø:POK
   EM+C+21,Ø:GOTO44                  :rem 198
42 IFDR=-1THENPOKEM+C-21,Ø:POKEM+C+1,Ø:PO
   KEM+C+23,Ø:GOTO44                 :rem 242
43 POKEM+C-23,Ø:POKEM+C-22,Ø:POKEM+C-21,Ø
                                     :rem 96
44 M=M+DR:FC=1:IFAF=3THEN46          :rem 138
45 GOTO14                            :rem 7
46 POKESB,27:FORT=1TO3ØØØ:NEXT:POKECL,24Ø
   :PRINTCHR$(147):PRINT" YOU SAVED THE C
   ASTLE"                            :rem 12
47 PRINT:PRINT:PRINT" PLAY AGAIN?":GOTO52
                                     :rem 121
48 POKES4,22Ø:FORJ=15TOØSTEP-.5:POKESB,12
   7:POKEV,J:FORT=1TO1Ø:NEXT         :rem 232
49 POKESB,42:FORT=1TO1Ø:NEXT:NEXT:POKES4,
   Ø                                 :rem 232
5Ø FORT=1TO5ØØØ:NEXT                 :rem 239
51 POKECL,24Ø:PRINTCHR$(147):PRINT"PLAY A
   GAIN? Y/N"                        :rem 81
52 GETA$:IFA$=""THEN52               :rem 241
53 IFA$="Y"THENRESTORE:GOTO1         :rem 45
54 IFA$<>"N"THEN52                   :rem 255
55 END                               :rem 65
56 DATAØ,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
   1,1,1,1,1,1,3,4,6,8,1,3,4,2,1,1,2,2,1,
   1,1,1,2                           :rem 25Ø
57 DATA1,2,1,1,1,2,2,4,2,4,2,1,2,5,2,2,1,
   1,2,1,3,2,1,2,3,2,2,3,4,2,1,2,1,5,2,2,
   1,2,4,5                           :rem 18
58 DATA1,1,1,1,1,1,2,3,2,1,1,1,1,2,1,1,1,
   1,7,3,6,2,3,1,2,1,1,2,1,1,1,1,2,1,1,1,
   1,2,3,1                           :rem 254
59 DATA2,6,2,2,4,2,1,2,1,2,3,3,4,2,7,1,2,
   1,1,1,1,1,1,1,1,1,4,1,2,1,1,2,1,12,2,3,2
   ,2,1,1,1                          :rem 61
6Ø DATA2,1,2,2,1,2,2,3,2,2,1,2,2,3,3,2,1,
   1,1,1,1,2,2,1,2,2,1,1,1,3,3,4,2,2,1,4,
   3,1,1,1                           :rem 254
61 DATA1,2,2,1,1,4,1,2,2,9,6,2,1,2,1,1,1,
   1,1,1,1,1,1,1,2,1,1,2,1,1,1,1,1,4,2,5,2,
   8,1,6                             :rem 173
62 DATA3,4,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
   1,1,1,1,1,1,1,1,1,-1              :rem 252
63 DATA67,72,1Ø2,127,14Ø,24Ø,28Ø,316,419
                                     :rem 227
64 DATA23,24,26,27,28,45,46,48,49,5Ø,82,8
   3,84,1Ø5,1Ø6,123,124,125,145,146,147,1
   62,163                            :rem 154
65 DATA184,185,195,196,217,218,3Ø1,3Ø2,32
   3,324,338,339,36Ø,361,441,442,443,445
                                     :rem 85
66 DATA463,464,465,466,467           :rem 81
67 B=INT(RND(1)*5Ø1)+Ø               :rem 169
68 IFPEEK(B+L)<>PCTHEN67             :rem 213
69 RETURN                            :rem 79
7Ø POKES3,185:POKES2,2Ø2:POKES1,2Ø2:FORJ=
   15TOØSTEP-.5:POKEV,J:NEXT:POKES3,21Ø
                                     :rem 74
71 POKES1,232:POKES2,232:FORJ=15TOØSTEP-.
   Ø5:POKEV,J:NEXT                   :rem 26
72 FORJ=ØTO2:POKES1+J,Ø:NEXT:POKEV,15:RET
   URN                               :rem 126
73 POKEM,32:POKEM+DR,37:POKES4,13Ø:FORJ=1
   5TOØSTEP-1:POKEV,J:NEXT:POKES4,Ø:POKEM
   +DR,34                            :rem 179
74 POKEM,58:POKEV,15:DR=Ø:FORT=1TO5ØØ:NEX
   T:RETURN                          :rem 188
75 POKES3,22Ø:FORT=1TO5Ø:NEXT:POKES3,Ø:RE
   TURN                              :rem 17Ø
76 FORJ=254TO18ØSTEP-.5:POKES2,J:POKES3,J
   :POKEV,FV:FV=FV-.1:NEXT:POKES2,Ø:POKES
   3,Ø                               :rem 23
77 POKECL,24Ø:PRINTCHR$(147):PRINT" YOU F
   ELL INTO A PIT":GOTO81            :rem 18
78 FORG=19ØTO235:POKES4,G:FORT=1TO1Ø:NEXT
   :NEXT:FORG=235TO22ØSTEP-1:POKES4,G:FOR
   T=1TO2Ø                           :rem 72
79 NEXT:NEXT:FORJ=15TO5STEP-.1:POKEV,J:NE
   XT:POKES4,Ø:POKEV,15:IFCS=1THEN24
                                     :rem 131
8Ø POKECL,24Ø:PRINTCHR$(147):PRINT" YOU L
   OST TO A BEAST!!"                 :rem 124
81 POKESB,27:FORT=1TO3ØØØ:NEXT:GOTO51
                                     :rem 93
82 FORI=ØTO3:FORJ=ØTO15:POKEV,J:N=18Ø+I*J
   :POKES3,N:NEXTJ,I:POKES3,Ø:RETURN
                                     :rem 63
```

Program 3:

Castle Dungeon—64 Version

```
Ø GOSUB 5ØØØ:GOSUB15ØØ:GOSUB2ØØ:GOSUB1ØØØ
  :GOSUB11ØØ:GOSUB13ØØ::GOSUB1Ø1Ø:rem 1Ø8
1 GOSUB 114Ø                         :rem 119
5 GOSUB 3ØØØ:GOSUB 4ØØØ:GOSUB 15ØØ:TI$="Ø
  ØØØØØ"                             :rem 18
7 GOSUB 2Ø:GOTO 7                    :rem 195
1Ø JOY=Ø:JY=PEEK(56321Ø):J2=15-(JYAND15):O
   NJ2 GOTO11,12,12,13,13,13,13,14:RETURN
                                     :rem 25Ø
11 JOY=-4Ø:RETURN                    :rem 6Ø
12 JOY=4Ø:RETURN                     :rem 16
13 JOY=-1:RETURN                     :rem 11
14 JOY=1:RETURN                      :rem 223
2Ø GOSUB 1Ø                          :rem 67
25 GET A$:IFA$="L" THEN GOSUB 491:LEVITAT
   E=-1                              :rem 39
27 IF TI$>"ØØØ3ØØ"THEN A$="YOUR TIME RAN
   {SPACE}OUT":GOTO 6ØØ              :rem 211
3Ø TEMP=PEEK(MAN+JOY)-4Ø:IFTEMP<ØTHENLEVI
   TATE=Ø:GOTO17Ø                    :rem 7Ø
4Ø ON TEMP GOSUB 5Ø,6Ø,8Ø,9Ø,11Ø,13Ø,14Ø,
   15Ø,16Ø:GOTO 17Ø                  :rem 166
5Ø GOSUB 22Ø:JOY=Ø:RETURN            :rem 44
6Ø IFNOTLEVITATETHENPOKEMAN,BLANK:GOSUB12
   5:A$="YOU FELL INTO A PIT":GOTO1Ø5
                                     :rem 21
```

```
70 K=1:RETURN                                   :rem 58
80 KEY=-1:POKE1303,43:POKE55575,1:GOSUB 4
   01:RETURN                                    :rem 233
90 GOSUB 320:IF NOT SWRD THEN A$="YOU LOS
   T TO A BEAST":GOTO 600                       :rem 40
100 GOSUB 360:RETURN                            :rem 196
105 GOSUB 220:FORTD=1TO200:NEXT:GOTO 600
                                                :rem 136
110 BOMB=BOMB-1:POKE 1823+40*BOMB,45:POKE
    56095+40*BOMB,1:IF BOMB=0 THEN 500
                                                :rem 212
120 RETURN                                      :rem 115
125 GOSUB260:GOSUB220:FORTD=1TO200:RETURN
                                                :rem 116
130 RETURN                                      :rem 116
140 IF NOT KEY THEN GOSUB 220:JOY=0:RETUR
    N                                           :rem 244
150 RETURN                                      :rem 118
160 SWRD=-1:GOSUB 401:POKE 1423,49:POKE55
    695,1:RETURN                                :rem 123
170 XH=0:POKE MAN,BLANK:MAN=MAN+JOY:POKE
    {SPACE}MAN,FIGURE:IF K=1 THEN POKE MA
    N,42:K=1                                    :rem 62
180 Q=MAN-1024:Y=INT(Q/40):X=(Q-Y*40)*8:I
    FX>255THENXH=INT(X/256):X=X-XH*256
                                                :rem 240
190 Y=Y*8+32:POKE HXSPRITE,XH:POKE LXSPRI
    TE,X:POKE YSPRITE,Y:BLANK=32 :rem 241
195 IFK=1THENK=0:BLANK=42                       :rem 19
196 RETURN                                      :rem 128
200 FOR I=1TO4:READ NO(1,I),DUR(1,I):NEXT
                                                :rem 120
201 FOR I=1TO7:READ NO(2,I),NL(2,I),DUR(2
    ,I):NEXT                                    :rem 60
205 RETURN                                      :rem 119
210 DATA 8,100,7,50,7,50,12,300,14
                                                :rem 105
211 DATA 2,24,100,2,24,100,2,24,100,2,163
    ,100,2,24,100,2,163,100,3,35 :rem 235
220 REM BUMP SOUND                              :rem 53
230 POKE H1,15:POKE L1,10                       :rem 66
240 POKE VOLUME,15:POKE ATTDEC,20:POKE SU
    STREL,5:POKE WAVE,NOISE                     :rem 29
250 FOR TD=1TO10:NEXT:POKE WAVE,NOISE-1:F
    OR TD=1TO2:NEXT:RETURN                      :rem 52
260 REM FALLING SOUND                           :rem 2
270 POKE ATTDEC,10:POKE SUSTREL,10:POKE V
    OLUME ,15                                   :rem 5
280 FOR I=200 TO 100 STEP-1                     :rem 103
290 POKE H1,I:POKE H2,100:POKE WAVE,TRIAN
    GLE                                         :rem 118
300 FOR TD=1TO3:NEXT:POKE WAVE,TRIANGLE-1
    :FOR TD=1TO3:NEXT:NEXT I                    :rem 131
310 RETURN                                      :rem 116
320 REM LION SOUND                              :rem 52
330 POKE ATTDEC,100:POKE SUSTREL,100:POKE
     VOLUME,15                                  :rem 98
340 POKE H1,2 :POKE H2,10:POKE WAVE,NOISE
    :FORTD=1TO1000:NEXT:POKE WAVE,NOISE-1
                                                :rem 251
350 FORTD=1TO100:NEXT:RETURN                    :rem 76
360 REM VICTORY SOUND                           :rem 54
370 POKE ATTDEC,100:POKE SUSTREL,100:POKE
     VOLUME,15                                  :rem 102
380 FORI= 1 TO 4:POKE H1,NO(1,I):POKE H2,
    100:POKE WAVE,SAW                           :rem 5
390 FOR TD=1 TO DUR(1,I):NEXT:POKE WAVE,S
    AW-1:FOR TD=1 TO DUR(1,I):NEXT:NEXT
                                                :rem 54
400 RETURN                                      :rem 116
401 POKE ATTDEC,14:POKE SUSTREL,201
                                                :rem 100
402 POKE H1,120:POKE WAVE,SAW:FORTD=1TO10
    0:NEXT:POKEWAVE,SAW-1                       :rem 149
403 FOR TD=1TO200:NEXT:RETURN                   :rem 76
410 REM WIN SOUND AND OPENING SOUND
                                                :rem 92
420 POKEVOLUME,15:POKE ATTDEC,17:POKE SUS
    TREL,140                                    :rem 61
430 FOR I= 1TO7:POKE H2,NO(2,I):POKE H1,N
    L(2,I):POKE WAVE ,SAW                       :rem 6
440 FOR TD=1TODUR(2,I):NEXT:POKEWAVE,SAW-
    1:FORTD=1TODUR(2,I):NEXT:NEXT:RETURN
                                                :rem 78
450 REM LOSE SOUND                              :rem 57
460 POKE ATTDEC,100:POKE SUSTREL,120:POKE
     VOLUME,15                                  :rem 104
470 POKE H1,4:POKE L1,48:POKE WAVE,SAW:FO
    RTD=1TO500:NEXT:POKE WAVE,SAW-1
                                                :rem 189
480 FOR TD=1TO400:NEXT:POKE H1,2:POKEL1,2
    4:POKEWAVE,SAW:FORTD=1TO900:NEXT
                                                :rem 128
485 POKEWAVE,SAW-1:FOR TD=1TO500:NEXT:RET
    URN                                         :rem 106
490 REM LEVITATE SOUND                          :rem 104
491 POKE ATTDEC,100:POKE SUSTREL,250:POKE
     VOLUME,15                                  :rem 112
492 FOR A=10TO20:FOR B=20 TO27:POKE H1,A+
    B:POKE L1,40:POKE WAVE,TRIANGLE
                                                :rem 200
493 FOR TD=1 TO2 :NEXT:POKE WAVE,TRIANGLE
    -1:FOR TD=1 TO 1:NEXT:NEXT:NEXT
                                                :rem 189
494 RETURN                                      :rem 129
500 POKE53269,0:GOSUB 410:POKE BACKGROUND
    ,1:FOR TD=1 TO 2000:NEXT                    :rem 35
510 PRINT"{CLR}":POKE 251,50:PRINT"{BLK}
    {7 DOWN}{11 RIGHT}CONGRATULATIONS !"
                                                :rem 123
520 PRINT"{2 DOWN}{9 RIGHT}YOU SAVED THE
    {SPACE}CASTLE"                              :rem 156
530 GOTO 700                                    :rem 104
600 POKE 53269,0:POKE BACKGROUND,1:GOSUB
    {SPACE}450:FOR TD=1 TO 2000:NEXT
                                                :rem 40
610 PRINT"{CLR}{8 DOWN}{BLK}"SPC(((40-(LE
    N(A$)))/2))A$                               :rem 41
620 PRINT"{3 DOWN}{10 RIGHT}YOU LOST THE
    {SPACE}CASTLE!"                             :rem 187
630 GOTO 710                                    :rem 106
700 POKE251,0:FORTD=1TO1500:NEXT:PRINT"
    {2 DOWN}{14 RIGHT}THIS TIME" :rem 237
710 FORTD=1TO500:NEXT:PRINT"{5 DOWN}
    {6 SPACES}HIT Y FOR FURTHER ADVENTURE
    S"                                          :rem 220
720 GET ANS$:IFANS$=""THEN720                   :rem 151
730 IF ANS$="Y"THEN POKE BACKGROUND,0:GOT
    O 5                                         :rem 75
740 POKE251,0:END                               :rem 207
1000 FORL=679TO703:READA:POKEL,A:NEXT:POK
     E806,167:POKE807,2:POKE251,40:RETURN
                                                :rem 40
1010 PRINTCHR$(144)"{CLR}{5 DOWN}{WHT}
     {2 SPACES}FIND THE BOMBS HIDDEN IN T
     HE DUNGEON."                               :rem 66
1020 PRINT"{DOWN}{2 RIGHT}DON'T FALL INTO
      A PIT OR GET EATEN BY "                   :rem 75
1030 PRINT"{2 RIGHT}A BEAST. PRESS THE 'L
       KEY FOR A "                              :rem 198
```

```
1040 PRINT"{DOWN}{2 SPACES}LEVITATION SPE
     LL. YOU HAVE 3 MINUTES"      :rem 205
1050 PRINT"{DOWN}{2 SPACES}TO FULFILL YOU
     R QUEST."                    :rem 104
1060 PRINT:PRINT:PRINT"{9 SPACES}(HIT BUT
     TON TO BEGIN)"               :rem 64
1065 GOSUB 10:IF JY<>111 THEN 1065:rem 33
1067 RETURN                       :rem 174
1070 DATA72,138,72,152,72,166,251,240,8,1
     60,255,136,208,253,202       :rem 81
1080 DATA 208,248,104,168,104,170,104,76,
     202,241                      :rem 53
1100 CM=14336:CS=53248:POKE56334,PEEK(563
     34)AND254:POKE1,PEEK(1)AND251
                                  :rem 254
1110 FORI=0TO1023:POKECM+I,PEEK(CS+I):NEX
     T                            :rem 91
1120 FORI=CM+320TOCM+320+95:READA:POKEI,A
     BS(A-255):NEXT               :rem 44
1130 POKE1,PEEK(1)OR4:POKE56334,PEEK(5633
     4)OR1:RETURN                 :rem 206
1140 POKE53272,(PEEK(53272)AND240)OR14
                                  :rem 93
1150 RETURN                       :rem 167
1160 DATA247,227,246,193,215,247,235,235,
     255,34,34,34,255,68,68,68     :rem 7
1170 DATA255,231,195,129,129,131,199,255,
     255,191,95,64,90,186,255,255:rem 163
1180 DATA191,121,112,1,0,135,55,115,255,2
     39,247,231,195,195,231,255   :rem 19
1190 DATA255,255,255,255,255,255,255,255,
     255,255,255,129,129,255,255,255
                                  :rem 57
1200 DATA255,255,255,255,255,255,255,255,
     255,253,251,247,143,207,175,255
                                  :rem 38
1210 DATA191,121,112,1,0,135,55,115,255,2
     55,231,0,0,231,255,255       :rem 44
1300 PRINT CHR$(147):POKE BACKGROUND,0:PO
     KE BRDER,0                   :rem 24
1305 READ L,N:IFN=-1THEN GOSUB410:FORTD=1
     TO2500:NEXT:RETURN           :rem 197
1310 FORJ=0TON:READS:POKEL+J,S:NEXT:GOTO1
     305                          :rem 91
1360 DATA1234,17,85,64,73,112,64,110,85,6
     4,75,64,114,64,112,32,32,112,64,75
                                  :rem 160
1370 DATA1274,16,93,32,32,107,64,115,74,6
     4,73,32,93,32,93,32,32,107,64
                                  :rem 174
1380 DATA1314,17,74,64,75,75,32,125,85,64
     ,75,32,75,32,74,64,75,109,64,73
                                  :rem 45
1420 DATA1393,20,114,64,73,73,32,110,85,7
     3,110,85,64,73,112,64,75,85,64,73
                                  :rem 119
1421 DATA85,73                    :rem 229
1430 DATA 110                     :rem 116
1440 DATA1433,20,93,32,93,93,32,93,93,93,
     93,93,64,73,107,64,32,93,32,93,93
                                  :rem 142
1441 DATA93,93                    :rem 232
1450 DATA1473,20,113,64,75,74,64,115,125,
     74,75,74,64,75,109,64,73,74,64,75,12
     5                            :rem 83
1455 DATA 74,75,-1,-1             :rem 0
1500 WAVE=54276:NOISE=129:TRIANGLE=17:VOL
     =54296:SOUND=54272:ATTDEC=54277
                                  :rem 175
1505 BACKGROUND=53281:BRDER=53280:LEVITAT
     E=0:SWRD=0:KEY=0:BOMB=3:LOIN=44
                                  :rem 42
1510 SAW=33:SUSTREL=54278:H1=54273:H2=542
     72:FIGURE=40:BLANK=32        :rem 211
1515 HXSPRITE=53264:LXSPRITE=53248:YSPRIT
     E=53249:POKE 53271,1:POKE 53277,1
                                  :rem 192
1517 POKE 53269,1:FORI=832TO896:POKEI,255
     :NEXT:POKE2040,13            :rem 163
1518 POKE 53287,1:POKE53275,1     :rem 48
1520 FOR I=SOUND TO SOUND+28:POKEI,0:NEXT
     :POKE VOLUME,15:RETURN       :rem 243
3000 I=49152:IF PEEK(I+2)=216THENSYS49160
     :GOTO 3025                   :rem 120
3010 READ A:IF A=256 THENSYS49160:GOTO 30
     25                           :rem 116
3020 POKE I,A:I=I+1:GOTO 3010     :rem 70
3025 FORI=1024TO1062:POKEI,41:POKEI+960,4
     1:NEXT                       :rem 51
3027 FORI=55296TO56295:POKEI,0:NEXT:RETUR
     N                            :rem 147
3030 DATA 1,0,216,255,255,255,40  :rem 26
3040 DATA 0,169,41,133,251,169,40 :rem 81
3050 DATA 133,253,169,4,133,252,133
                                  :rem 182
3060 DATA 254,169,147,32,210,255,162
                                  :rem 239
3070 DATA 0,160,0,169,41 ,145,253 :rem 28
3080 DATA 200,192,39,208,249,24,165
                                  :rem 194
3090 DATA 253,105,40,133,253,144,2
                                  :rem 126
3100 DATA 230,254,232,224,23,208,229
                                  :rem 225
3110 DATA 160,0,169,4,145,251,169 :rem 84
3120 DATA 255,141,15,212,169,128,141
                                  :rem 230
3130 DATA 18,212,173,27,212,41,3  :rem 24
3140 DATA 133,173,170,10,168,24,185
                                  :rem 184
3150 DATA 0,192,101,251,133,170,185
                                  :rem 174
3160 DATA 1,192,101,252,133,171,24
                                  :rem 122
3170 DATA 185,0,192,101,170,133,253
                                  :rem 178
3180 DATA 185,1,192,101,171,133,254
                                  :rem 182
3190 DATA 160,0,177,253,201,41,208
                                  :rem 129
3200 DATA 18,138,145,253,169,32,145
                                  :rem 191
3210 DATA 170,165,253,133,251,165,254
                                  :rem 28
3220 DATA 133,252,76,62,192,232,138
                                  :rem 189
3230 DATA 41,3,197,173,208,189,177
                                  :rem 153
3240 DATA 251,170,169,32,145,251,224
                                  :rem 234
3250 DATA 4,240,26,138,10,168,162 :rem 83
3260 DATA 2,56,165,251,249,0,192  :rem 41
3270 DATA 133,251,165,252,249,1,192
                                  :rem 190
3280 DATA 133,252,202,208,238,76,62
                                  :rem 191
3290 DATA 192,169,1,160,0,153,0   :rem 237
3300 DATA 216,153,0,217,153,0,218 :rem 72
3310 DATA 153,0,219,200,208,241,96,256
                                  :rem 73
```

```
4000 REM CHARACTER PLACEMENTS      :rem 33
4010 T=32:Q=42:FOR I=1 TO 7:GOSUB 4100:NE
     XT                            :rem 147
4020 Q=44:FOR I=1 TO 6:GOSUB 4100:NEXT
                                   :rem 101
4030 Q=45:FOR I=1 TO 3:GOSUB 4100:NEXT
                                   :rem 100
4040 Q=43:FOR I=1 TO 2:GOSUB 4100:NEXT
                                   :rem 98
4050 Q=47:FOR I=1 TO 5             :rem 114
4055 GOSUB 4100                    :rem 19
4060 IF(PEEK(PLACE-40)<>32)OR(PEEK(PLACE+
     40)<>32)THENPOKEPLACE,BLANK:GOTO4055
                                   :rem 200
4070 NEXT                          :rem 10
4080 Q=49:GOSUB 4100               :rem 70
4090 Q=FIGURE:GOSUB 4100:MAN=PLACE:rem 84
4095 T=41:Q=32:FORI=1TO50:GOSUB 4100:NEXT
     :RETURN                       :rem 231
4100 X=(RND(1)*36+2):Y=INT(RND(1)*20+3):P
     L=Y*40+X+1024:IFPEEK(PLACE)<>TTHEN41
     00                            :rem 50
4110 POKE PLACE,Q:RETURN           :rem 241
5000 PRINT"{CLR}{11 DOWN}{14 RIGHT}{RVS}P
     LEASE WAIT"                   :rem 123
5010 RETURN                        :rem 166
```

Revenge Of Cyon

See article on page 56.

BEFORE TYPING...

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

Program 1: VIC Version

```
1 DIMJS(2,2):POKE37139,0:DD=37154:PA=3713
  7:PB=37152:V$=" SECONDS"         :rem 180
2 FORI=0TO2:FORJ=0TO2:READJS(J,I):NEXT:NE
  XT:FORT=0TO10:READB$(T):NEXT     :rem 6
3 CO=30720:SC=7680:BG=1:BE=7955    :rem 233
4 POKE36879,255:PRINT"{CLR}{6 DOWN}
  {3 RIGHT}{RVS}{PUR}REVENGE OF CYON{BLU}
  {OFF}":PRINT"{5 DOWN}{RIGHT}SKILL LEVEL
   (1-6) ?";                       :rem 163
5 GETA$:IFA$=""THEN5               :rem 141
6 D=VAL(A$):IFD<1ORD>6THEN4        :rem 134
7 PRINT"{CLR}":POKE36879,8:A=32:A1=32:A2=
  32                               :rem 51
8 FORI=0TO220:POKE38796+I,5:NEXT:POKE3877
  5,2:FORP=1TO20:POKE38778+P,5:NEXT:POKE3
  8767,6                           :rem 54
9 PRINT"{HOME}"BG:J=0:K=0:VB=0:POKE7708,4
  6:POKE7818,46:POKE7738,46:POKE7727,81:T
  H=0:S=0                          :rem 94
10 POKE8076,78:POKE8077,99:POKE8078,77:PO
   KE8101,77:POKE8102,100:POKE8103,78
                                   :rem 69
11 POKE8104,99:POKE8083,78:POKE8084,99:PO
   KE8063,78:POKE8064,77:POKE8086,103:POK
   E8109,77                        :rem 44
12 POKE8110,79:POKE8089,79:POKE8068,78:PO
   KE8069,99:POKE8070,77:POKE8093,80:POKE
   8132,77                         :rem 2
13 POKE8133,100:POKE8134,100:POKE8135,100
   :POKE8136,100:POKE8137,78:POKE8115,103
                                   :rem 211
14 POKE8094,78:POKE8073,78:POKE8074,77:PO
   KE8075,78:POKE8112,108:POKE8113,123
                                   :rem 122
15 POKE8152,32:POKE8055,88:POKE8047,93:PO
   KE7782,46:POKE7901,46:POKE7691,46:POKE
   8025,87                         :rem 241
16 POKE38745,4                      :rem 1
17 POKE36878,15:FORL=1TO5:FORM=180TO235ST
   EP2:POKE36876,M:FORN=1TO10:NEXT:rem 37
18 NEXT:POKE36876,0:FORM=1TO50:NEXT:NEXT:
   POKE36878,0:PRINT"{HOME}{3 SPACES}"
                                   :rem 68
19 FORN=1TO2:POKE8115-N,46:FORQ=1TO120:NE
   XT:POKE8115-N,32:NEXT:LK=0:TI$="000000
   "                               :rem 81
20 POKEBE+VB,A                     :rem 72
21 POKEDD,127:S3=-((PEEK(PB)AND128)=0):PO
   KEDD,255                        :rem 72
22 P=PEEK(PA):S1=-((PAND8)=0):S2=((PAND16
   )=0):S0=((PAND4)=0)             :rem 127
23 FR=-((PAND32)=0):X=S2+S3:Y=S0+S1:IFLK>
   8164THEN60                      :rem 11
24 C=C+1:IFFR=1THENTH=TH+1         :rem 84
25 POKELK+PO+CO,D1:IFA1<>43THENPOKELK+PO,
   A1                              :rem 206
26 IFBG>6ANDJ<>1THENPOKELK+PO+30726,D2:IF
   A2<>43THENPOKELK+PO+6,A2        :rem 38
27 IFK<>1THENLK=INT(RND(1)*150)+SC:rem 20
28 POKE8025,81:POKE36878,0:IFFR=1ANDTH<=7
   THENGOSUB46                     :rem 217
29 K=1:PO=INT(RND(1)*D)+1:IFTH>7THENFR=0
                                   :rem 180
30 POKE36878,0:VB=VB+JS(X+1,Y+1)+(VB>220)
   *22-(VB<-264)*22                :rem 112
31 A=PEEK(BE+VB):POKEBE+VB,43:IFC=2THENLK
   =LK+22:C=0                      :rem 118
32 A1=PEEK(LK+PO):D1=PEEK(LK+PO+CO)
                                   :rem 107
33 IFBG>6ANDJ<>1THENA2=PEEK(LK+PO+6):D2=P
   EEK(LK+PO+30726)                :rem 194
34 IFS<>1THENPOKELK+PO+CO,7:POKELK+PO,90
                                   :rem 50
35 A$="SHOTS":IFTH=1THENA$="SHOT " :rem 5
36 PRINT"{HOME}";:XC=BE+VB:IFXC=LK+POANDF
   R=1THENPRINTTH;A$:PRINTRIGHT$(TI$,2)V$
   :GOTO49                         :rem 234
37 IFBG>6ANDJ<>1THENPOKELK+PO+30726,7:POK
   ELK+PO+6,90                     :rem 205
38 IFBG>6ANDJ<>1ANDXC=LK+PO+6ANDFR=1THENP
   RINTTH;A$:PRINTRIGHT$(TI$,2)V$:GOSUB41
                                   :rem 111
39 POKE8025,87:IFJ=1ANDS=1THENBG=BG+1:GOT
   O47                             :rem 210
40 POKESC+QW,32:QW=INT(RND(1)*176)+44:POK
   ESC+QW,46:GOTO20                :rem 230
41 POKE36878,15:FORW=1TO6:FORY=220TO254:P
   OKE36876,Y:NEXT:NEXT:POKELK+PO+CO+6,1
                                   :rem 177
42 POKELK+PO+6,43                  :rem 234
43 FORZ=22TOLK-SCSTEP22:G=PEEK(LK+PO-Z+6)
   :G1=PEEK(LK+PO-Z+CO+6):POKELK+PO-Z+CO+
   6,7                             :rem 160
44 POKELK+PO-Z+6,90:FORH=1TO70:NEXT:POKEL
   K+PO-Z+CO+6,G1:POKELK+PO-Z+6,G:NEXT
                                   :rem 128
45 TH=0:J=1:RETURN                 :rem 126
46 POKE36878,15:POKE36876,244:FORFV=1TO70
   :NEXT:POKE36878,0:RETURN        :rem 21
```

```
47 IFBG<=11THEN7                          :rem 196
48 GOTO76                                 :rem 18
49 IFBG>6ANDJ<>1THENPOKELK+PO+CO+6,7:POKE
   LK+PO+6,90                             :rem 193
50 S=1:POKE36878,15:FORL=1TO15:FORM=250TO
   240STEP-1:POKE36876,M:NEXT             :rem 147
51 FORM=240TO250:POKE36876,M:NEXT:POKE368
   76,0:NEXT:POKE36878,0:POKELK+PO+CO,1:T
   H=0                                    :rem 88
52 POKELK+PO,43                           :rem 138
53 FORZ=22TOLK-SCSTEP22:G=PEEK(LK+PO-Z):G
   1=PEEK(LK+PO-Z+CO):POKELK+PO-Z+CO,7
                                          :rem 126
54 POKELK+PO-Z,90:FORH=1TO70:NEXT:POKELK+
   PO-Z+CO,G1:POKELK+PO-Z,G:NEXT          :rem 94
55 POKE36876,0                            :rem 3
56 IFBG<7THENBG=BG+1:GOTO7                :rem 122
57 IFJ=1ANDS=1THENBG=BG+1:IFBG<12THEN7
                                          :rem 120
58 IFJ=1ANDS=1ANDBG=12THEN76              :rem 224
59 GOTO37                                 :rem 17
60 JH=JH+1:IFJH<3THEN7                    :rem 83
61 POKE36876,0:POKE36877,220:FORL=15TO0ST
   EP-1:POKE36878,L:FORZN=1TO50           :rem 229
62 POKE36879,ZN:NEXT:NEXT:POKE36877,0:POK
   E36878,0                               :rem 23
63 PRINT"{CLR}":FORZN=8TO255:POKE36879,ZN
   :NEXT                                  :rem 17
64 PRINT"{2 DOWN}{3 RIGHT}{GRN}YOU SURVIV
   ED"BG-1:PRINT"{4 RIGHT}ATTACK WAVES.":
   GOSUB69                                :rem 66
65 PRINT"{2 DOWN}{RVS}{BLU}PLAY AGAIN?(Y,
   N){OFF}"                               :rem 26
66 GETA$:IFA$=""THEN66                    :rem 251
67 IFA$="Y"THENRUN                        :rem 95
68 END                                    :rem 69
69 PRINT"{DOWN}{3 RIGHT}YOUR RANK IS:":PR
   INT"{DOWN}{4 RIGHT}"B$(BG-1)"{GRN}.":R
   ETURN                                  :rem 209
70 DATA-23,-22,-21,-1,0,1,21,22,23
                                          :rem 127
71 DATA"PRIVATE{BLU}","{RED}CORPORAL{BLU}
   ","{CYN}SERGEANT{BLU}"                 :rem 148
72 DATA"{PUR}SECOND LIEUTENANT {BLU}","
   {BLK}FIRST LIEUTENANT"                 :rem 184
73 DATA"{RED}CAPTAIN{BLU}","{BLK}MAJOR
   {BLU}","{PUR}COLONEL{BLU}"             :rem 210
74 DATA"{PUR}GENERAL{BLU}","{BLK}FIELD MA
   RSHALL{BLU}"                           :rem 89
75 DATA"{RED}MR. PRESIDENT"               :rem 97
76 PRINT"{CLR}{RED}YOU HAVE SAVED EARTH
   {2 SPACES}FROM THE ALIEN ATTACK!{BLU}"
   :GOTO65                                :rem 73
```

Program 2: 64 Version

```
100 CO=54272:FORT=COTOCO+24:POKET,0:NEXT
                                          :rem 0
110 POKE53280,0:POKE53281,1:V$=" SECONDS"
                                          :rem 44
120 PRINT"{CLR}{7 DOWN}"TAB(12)"{RVS}
    {RED}REVENGE OF CYON{OFF}"            :rem 215
130 PRINT"{2 DOWN}"TAB(11)"{BLU}(USE JOYS
    TICK #2)"                             :rem 119
140 PRINT"{5 DOWN}{8 RIGHT}WHAT SKILL LEV
    EL (1-6)?"                            :rem 242
150 GETA$:IFA$<"1"ORA$>"6"THEN150         :rem 61
160 D=VAL(A$):SC=1024                     :rem 133
170 POKECO+24,15:POKECO+5,17:POKECO+6,248
    :POKECO,100                           :rem 73
180 BG=1                                  :rem 144
190 POKE53281,1:PRINT"{CLR}":POKE53281,0:
    PRINT"{CYN}"BG:TI$="000000"           :rem 94
200 LK=INT(RND(1)*150)+SC                 :rem 137
210 FORI=1TO10:O=1064+INT(RND(1)*520):POK
    EO,46:POKEO+CO,1:NEXT                 :rem 237
220 POKE1154,81:POKE1154+CO,1             :rem 113
230 PRINT"{13 DOWN}"                      :rem 67
240 PRINT"{GRN}{6 DOWN}N{2 T}{UP}{LEFT}
    {RED}X{GRN}{DOWN}{T}M{DOWN}M{2 @}
    N{UP}N{UP}N{2 T}{UP}N{UP}N{2 T}M
    {DOWN}M{DOWN}M{DOWN}M{DOWN}M{7 @}N
    {UP}N{UP}NM{DOWN}M{2 @}N{UP}N{UP}N
    {T}"                                  :rem 43
250 PRINT"{21 RIGHT}N{UP}{G}{UP}{LEFT}
    {@}N{2 T}{UP}{LEFT}{BLU}-{GRN}
    {DOWN}{T}M{@}{DOWN}{LEFT}{M}
    {DOWN}M"                              :rem 83
260 POKE1769,87:POKE1769+CO,4:POKE1887,12
    4:POKE1887+CO,5                       :rem 249
270 FORI=1890TO1887STEP-1:POKEI+1,32:POKE
    I+CO+1,0:POKEI,124:POKEI+CO,5:rem 248
280 FORJ=1TO80:NEXTJ,I                    :rem 124
290 J=0:K=0                               :rem 68
300 X=20:Y=12:TH=0:S1=0:A=32:A1=32:A2=32
                                          :rem 229
310 FORW=1TO3:FORI=10TO200STEP10:POKECO+1
    ,I:POKECO+4,33:FORJ=1TO10:NEXTJ,I,W
                                          :rem 64
320 PRINT"{HOME}{DOWN}{4 SPACES}":rem 138
330 POKECO+4,32                           :rem 71
340 S=1524                                :rem 243
350 POKES,A                               :rem 135
360 P=PEEK(56320)AND15:Y=Y+((PAND1)=0)-((
    PAND2)=0):X=X+((PAND4)=0)-((PAND8)=0)
                                          :rem 108
370 FR=PEEK(56320)AND16:IFX<0THENX=39
                                          :rem 220
380 IFY<0THENY=0                          :rem 228
390 IFX>39THENX=0                         :rem 33
400 IFY>24THENY=24                        :rem 75
410 IFLK>1983THEN610                      :rem 148
420 C=C+1:IFFR=0THENTH=TH+1               :rem 131
430 POKELK+CO+PO,D1:IFA1<>43THENPOKELK+PO
    ,A1                                   :rem 254
440 IFBG>6ANDJ<>1THENPOKELK+CO+PO+6,D2:IF
    A2<>43THENPOKELK+PO+6,A2              :rem 71
450 IFFR=0ANDTH<=7THENGOSUB1030           :rem 191
460 PO=INT(RND(1)*D)+1:IFTH>7THENFR=1
                                          :rem 241
470 POKEOS,32:NS=1064+INT(RND(1)*520):POK
    ENS,46:OS=NS                          :rem 200
480 S=SC+X+Y*40                           :rem 87
490 A=PEEK(S):POKES,43:IFC=2THENLK=LK+40:
    C=0                                   :rem 193
500 A1=PEEK(LK+PO):D1=PEEK(LK+PO+CO)
                                          :rem 155
510 IFBG>6ANDJ<>1THENA2=PEEK(LK+PO+6):D2=
    PEEK(LK+PO+CO+6)                      :rem 227
520 IFS1<>1THENPOKELK+PO+CO,7:POKELK+PO,9
    0                                     :rem 147
530 A$="SHOTS":IFTH=1THENA$="SHOT "
                                          :rem 53
540 PRINT"{HOME}{BLU}";:IFS=LK+POANDFR=0T
    HENPRINTTH;A$:PRINTRIGHT$(TI$,2)V$:GO
    TO910                                 :rem 193
550 IFBG>6ANDJ<>1THENPOKELK+PO+CO+6,7:POK
    ELK+PO+6,90                           :rem 238
560 IFBG>6ANDJ<>1ANDS=LK+PO+6ANDFR=0THENP
    RINTTH;A$:PRINTRIGHT$(TI$,2);:GOSUB86
    0                                     :rem 80
570 IFPEEK(1769)=87THENPOKE1769,81:POKE17
    69+CO,4:GOTO590                       :rem 92
```

```
580 POKE1769,87:POKE1769+CO,4      :rem 155
590 IFJ=1ANDS1=1THENBG=BG+1:GOTO1040
                                   :rem 188
600 POKESC+QW,32:QW=INT(RND(1)*360)+80:PO
    KESC+QW,46:GOTO350             :rem 73
610 JH=JH+1:IFJH<3THEN190          :rem 231
620 POKECO+1,10:POKECO+4,129       :rem 100
630 FORI=0TO15:FORJ=0TO15:POKE53280,I:POK
    E53281,J:NEXTJ,I               :rem 26
640 POKE53280,0:POKE53281,0        :rem 239
650 POKECO+4,128                   :rem 130
660 PRINT"{CLR}{7 DOWN}{CYN}{6 RIGHT}YOU
    {SPACE}SURVIVED"BG-1"ATTACK WAVES.
    {7 DOWN}"                      :rem 71
670 ONBGGOSUB730,740,750,760,770,780,790,
    800,810,820,830                :rem 172
680 PRINT"{YEL}{5 DOWN}{11 RIGHT}PLAY AGA
    IN (Y/N)?"                     :rem 157
690 GETA$:IFA$=""THEN690           :rem 97
700 IFA$="Y"THENRUN                :rem 137
710 IFA$="N"THENEND                :rem 97
720 GOTO690                        :rem 113
730 PRINT"{PUR}{13 RIGHT}SORRY, PRIVATE."
    :RETURN                        :rem 174
740 PRINT"{PUR}{10 RIGHT}NOT TOO BAD, COR
    PORAL.":RETURN                 :rem 170
750 PRINT"{PUR}{10 RIGHT}NICE GOING, SERG
    EANT.":RETURN                  :rem 139
760 PRINT"{PUR}{6 RIGHT}WAY TO GO, SECOND
     LIEUTENANT.":RETURN           :rem 11
770 PRINT"{PUR}{6 RIGHT}YOU ARE NOW FIRST
     LIEUTENANT.":RETURN           :rem 75
780 PRINT"{PUR}{10 RIGHT}YOU ARE NOW A CA
    PTAIN.":RETURN                 :rem 128
790 PRINT"{PUR}{11 RIGHT}YOU ARE NOW A MA
    JOR.":RETURN                   :rem 23
800 PRINT"{PUR}{9 RIGHT}YOU ARE NOW A COL
    ONEL.":RETURN                  :rem 104
810 PRINT"{PUR}{9 RIGHT}YOU ARE NOW A GEN
    ERAL.":RETURN                  :rem 91
820 PRINT"{PUR}{6 RIGHT}YOU ARE NOW A FIE
    LD MARSHALL.":RETURN           :rem 191
830 PRINT"{PUR}{10 RIGHT}HELLO, MR. PRESI
    DENT.":RETURN                  :rem 141
840 PRINT"{CLR}{YEL}{8 DOWN}{RIGHT}YOU SA
    VED THE EARTH FROM ALIEN ATTACK!!
    {7 DOWN}"                      :rem 22
850 GOTO 680                       :rem 116
860 FORF=1TO3:FORL=150TO100STEP-1:POKECO+
    1,L:POKECO+4,17:NEXT:NEXT:POKECO+4,16
                                   :rem 185
870 PRINTV$:POKELK+PO+CO+6,1:POKELK+PO+6,
    43                             :rem 169
880 FORZ=40TOLK-SCSTEP40:G=PEEK(LK+PO-Z+6
    ):G1=PEEK(LK+PO-Z+CO+6)        :rem 7
890 POKELK+PO-Z+CO+6,7:POKELK+PO-Z+6,90:F
    ORH=1TO70:NEXT:POKELK+PO-Z+CO+6,G1
                                   :rem 237
900 POKELK+PO-Z+6,G:NEXT:TH=0:J=1:RETURN
                                   :rem 76
910 IFBG>6ANDJ<>1THENPOKELK+PO+CO+6,7:POK
    ELK+PO+6,90                    :rem 238
920 S1=1                           :rem 141
930 FORT=1TO3:FORTY=100TO150:POKECO+1,TY:
    POKECO+4,17:NEXT:NEXT:POKECO+4,16
                                   :rem 237
940 S1=1:POKELK+PO+CO,1:TH=0:POKELK+PO,43
                                   :rem 19
950 FORZ=40TOLK-SCSTEP40:G=PEEK(LK+PO-Z):
    G1=PEEK(LK+PO-Z+CO)            :rem 67
960 POKELK+PO-Z+CO,7:POKELK+PO-Z,90:FORH=
    1TO70:NEXT:POKELK+PO-Z+CO,G1   :rem 200
970 POKELK+PO-Z,G:NEXT             :rem 163
980 IFBG<7THENBG=BG+1:GOTO190      :rem 19
990 IFJ=1ANDS=1THENBG=BG+1:IFBG<12THEN190
                                   :rem 17
1000 IFJ=1ANDS=1ANDBG=12THEN840    :rem 99
1010 GOTO550                       :rem 149
1020 REM FIRE NOISE                :rem 75
1030 POKECO+1,200:POKE CO+4,129:POKECO+4,
     128:FORT=1TO100:NEXT:RETURN   :rem 140
1040 IFBG<=11THEN190               :rem 129
1050 GOTO840                       :rem 155
```

Therapy

See article on page 78.

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: Therapy—64 Version

```
100 PRINTCHR$(142)CHR$(8)CHR$(30):POKE532
    81,0:POKE53280,0:GOSUB1230:POKE198,0
                                   :rem 188
105 Q=0:QD=0                       :rem 144
110 PRINTCHR$(147);"HELLO. I'M DR. ROM. W
    HAT'S YOUR NAME?"              :rem 40
115 GOSUB1160:A$=P1$:PRINT         :rem 39
120 PRINT"IN ONE WORD, ";A$;",":PRINT"WHA
    T IS YOUR PROBLEM?":GOSUB1160:B$=P1$
                                   :rem 14
130 PRINT:PRINTB$;"...?":PRINT:PRINT"CAN
    {SPACE}YOU TELL ME MORE?"      :rem 108
140 GOSUB1160:GOSUB900             :rem 48
150 PRINT:PRINT"I UNDERSTAND ";B$;" IS DI
    FFICULT":PRINT"FOR YOU."       :rem 226
160 GOSUB1160:IFP1$="NO"THENPRINT"MAYBE I
    'M NOT QUITE UNDERSTANDING..."
                                   :rem 111
170 PRINT:PRINT"CAN YOU BE MORE SPECIFIC?
     HOW IS":PRINTB$;" A PROBLEM FOR YOU?
    "                              :rem 233
180 GOSUB1160:GOSUB900             :rem 52
190 PRINT:PRINT"HOW DOES THIS MAKE YOU FE
    EL, ";A$;"?":GOSUB1160:C$=P1$  :rem 73
200 PRINTCHR$(147)                 :rem 13
205 PRINT"SO WHAT YOU'RE SAYING, ";A$;","
    :PRINT"IS THAT YOUR PROBLEM WITH ";B$
                                   :rem 80
210 PRINT"IS MAKING YOU FEEL ";C$;".":GOS
    UB1160                         :rem 161
220 PRINT:PRINT"CAN YOU ELABORATE ON YOUR
     FEELINGS?":GOSUB1160:GOSUB900
                                   :rem 215
230 PRINT:PRINT"HAS THIS BEEN A PROBLEM F
    OR YOU BEFORE? (YES OR NO)":GOSUB1160
                                   :rem 133
240 IFP1$<>"NO"THEN260             :rem 236
250 PRINT"I SEE. THEN THIS NEW SITUATION
    {SPACE}MUST BE{2 SPACES}DIFFICULT FOR
     YOU.":GOTO320                 :rem 81
260 PRINT:PRINT"DID YOU ALSO FEEL ";C$;"
    {SPACE}THEN?"                  :rem 216
270 GOSUB1160:PRINT"TELL ME MORE.":rem 16
280 GOSUB1160:GOSUB900             :rem 53
```

```
290 PRINTCHR$(147)"I THINK WE HAVE SOMETH
    ING HERE. DO YOU{2 SPACES}SEE A PATTE
    RN?"                                :rem 236
300 GOSUB1160:PRINT:PRINT"GO ON..."
                                        :rem 106
310 GOSUB1160:PRINT:PRINT"THIS SOUNDS DIF
    FICULT FOR YOU.":GOSUB1160          :rem 240
320 PRINT:PRINT"DO YOU HAVE A PLAN TO DEA
    L WITH THIS{4 SPACES}CURRENT SITUATIO
    N?"                                 :rem 156
330 PRINT"YES OR NO.":GOSUB1160         :rem 70
340 IFP1$<>"YES"THEN350                 :rem 65
343 PRINT"DO YOU THINK THIS PLAN WILL BE"
    :PRINT"SUCCESSFUL?":GOTO360         :rem 241
350 PRINT:PRINT"WHY DON'T YOU MAKE A LIST
     OF POSSIBLE{3 SPACES}SOLUTIONS, THEN
    ."                                  :rem 107
360 GOSUB1160:GOSUB900                  :rem 52
370 FORT=1TO500:NEXTT:PRINTCHR$(147)
                                        :rem 253
380 PRINT"OKAY, WHAT SINGLE WORD BEST DES
    CRIBES"                             :rem 192
385 PRINT"HOW YOU ARE FEELING RIGHT NOW?"
                                        :rem 223
390 GOSUB1160:D$=P1$::PRINT:PRINTD$;"...?
    "                                   :rem 224
400 GOSUB1160:GOSUB900:PRINT            :rem 246
410 PRINT"I'M THINKING OF DOING SOMETHING
     HERE.{3 SPACES}LET'S TRY SOME WORD";
                                        :rem 142
430 PRINT" ASSOCIATION":PRINT"AND SEE WHE
    RE IT LEADS US."                    :rem 183
440 PRINT"WHAT DO YOU THINK(YES OR NO)?":
    GOSUB1160                           :rem 236
450 IFP1$="YES"THEN490                  :rem 11
460 PRINT:PRINT"YOU SEEM TO BE HAVING SOM
    E PROBLEMS WITHTHIS."               :rem 122
470 PRINT"CAN YOU TELL ME ABOUT IT?":GOSU
    B1160:IFP1$="NO"THEN840             :rem 46
480 PRINT:PRINT"I REALLY THINK A WORD ASS
    OCIATION WOULD BE USEFUL RIGHT NOW."
                                        :rem 4
490 PRINT:PRINT"LET'S DO IT."           :rem 242
500 PRINT"I'LL SAY A WORD. YOU SAY THE FI
    RST WORD THAT COMES TO YOUR MIND."
                                        :rem 133
510 REM ***WORD ASSOCIATION***          :rem 239
520 FORT=1TO5000:NEXTT:PRINTCHR$(147);"DO
    G":PRINT:GOSUB1160                  :rem 204
530 PRINT:PRINT"DRINK":PRINT:GOSUB1160
                                        :rem 241
540 PRINT:PRINT"HOME":PRINT:GOSUB1160:E$=
    P1$                                 :rem 40
550 PRINT:PRINTB$:PRINT:GOSUB1160:F$=P1$
                                        :rem 35
560 PRINT:PRINT"FEELINGS":PRINT:GOSUB1160
                                        :rem 201
570 PRINT:PRINT"FUN":PRINT:GOSUB1160:G$=P
    1$                                  :rem 237
580 PRINT:PRINT"MOM":PRINT:GOSUB1160:I$=P
    1$                                  :rem 240
590 PRINT:PRINTC$:PRINT:GOSUB1160:J$=P1$
                                        :rem 44
600 FORT=1TO1000:NEXTT:PRINTCHR$(147)
                                        :rem 37
610 PRINT"I NOTICED WHEN I SAID HOME":PRI
    NT"THAT YOU SAID ";E$;"."           :rem 39
620 PRINT"DOES THIS SOMEHOW REFLECT HOW Y
    OU FEEL{2 SPACES}ABOUT YOURSELF?"
                                        :rem 45
630 PRINT"YES OR NO":GOSUB1160:IFP1$<>"YE
    S"THEN650                           :rem 2
640 PRINT:PRINT"IN WHAT WAY?":GOSUB1160:G
    OSUB900                             :rem 2
650 PRINT:PRINT"HOW DOES THIS RELATE TO Y
    OUR PROBLEM":PRINT"WITH ";B$        :rem 44
660 GOSUB1160:GOSUB900:PRINT:PRINT"WHEN I
     SAID ";B$;" YOU SAID ";F$          :rem 136
670 PRINT"WHAT DO YOU THINK THIS MEANS?":
    GOSUB1160:GOSUB900                  :rem 112
680 PRINT:PRINT"ARE YOU DISTRESSED? DO YO
    U WANT A{7 SPACES}KLEENEX?":GOSUB1160
                                        :rem 28
690 IFP1$<>"YES"THEN710                 :rem 73
700 PRINT"HERE.":FORT=1TO1000:NEXTT
                                        :rem 206
710 PRINT:PRINT"IT'S INTERESTING THAT WHE
    N I SAID FUN,{2 SPACES}YOU SAID ";G$
                                        :rem 57
720 GOSUB1160:GOSUB900:PRINTCHR$(147);"HM
    MMM..."                             :rem 110
730 PRINT:PRINT"IT SEEMS TO ME, ";A$;","
                                        :rem 248
735 PRINT"THAT THIS ALL TIES IN TO YOUR P
    ROBLEM"                             :rem 129
740 PRINT"WITH ";B$                     :rem 73
750 GOSUB1160:GOTO770                   :rem 245
760 REM ***DREAMS***                    :rem 57
770 PRINT:PRINT"LET'S TRY A DIFFERENT":PR
    INT"APPROACH,";A$                   :rem 145
780 PRINT"TELL ME ABOUT ONE OF YOUR DREAM
    S.":GOSUB1160:GOSUB1040:IFQD=1THEN840
                                        :rem 246
790 PRINT:PRINT"HOW WOULD YOU DESCRIBE YO
    UR FEELINGS{4 SPACES}IN THE DREAM?"
                                        :rem 171
791 GOSUB1160                           :rem 233
800 PRINT:PRINT"DID THE DREAM HAVE ANYTHI
    NG TO DO WITH{2 SPACES}";I$         :rem 235
810 GOSUB1160:FORT=1TO1000:NEXTT        :rem 245
820 REM ***ALL DONE***                  :rem 121
830 PRINT:PRINT"I THINK WE'RE MOVING IN A
    {15 SPACES}GOOD DIRECTION.":PRINT
                                        :rem 187
840 PRINT"WE'VE DISCUSSED YOUR PROBLEM WI
    TH":PRINTB$;" AND HOW THIS MAKES YOU
                                        :rem 255
850 PRINT"FEEL ";C$;";"                 :rem 230
860 PRINT"AND DISCUSSED SOME POSSIBLE SOL
    UTIONS."                            :rem 124
870 PRINT:PRINT"I SEE YOUR TIME IS UP.
    {18 SPACES}SEE YOU NEXT WEEK."
                                        :rem 189
880 END                                 :rem 119
890 REM ***KEYWORDS***                  :rem 249
900 IFQ>0THENRETURN                     :rem 246
910 FORJ=1TOLEN(P1$)-5                  :rem 19
920 IFMID$(P1$,J,5)<>" FUN "THEN930
                                        :rem 103
925 PRINT:PRINT"WHAT ARE YOUR FEELINGS AB
    OUT FUN?":GOTO950                   :rem 148
930 NEXTJ                               :rem 37
940 RETURN                              :rem 125
950 GOSUB1160:Q=1:PRINT:PRINT"THESE FEELI
    NGS SEEM IMPORTANT."                :rem 141
960 GOSUB1160:RETURN                    :rem 1
1040 REM ***DREAM KEYWORD SEARCH***
                                        :rem 233
1050 FORJ=1TOLEN(P1$)-7                 :rem 65
1060 IFMID$(P1$,J,7)=" DON'T "THEN1120
                                        :rem 243
1070 NEXTJ                              :rem 81
```

```
1080 FORJ=1TOLEN(P1$)-6                :rem 67
1090 IFMID$(P1$,J,6)=" DONT "THEN1120
                                       :rem 206
1100 NEXTJ                             :rem 75
1110 RETURN                            :rem 163
1120 PRINTCHR$(147)"WHY DO YOU SUPPOSE TH
     AT IS?":GOSUB1160:GOSUB900        :rem 27
1130 PRINT"THIS MAY BE SOMETHING THAT WE'
     LL WANT"                          :rem 176
1140 PRINT"TO DISCUSS LATER. WE MAY FIND
     {SPACE}THAT IT"                   :rem 112
1150 PRINT"RELATES TO YOUR PROBLEM WITH "
     ;B$:QD=1:RETURN                   :rem 223
1160 REM ***COMMODORE PUNCTUATION INPUT**
     *                                 :rem 55
1170 P1$=""                            :rem 239
1180 GETP2$:IFP2$=""THEN1180           :rem 57
1190 PRINTP2$;                         :rem 57
1200 IFP2$=CHR$(13)THENRETURN          :rem 250
1210 P1$=P1$+P2$                       :rem 28
1220 GOTO1180                          :rem 200
1230 REM ***INTRODUCTION***            :rem 72
1240 PRINTCHR$(147);TAB(15)"THERAPY"
                                       :rem 108
1250 PRINT:PRINT"WOULD YOU LIKE AN INTROD
     UCTION (Y/N)"                     :rem 101
1260 GETQ$:IFQ$<>"Y"ANDQ$<>"N"THEN1260
                                       :rem 191
1270 IFQ$="N"THENRETURN                :rem 172
1280 PRINTCHR$(147);"WELCOME TO YOUR THER
     APY SESSION. DR. ROM";            :rem 31
1285 PRINT"WILL BE WITH YOU IN A ";
                                       :rem 172
1290 PRINT"MOMENT. WHILE YOU ARE WAITING,
      HERE ARE SOME HELPFUL"           :rem 104
1300 PRINT"SUGGESTIONS ON HOW TO GET THE
     {SPACE}MOST OUT{2 SPACES}OF YOUR THE
     RAPY SESSION."                    :rem 109
1305 PRINT:PRINT                       :rem 29
1310 PRINT"AS WITH MOST THINGS IN LIFE, W
     ITH{7 SPACES}THERAPY, THE MORE YOU "
     ;                                 :rem 42
1320 PRINT"PUT IN, THE MORE{2 SPACES}YOU
     {SPACE}GET OUT. YOU MAY FIND IT FUN
     {SPACE}TO TRY AND TRIP";          :rem 228
1330 PRINT" UP THE DOCTOR; MAKE FUN OF HI
     S GRAMMAR, OR INSULT HIM MERCILESSLY
     ."                                :rem 175
1340 PRINT"{DOWN}HOWEVER, EVEN THOUGH THI
     S IS A PARLOR{3 SPACES}GAME, YOU MAY
      STILL FIND ";                    :rem 230
1350 PRINT"YOURSELF HAVINGINTERESTING, AN
     D EVEN IMPORTANT,"                :rem 51
1360 PRINT"INSIGHTS. THIS WILL ONLY HAPPE
     N IF YOU{2 SPACES}TRY YOUR BEST TO U
     TILIZE ";                         :rem 172
1370 PRINT"THIS SESSION ASAN ENJOYABLE WA
     Y TO MULL OVER THE"               :rem 159
1380 PRINT"PROBLEMS AND PEEVES OF LIFE."
                                       :rem 127
1390 PRINT:PRINT:PRINTCHR$(18)"HIT ANY KE
     Y TO CONTINUE"                    :rem 165
1400 POKE198,0:WAIT198,1               :rem 96
1410 PRINTCHR$(147):PRINT:PRINT"I SEE THE
      DOCTOR IS IN NOW."               :rem 58
1420 PRINT:PRINT:PRINT"TO TALK TO DR. ROM
     , JUST TYPE IN YOUR"              :rem 228
1430 PRINT"RESPONSE; AND HIT ";CHR$(18);"
     RETURN";CHR$(146);" WHEN YOU ARE"
                                       :rem 254
1440 PRINT"FINISHED.":PRINT:PRINT:PRINT"E
     NJOY YOUR THERAPY SESSION."       :rem 238
1450 PRINTSPC(240);CHR$(18);"HIT ANY KEY
     {SPACE}TO BEGIN"                  :rem 87
1460 POKE198,0:WAIT198,1:RETURN        :rem 128
```

BEFORE TYPING...

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

Program 2: Therapy—VIC Version

```
100 PRINTCHR$(142)CHR$(8)CHR$(30):POKE368
    79,8:GOSUB1230:POKE198,0           :rem 11
105 Q=0:QD=0                           :rem 144
110 PRINTCHR$(147);"HELLO. I'M DR. ROM.
    {3 SPACES}WHAT'S YOUR NAME?"       :rem 40
115 GOSUB1160:A$=P1$:PRINT             :rem 39
120 PRINT"IN ONE WORD, ";A$;",":PRINT"WHA
    T IS YOUR PROBLEM?":GOSUB1160:B$=P1$
                                       :rem 14
130 PRINT:PRINTB$;"...?":PRINT:PRINT"CAN
    {SPACE}YOU TELL ME MORE?"          :rem 108
140 GOSUB1160:GOSUB900                 :rem 48
150 PRINT:PRINT"I UNDERSTAND ";B$:PRINT"I
    S DIFFICULT FOR YOU."              :rem 99
160 GOSUB1160:IFP1$="NO"THENPRINT"MAYBE I
    'M NOT QUITE{3 SPACES}UNDERSTANDING..
    ."                                 :rem 111
170 PRINT:PRINT"CAN YOU BE MORE{7 SPACES}
    SPECIFIC? HOW IS":PRINTB$;" A PROBLEM
    ?"                                 :rem 5
180 GOSUB1160:GOSUB900                 :rem 52
190 PRINT:PRINT"HOW DOES THIS MAKE YOUFEE
    L, ";A$;"?":GOSUB1160:C$=P1$       :rem 73
200 PRINTCHR$(147)                     :rem 13
205 PRINT"SO WHAT YOU'RE SAYING,";A$;", I
    S THAT YOUR":PRINT"PROBLEM WITH ";B$
                                       :rem 80
210 PRINT"IS MAKING YOU FEEL ":PRINTC$;".
    ":GOSUB1160                        :rem 45
220 PRINT:PRINT"CAN YOU ELABORATE ON
    {2 SPACES}YOUR FEELINGS?":GOSUB1160:G
    OSUB900                            :rem 215
230 PRINT:PRINT"HAS THIS BEEN A{7 SPACES}
    PROBLEM FOR YOU BEFORE? (YES OR NO)":
    GOSUB1160                          :rem 133
240 IFP1$<>"NO"THEN260                 :rem 236
250 PRINT"I SEE. THEN THIS NEW{2 SPACES}S
    ITUATION MUST BE{5 SPACES}DIFFICULT F
    OR YOU.":GOTO320                   :rem 81
260 PRINT:PRINT"DID YOU ALSO FEEL":PRINTC
    $;" THEN?"                         :rem 100
270 GOSUB1160:PRINT"TELL ME MORE.":rem 16
280 GOSUB1160:GOSUB900                 :rem 53
290 PRINTCHR$(147)"I THINK WE HAVE
    {7 SPACES}SOMETHING HERE. DO
    {4 SPACES}YOU SEE A PATTERN?":rem 236
300 GOSUB1160:PRINT:PRINT"GO ON..."
                                       :rem 106
310 GOSUB1160:PRINT"THIS SOUNDS DIFFICULT
     FOR YOU.":GOSUB1160               :rem 41
320 PRINT:PRINT"DO YOU HAVE A PLAN TO DEA
    L WITH THIS CURRENTSITUATION?";
                                       :rem 215
330 PRINT" (YES OR NO)":GOSUB1160:rem 105
340 IFP1$<>"YES"THEN350                :rem 65
```

```
343 PRINT"DO YOU THINK THIS PLANWILL BE S
    UCCESSFUL?":GOTO360          :rem 230
350 PRINT:PRINT"WHY DON'T YOU MAKE A
    {2 SPACES}LIST OF POSSIBLE{6 SPACES}S
    OLUTIONS, THEN."             :rem 107
360 GOSUB1160:GOSUB900           :rem 52
370 FORT=1TO500:NEXTT:PRINTCHR$(147)
                                 :rem 253
380 PRINT"OKAY, WHAT SINGLE WORDBEST DESC
    RIBES";                      :rem 251
385 PRINT" HOW YOUARE FEELING RIGHT NOW?"
                                 :rem 223
390 GOSUB1160:D$=P1$::PRINT:PRINTD$;"...?
    "                            :rem 224
400 GOSUB1160:GOSUB900:PRINT     :rem 246
410 PRINT"I'M THINKING OF DOING SOMETHING
     HERE.":PRINT"LET'S TRY SOME WORD"
                                 :rem 94
430 PRINT"ASSOCIATION AND SEE{3 SPACES}WH
    ERE IT LEADS US."            :rem 172
440 PRINT"WHAT DO YOU THINK{5 SPACES}(YES
     OR NO)?":GOSUB1160          :rem 236
450 IFP1$="YES"THEN490           :rem 11
460 PRINT:PRINT"YOU SEEM TO BE HAVING SOM
    E PROBLEMS WITH{4 SPACES}THIS."
                                 :rem 122
470 PRINT"CAN YOU TELL ME ABOUT IT?":GOSU
    B1160:IFP1$="NO"THEN840      :rem 46
480 PRINT:PRINT"I REALLY THINK A WORD ASS
    OCIATION WOULD BE{2 SPACES}USEFUL RIG
    HT NOW."                     :rem 4
490 PRINT:PRINT"LET'S DO IT."    :rem 242
500 PRINT"I'LL SAY A WORD. YOU{2 SPACES}S
    AY THE FIRST WORD{4 SPACES}THAT COMES
     TO YOUR{4 SPACES}MIND."     :rem 133
510 REM ***WORD ASSOCIATION***   :rem 239
520 FORT=1TO5000:NEXTT:PRINTCHR$(147);"DO
    G":PRINT:GOSUB1160           :rem 204
530 PRINT:PRINT"DRINK":PRINT:GOSUB1160
                                 :rem 241
540 PRINT:PRINT"HOME":PRINT:GOSUB1160:E$=
    P1$                          :rem 40
550 PRINT:PRINTB$:PRINT:GOSUB1160:F$=P1$
                                 :rem 35
560 PRINT:PRINT"FEELINGS":PRINT:GOSUB1160
                                 :rem 201
570 PRINT:PRINT"FUN":PRINT:GOSUB1160:G$=P
    1$                           :rem 237
580 PRINT:PRINT"MOM":PRINT:GOSUB1160:I$=P
    1$                           :rem 240
590 PRINT:PRINTC$:PRINT:GOSUB1160:J$=P1$
                                 :rem 44
600 FORT=1TO1000:NEXTT:PRINTCHR$(147)
                                 :rem 37
610 PRINT"I NOTICED WHEN I SAID HOME THAT
     YOU SAID":PRINTE$;"."       :rem 168
620 PRINT"DOES THIS SOMEHOW{5 SPACES}REFL
    ECT HOW YOU FEEL{2 SPACES}ABOUT YOURS
    ELF?"                        :rem 45
630 PRINT"YES OR NO":GOSUB1160:IFP1$<>"YE
    S"THEN650                    :rem 2
640 PRINT:PRINT"IN WHAT WAY?":GOSUB1160:G
    OSUB900                      :rem 2
650 PRINT:PRINT"HOW DOES THIS RELATE
    {2 SPACES}TO YOUR PROBLEM WITH":PRINT
    B$                           :rem 173
660 GOSUB1160:GOSUB900:PRINT:PRINT"WHEN I
     SAID ";B$:PRINT"YOU SAID ";F$:rem 20
670 PRINT"WHAT DO YOU THINK THISMEANS?":G
    OSUB1160:GOSUB900            :rem 112
680 PRINT:PRINT"ARE YOU DISTRESSED? DOYOU
    WANT A KLEENEX?":GOSUB1160   :rem 28
690 IFP1$<>"YES"THEN710          :rem 73
700 PRINT"HERE.":FORT=1TO1000:NEXTT
                                 :rem 206
710 PRINT:PRINT"IT'S INTERESTING THAT WHE
    N I SAID FUN,{2 SPACES}YOU SAID ";G$
                                 :rem 57
720 GOSUB1160:GOSUB900:PRINTCHR$(147);"HM
    MMM..."                      :rem 110
730 PRINT:PRINT"IT SEEMS TO ME, ";A$;","
                                 :rem 248
735 PRINT"THAT THIS ALL TIES IN TO YOUR P
    ROBLEM"                      :rem 129
740 PRINT"WITH ";B$              :rem 73
750 GOSUB1160:GOTO770            :rem 245
760 REM ***DREAMS***             :rem 57
770 PRINT:PRINT"LET'S TRY A DIFFERENT":PR
    INT"APPROACH,";A$            :rem 145
780 PRINT"TELL ME ABOUT ONE OF{2 SPACES}Y
    OUR DREAMS.":GOSUB1160:GOSUB1040:IFQD
    =1THEN840                    :rem 246
790 PRINT:PRINT"HOW WOULD YOU DESCRIBEYOU
    R FEELINGS IN THE{2 SPACES}DREAM?":GO
    SUB1160                      :rem 45
800 PRINT:PRINT"DID THE DREAM HAVE
    {4 SPACES}ANYTHING TO DO WITH":PRINTI
    $                            :rem 119
810 GOSUB1160:FORT=1TO1000:NEXTT :rem 245
820 REM ***ALL DONE***           :rem 121
830 PRINT:PRINT"{CLR}I THINK WE'RE MOVING
    {2 SPACES}IN A GOOD DIRECTION.":PRINT
                                 :rem 78

840 PRINT"WE'VE DISCUSSED YOUR{2 SPACES}P
    ROBLEM WITH":PRINTB$:PRINT"AND HOW TH
    IS MAKES YOU"                :rem 173
850 PRINT"FEEL ";C$;";"          :rem 230
860 PRINT"AND DISCUSSED SOME{4 SPACES}POS
    SIBLE SOLUTIONS."            :rem 124
870 PRINT:PRINT"I SEE YOUR TIME IS UP.SEE
     YOU NEXT WEEK."             :rem 189
880 END                          :rem 119
890 REM ***KEYWORDS***           :rem 249
900 IFQ>0THENRETURN              :rem 246
910 FORJ=1TOLEN(P1$)-5           :rem 19
920 IFMID$(P1$,J,5)<>" FUN "THEN930
                                 :rem 103

925 PRINT:PRINT"WHAT ARE YOUR{9 SPACES}FE
    ELINGS ABOUT FUN?":GOTO950   :rem 148
930 NEXTJ                        :rem 37
940 RETURN                       :rem 125
950 GOSUB1160:Q=1:PRINT:PRINT"THESE FEELI
    NGS SEEM{3 SPACES}IMPORTANT.":rem 141
960 GOSUB1160:RETURN             :rem 1
1040 REM ***DREAM KEYWORD SEARCH***
                                 :rem 233
1050 FORJ=1TOLEN(P1$)-7          :rem 65
1060 IFMID$(P1$,J,7)=" DON'T "THEN1120
                                 :rem 243
1070 NEXTJ                       :rem 81
1080 FORJ=1TOLEN(P1$)-6          :rem 67
1090 IFMID$(P1$,J,6)=" DONT "THEN1120
                                 :rem 206
1100 NEXTJ                       :rem 75
1110 RETURN                      :rem 163
1120 PRINTCHR$(147)"WHY DO YOU SUPPOSE TH
     AT IS?":GOSUB1160:GOSUB900  :rem 27
1130 PRINT"THIS MAY BE SOMETHING THAT WE'
     LL WANT"                    :rem 176
1140 PRINT"TO DISCUSS LATER. WE MAY FIND
     {SPACE}THAT IT"             :rem 112
```

```
1150 PRINT"RELATES TO YOUR PROBLEM WITH "
     ;B$:QD=1:RETURN              :rem 223
1160 REM ***COMMODORE PUNCTUATION INPUT**
     *                            :rem 55
1170 P1$=""                       :rem 239
1180 GETP2$:IFP2$=""THEN1180      :rem 57
1190 PRINTP2$;                    :rem 57
1200 IFP2$=CHR$(13)THENRETURN     :rem 250
1210 P1$=P1$+P2$                  :rem 28
1220 GOTO1180                     :rem 200
1230 REM ***INTRODUCTION***       :rem 72
1240 PRINTCHR$(147);TAB(6)"THERAPY"
                                  :rem 60
1250 PRINT:PRINT"WOULD YOU LIKE AN
     {5 SPACES}INTRODUCTION (Y/N)"
                                  :rem 101
1260 GETQ$:IFQ$<>"Y"ANDQ$<>"N"THEN1260
                                  :rem 191
1270 IFQ$="N"THENRETURN           :rem 172
1280 PRINTCHR$(147);"WELCOME TO YOUR
     {7 SPACES}THERAPY SESSION.{6 SPACES}
     DR. ROM";                    :rem 31
1285 PRINT" WILL BE WITH{2 SPACES}YOU IN
     {SPACE}A ";                  :rem 172
1290 PRINT"MOMENT. WHILEYOU ARE WAITING,
     {SPACE}HERE ARE SOME HELPFUL"
                                  :rem 104
1300 PRINT"SUGGESTIONS ON HOW TO GET THE
     {SPACE}MOST OUT OF{3 SPACES}YOUR THE
     RAPY SESSION."               :rem 109
1305 PRINT:PRINT                  :rem 29
1310 PRINT"AS WITH MOST THINGS INLIFE, WI
     TH THERAPY,{3 SPACES}THE MORE YOU ";
                                  :rem 42
1320 PRINT"PUT IN,{2 SPACES}THE MORE
     {2 SPACES}YOU GET OUT.YOU MAY FIND I
     T FUN TOTRY AND TRIP ";      :rem 228
1330 PRINT"UP THE{3 SPACES}DOCTOR; MAKE F
     UN OF{3 SPACES}HIS GRAMMAR, OR INSUL
     THIM MERCILESSLY."           :rem 175
1332 PRINT:PRINTCHR$(18)"HIT ANY KEY"
                                  :rem 210
1335 POKE198,0:WAIT198,1          :rem 103
1340 PRINT"{CLR}{DOWN}HOWEVER, EVEN THOUG
     H{2 SPACES}THIS IS A PARLOR
     {6 SPACES}GAME, YOU MAY STILL
     {3 SPACES}FIND ";            :rem 121
1350 PRINT"YOURSELF HAVING{2 SPACES}INTER
     ESTING, AND EVEN IMPORTANT,";
                                  :rem 110
1360 PRINT" INSIGHTS.{2 SPACES}THIS WILL
     {SPACE}ONLY HAPPEN IF YOU{2 SPACES}T
     RY YOUR BEST TO UTILIZE ";   :rem 172
1370 PRINT"THIS{7 SPACES}SESSION AS AN
     {9 SPACES}ENJOYABLE WAY TO MULL OVER
      THE ";                      :rem 218
1380 PRINT"PROBLEMS AND PEEVES OF LIFE."
                                  :rem 127
1390 PRINT:PRINTCHR$(18)"HIT ANY KEY"
                                  :rem 214
1400 POKE198,0:WAIT198,1          :rem 96
1410 PRINTCHR$(147):PRINT"I SEE THE DOCTO
     R IS IN NOW."                :rem 115
1420 PRINT:PRINT"TO TALK TO DR. ROM,
     {3 SPACES}JUST TYPE IN YOUR" :rem 29
1430 PRINT"RESPONSE; AND HIT ";CHR$(18):P
     RINT"RETURN";CHR$(146);" WHEN YOU AR
     E"                           :rem 138
1440 PRINT"FINISHED.":PRINT:PRINT:PRINT"E
     NJOY YOUR THERAPY SESSION." :rem 238
1450 PRINT:PRINTCHR$(18);" HIT ANY KEY TO
     BEGIN"                       :rem 22
1460 POKE198,0:WAIT198,1:RETURN   :rem 128
```

Spelling Critter

See article on page 82.

BEFORE TYPING...

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

Program 1:

Spelling Critter—VIC Version

```
2 PRINT"{CLR}{4 DOWN}{2 RIGHT}SPELLING CR
  ITTER"                          :rem 143
15 PRINT"{10 DOWN}{2 RIGHT}ONE MOMENT PLE
   ASE"                           :rem 135
20 POKE36869,255                  :rem 105
21 POKE52,28:POKE56,28:CLR        :rem 20
22 FORI=7168TO7679:POKEI,PEEK(I+25600):NE
   XT                             :rem 101
23 FORC=7168TO7175:READA:POKEC,A:NEXT
                                  :rem 66
24 FORC=7664TO7671:READA:POKEC,A:NEXT
                                  :rem 69
25 FORC=7656TO7663:READA:POKEC,A:NEXT
                                  :rem 72
30 DATA96,128,77,63,31,13,0,0     :rem 186
31 DATA3,52,184,240,240,184,52,3  :rem 76
32 DATA0,54,58,242,242,58,54,0    :rem 239
50 DIMW$(50)                      :rem 112
60 PRINT"{CLR}"                   :rem 202
61 POKE36878,15                   :rem 56
70 PRINT"HI,WHAT IS YOUR"         :rem 20
75 INPUT"NAME";N$                 :rem 14
80 GOSUB5000                      :rem 173
90 W=0:R=0:Z=0                    :rem 39
100 FORB=1TON                     :rem 25
120 PRINT"{CLR}{2 DOWN}{5 RIGHT}"W$(B)
                                  :rem 184
140 FORT=1TO1000:NEXT             :rem 27
160 PRINT"{CLR}":A$=""            :rem 27
165 PRINT"{RED}{RVS}{2 SPACES}TO REVIEW T
    HE WORD{2 SPACES}"            :rem 45
166 PRINT"{RVS}{2 SPACES}TYPE ? AND RETUR
    N{3 SPACES}{BLU}"             :rem 211
180 PRINT"{3 DOWN}HOW DO YOU SPELL":PRINT
                                  :rem 98
190 INPUT"THAT WORD";A$           :rem 123
195 IFA$=""ORLEN(A$)=0THEN160     :rem 125
200 IFA$="THAT WORD"THENGOSUB6000:GOTO160
                                  :rem 243
210 IFA$="?"ORA$="/"THEN120       :rem 191
220 IFA$=W$(B)THENGOSUB7000:GOTO400
                                  :rem 80
230 GOSUB8000                     :rem 221
240 PRINT"{CLR}{5 DOWN}I'M SORRY,"N$:PRIN
    T                             :rem 16
260 PRINT"THAT IS WRONG":PRINT    :rem 138
270 POKE36874,223                 :rem 151
274 FORT=1TO500:NEXT              :rem 247
275 POKE36874,0                   :rem 53
280 PRINT"THE CORRECT WAY IS:"    :rem 37
```

```
300 PRINT"{3 DOWN}{5 RIGHT}{BLK}"W$(B)"
    {BLU}"                          :rem 41
320 W=W+1                           :rem 220
340 GOTO450                         :rem 105
400 PRINT"{2 DOWN}THAT IS RIGHT"    :rem 210
420 PRINT"{RED}{DOWN}VERY GOOD{BLU} ";N$
                                    :rem 207
424 M=220                           :rem 184
425 FORP=1TO8                       :rem 27
427 M=M+3                           :rem 210
429 POKE36876,M                     :rem 85
430 FORT=1TO25                      :rem 74
431 POKE36876,0                     :rem 49
432 NEXTP                           :rem 40
433 RESTORE                         :rem 190
440 R=R+1                           :rem 213
450 POKE198,0:PRINT"{3 DOWN}HIT ANY KEY"
                                    :rem 186
460 GETZ$:IFZ$=""THEN460            :rem 137
480 NEXTB                           :rem 29
500 PRINT"{CLR}YOU HAD{RED} "R" {BLU}RIGH
    T AND{RED} ":PRINT              :rem 200
510 PRINTW" {BLU}WRONG."            :rem 152
550 PRINT"{3 DOWN}WOULD YOU LIKE TO":PRIN
    T                               :rem 181
560 PRINT"TRY AGAIN (Y/N)?"         :rem 49
580 GETZ$:IFZ$=""OR(Z$<>"Y"ANDZ$<>"N")THE
    N580                            :rem 115
590 IFZ$="Y"THEN80                  :rem 28
600 PRINT"{CLR}IT HAS BEEN VERY NICE":PRI
    NT:PRINT"SPELLING WITH YOU":PRINT
                                    :rem 34
620 PRINTN$"."                      :rem 9
640 PRINT"{3 DOWN}PLEASE COME BACK AGAIN"
    :PRINT                          :rem 180
650 PRINT"TO PLAY REAL SOON."       :rem 214
660 PRINT"{2 DOWN}{9 RIGHT}{RED}BYE!"
                                    :rem 177
700 END                             :rem 110
5000 REM INPUT                      :rem 57
5050 PRINT"{CLR}{DOWN}HOW MANY WORDS DO W
    E"                              :rem 32
5060 INPUT"HAVE TODAY";Z$:N=VAL(Z$):IFN>5
    0ORN=0THEN5050                  :rem 131
5100 FORX=1TON                      :rem 100
5150 PRINT"{CLR}{RED}{RVS}{SHIFT-SPACE}CH
    ECK YOUR SPELLING!!{OFF}{BLU}"
                                    :rem 251
5200 PRINT"{3 DOWN}WORD #";X;:INPUT"
    {LEFT}";W$(X)                   :rem 2
5210 IFLEN(W$(X))=0ORLEN(W$(X))>10THEN515
    0                               :rem 166
5220 IFASC(W$(X))<65ORASC(W$(X))>90THEN51
    50                              :rem 217
5250 NEXTX                          :rem 99
5300 RETURN                         :rem 168
6000 REM WISEGUY                    :rem 215
6010 PRINT"{RED}{2 DOWN}VERY FUNNY,WISEGU
    Y{BLU}":PRINT                   :rem 235
6015 FORT=1TO1000:NEXT              :rem 82
6020 FORT=1TO500:NEXT               :rem 34
6030 RETURN                         :rem 169
7000 Q=0                            :rem 133
7001 L=7900:CO=38620                :rem 45
7010 POKEL+Q,0:POKECO+Q,4           :rem 242
7020 POKEL+Q+1,61:POKECO+Q+1,4      :rem 226
7030 FORT=1TO50:NEXT                :rem 244
7040 POKEL+Q+1,62:POKECO+Q+1,4      :rem 229
7042 POKE36877,160:FORT=1TO10:NEXT:rem 46
7045 POKE36877,0                    :rem 106
7050 FORT=1TO50:NEXT                :rem 246
7060 IFQ=9THENRETURN                :rem 50
7070 POKEL+Q,32                     :rem 86
7075 Q=Q+1                          :rem 14
7080 GOTO7001                       :rem 208
8000 Q=0                            :rem 134
8001 L=7900:CO=38620                :rem 46
8010 POKEL+Q,0:POKECO+Q,4           :rem 243
8020 POKEL+Q+1,61:POKEL+Q+1,4       :rem 157
8030 FORT=1TO50:NEXT                :rem 245
8040 POKEL+Q+1,62:POKECO+Q+1,4      :rem 230
8042 POKE36877,160:FORT=1TO10:NEXT:rem 47
8045 POKE36877,0                    :rem 107
8050 FORT=1TO50:NEXT                :rem 247
8060 POKEL+Q,32                     :rem 86
8070 IFQ=20THENPOKEL+Q+1,32:RETURN
                                    :rem 123
8075 Q=Q+1                          :rem 15
8080 GOTO8001                       :rem 210
9000 END                            :rem 160
```

Program 2:
Spelling Critter—64 Version

```
10 POKE52,48:POKE56,48:CLR          :rem 22
20 POKE53281,1:PRINT"{CLR}":POKE53281,6:P
   OKE53280,6                       :rem 42
30 S=54272:V=54296:AD=54277:SR=5427F=5
   4273:LF=S:SD=54276               :rem 117
40 FORL=STOS+24:POKEL,0:NEXT:POKEAD,14:PO
   KESR,204                         :rem 135
50 FORI=1436TO1443:POKEI,INT(RND(1)*25)+1
   :NEXT:FORI=1445TO1451            :rem 216
60 POKEI,INT(RND(1)*25)+1:NEXT:FORI=1 TO
   {SPACE}500:NEXT                  :rem 155
70 FORI=1436TO1443:READL:POKEI,L+128:GOSU
   B110:FORJ=1TO70:NEXTJ:POKESD,32:NEXTI
                                    :rem 232
80 FORI=1445TO1451:READL:POKEI,L+128:GOSU
   B110:FORJ=1TO70:NEXTJ:POKESD,32:NEXTI
                                    :rem 232
90 DATA19,16,5,12,12,9,14,7,3,18,9,20,20,
   5,18                             :rem 195
100 POKE1444,160:GOTO 120           :rem 86
110 POKEV,15:POKEHF,40:POKELF,50:POKESD,3
    3:POKEV,0:RETURN                :rem 131
120 PRINT"{HOME}{WHT}{15 DOWN}{3 RIGHT}ON
    E MOMENT.....PLEASE!"
                                    :rem 63
130 PRINTCHR$(142):POKE56334,PEEK(56334)A
    ND254                           :rem 141
140 POKE1,PEEK(1)AND251:FORI=0TO511:POKEI
    +12288,PEEK(I+53248):NEXT       :rem 187
150 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334
    )OR1                            :rem 133
160 FORI=1TO3:READW:FORC=WTOW+7:READA:POK
    EC,A:NEXTC:NEXTI                :rem 58
170 DATA12288,96,128,77,63,31,13,0,0
                                    :rem 32
180 DATA12784,3,52,184,240,240,184,52,3
                                    :rem 179
190 DATA12776,0,54,58,242,242,58,54,0
                                    :rem 87
200 DIMW$(50):CL=54272              :rem 167
210 FORJ=1TO7:READM$(J):NEXTJ       :rem 49
220 POKE53272,(PEEK(53272)AND240)OR12
                                    :rem 41
230 POKE53281,1:PRINT"{CLR}":POKE53281,6:
    POKE53280,6                     :rem 93
240 R=1464:J=1:GOSUB450:GOSUB850    :rem 144
250 N$=NM$:POKE53272,21             :rem 0
```

```
260 GOSUB710:POKE53281,1:PRINT"{CLR}":POK
    E53281,6:POKE53280,6          :rem 178
270 WG=0:RT=0                     :rem 241
280 FORB=1TON                     :rem 34
290 PRINT"{CLR}{WHT}":POKE214,12:PRINT:PO
    KE211,INT((40-LEN(W$(B)))/2)-1:PRINTW
    $(B)                          :rem 115
300 FORT=1TO1000:NEXT             :rem 25
310 PRINT"{CLR}{WHT}{5 SPACES}TO REVIEW T
    HE WORD PRESS ? KEY"          :rem 68
320 POKE53272,28:R=1464:J=2:GOSUB450:GOSU
    B850                          :rem 146
330 A$=NM$                        :rem 247
340 IFA$="THAT WORD"THENJ=6:R=1704:GOSUB1
    020:GOSUB450:R=1544:GOSUB790:GOTO290
                                  :rem 201
350 IFASC(A$)=63ORASC(A$)=47THEN290
                                  :rem 250
360 IFA$=W$(B)THENGOSUB1020:J=5:M$(J)=M$(
    J)+" "+N$:R=1704:GOSUB450     :rem 195
370 IFJ=5THENRT=RT+1:GOSUB400:GOSUB570:GO
    TO560                         :rem 171
380 WG=WG+1:GOSUB1020:J=3:M$(J)=M$(J)+" "
    +N$:R=1704:GOSUB450:GOSUB570  :rem 186
390 J=4:R=1824:M$(J)=M$(J)+" "+W$(B):GOSU
    B450:R=1544:GOSUB790:GOSUB570:rem 192
395 M$(J)="THE CORRECT ANSWER IS:":GOTO59
    0                             :rem 216
400 REM CORRECT ANSWER SOUND      :rem 227
410 POKEV,15:POKESD,17:FORS=40TO100
                                  :rem 137
420 POKEHF,INT(RND(1)*100+40):POKELF,S:FO
    RD=1TO20:NEXT                 :rem 54
430 NEXT:POKESD,16:RETURN         :rem 131
440 REM MESSAGE ROUTINE           :rem 167
450 C=((R-1024)/40)-1:CN=INT((40-LEN(M$(J
    )))/2)-1                      :rem 19
460 FORI=1TOLEN(M$(J)):LR=R+CN+I:IFLR>202
    3THENLR=2023                  :rem 235
470 FORK=1TO20:NEXTK:POKELR+1,32  :rem 248
480 POKELR,0:POKELR+1,62:POKELR+CL,1:POKE
    LR+CL+1,1:GOSUB110:POKESD,32  :rem 49
490 FORK=1TO20:NEXTK:POKELR,32    :rem 158
500 A=ASC(MID$(M$(J),I,1)):IFA=32THENPOKE
    LR,A:GOTO530                  :rem 136
510 IFA<65ORA>90THENPOKE214,C:PRINT:POKE2
    11,CN+I:PRINTCHR$(A):GOTO530  :rem 84
520 CH=A-64:POKELR,CH:POKELR+CL,1 :rem 19
530 NEXTI                         :rem 32
540 FORD=LR+1TOR+38:POKED,0:POKED+1,62:PO
    KED+CL,3:POKED+CL+1,3:GOSUB110:rem 96
550 POKESD,32:FORK=1TO10:NEXTK:POKED,32:P
    OKED+1,32:NEXTD:RETURN        :rem 66
560 POKE198,0:PRINT"{2 DOWN}{15 RIGHT}HIT
     ANY KEY":GOTO 580            :rem 110
570 M$(J)=MID$(M$(J),1,LEN(M$(J))-LEN(N$)
    -1):RETURN                    :rem 76
580 GETZ$:IFZ$=""THEN580          :rem 143
590 NEXTB                         :rem 31
600 REM GAME OVER ROUTINE         :rem 246
610 POKE53272,21:PRINT"{CLR}{WHT}{3 DOWN}
    {11 RIGHT}THE GAME IS OVER"   :rem 64
620 PRINT"{3 DOWN}{2 RIGHT}RIGHT";TAB(33)
    ;"WRONG"                      :rem 41
630 PRINT"{DOWN}{2 RIGHT}";RT;TAB(33);WG
                                  :rem 56
640 PRINT"{8 DOWN}{2 SPACES}WOULD YOU LIK
    E TO PLAY AGAIN (Y/N)?"       :rem 63
650 GETZ$:IFZ$=""OR(Z$<>"Y"ANDZ$<>"N")THE
    N650                          :rem 111
660 IFZ$="Y"THENFORI=1TON:W$(I)="":NEXT:G
    OTO260                        :rem 149
670 PRINT"{CLR}{8 DOWN}{3 SPACES}IT HAS B
    EEN VERY NICE SPELLING{10 SPACES}WITH
     YOU ";N$;"."                 :rem 114
680 PRINT"{2 DOWN}{4 SPACES}PLEASE COME B
    ACK TO PLAY AGAIN."           :rem 231
690 POKE53272,28:R=1704:J=7:GOSUB450:POKE
    53272,21:END                  :rem 83
700 REM INPUT                     :rem 11
710 POKE53280,3:POKE53281,3       :rem 243
720 PRINT"{CLR}{BLU}{12 DOWN}HOW MANY WOR
    DS DO WE HAVE TODAY";:INPUTZ$:N=VAL(Z
    $)                            :rem 104
725 IFN>50ORN=0THEN720            :rem 66
730 PRINT"{CLR}{BLU}{3 DOWN}{9 SPACES}
    {RVS}{SHIFT-SPACE}CHECK YOUR SPELLING
    !!{OFF}{BLU}"                 :rem 0
740 FORX=1TON                     :rem 57
750 POKE214,12:PRINT:POKE211,5:PRINT"ENTE
    R WORD # ";X;:INPUT"{LEFT}";W$(X)
                                  :rem 214
760 FORI=1TO17:POKE214,12:PRINT:POKE211,2
    1+I:PRINTCHR$(32):NEXTI       :rem 220
765 IFLEN(W$(X))=0ORLEN(W$(X))>10THEN750
                                  :rem 81
766 IFASC(W$(X))<65ORASC(W$(X))>90THEN750
                                  :rem 132
770 NEXTX:RETURN                  :rem 79
780 REM ERASE ROUTINE             :rem 25
790 FORI=RTOR+38                  :rem 234
800 CH=62:IF(I+1)/2=INT((I+1)/2)THENCH=61
                                  :rem 22
810 POKEI+CL,1:POKEI+CL+1,1       :rem 77
820 POKEI,0:POKEI+1,CH:FORD=1TO50:NEXTD
                                  :rem 203
830 POKEI,32:GOSUB110:POKESD,32:NEXTI:POK
    EI,32:RETURN                  :rem 160
840 REM ANSWER ROUTINE            :rem 118
850 POKE198,0:POKE1561+CL,1:POKE1562+CL,1
    :POKE1561,0:POKE1562,62:CT=1:NM$=""
                                  :rem 123
860 GETZ$:IFZ$=""THEN860          :rem 145
870 IFASC(Z$)=63ORASC(Z$)=47THENNM$=Z$:RE
    TURN                          :rem 44
880 IFASC(Z$)=32THENA=32:GOTO920  :rem 151
890 IFASC(Z$)=13ANDLEN(NM$)<>0THEN950
                                  :rem 176
900 IFASC(Z$)<65ORASC(Z$)>90THEN860
                                  :rem 48
910 A=ASC(Z$)-64                  :rem 85
920 NM$=NM$+Z$:POKE1560+CT,A:POKE1561+CT,
    0:POKE1561+CT+1,62            :rem 159
930 POKE1560+CT+CL,1:POKE1561+CT+CL,1:POK
    E1561+CT+CL+1,1:CT=CT+1       :rem 235
940 GOSUB110:POKESD,32:GOTO860    :rem 81
950 IFJ<>1THENRETURN              :rem 49
960 FORI=1561TO1582:POKEI,32:FORJ=1TOLEN(
    NM$)                          :rem 250
970 A=I+J:IFA>1582THENA=1582:POKEA,32:GOT
    O 990                         :rem 193
975 AS=ASC(MID$(NM$,J,1)):IFAS<65ORAS>90T
    HENPOKEA,32:GOTO990           :rem 164
980 POKEA+CL,1:POKEA,AS-64        :rem 41
990 NEXTJ:A=I+J+1:IFA>1582THENA=1582:POKE
    A,32:GOTO 1010                :rem 2
1000 POKEA-1+CL,1:POKEA+CL,1:POKEA-1,0:PO
     KEA,62                       :rem 9
1010 FORK=1TO30:NEXTK:GOSUB110:POKESD,32:
     NEXTI:RETURN                 :rem 229
1020 FORI=1561+LEN(NM$)TO1581     :rem 137
1030 POKEI+CL,1:POKEI+1+CL,1:POKEI+2+CL,1
```

```
    :POKEI,32:POKEI+1,0:POKEI+2,62
                                    :rem 238
1040 GOSUB1110:POKESD,32:NEXTI:POKEI,32:PO
    KEI+1,32:RETURN                 :rem 38
1050 DATA"HI, WHAT IS YOUR NAME"    :rem 33
1060 DATA"HOW DO YOU SPELL THAT WORD"
                                    :rem 144
1070 DATA"I'M SORRY, THAT IS WRONG":rem 8
1080 DATA"THE CORRECT ANSWER IS:":rem 192
1090 DATA"THAT IS RIGHT, VERY GOOD"
                                    :rem 14
1100 DATA"VERY FUNNY, WISEGUY"      :rem 79
1110 DATA"BYE!{2 SPACES}BYE!{2 SPACES}BYE
    !"                              :rem 36
```

Word Scramble

See article on page 86.

BEFORE TYPING...

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

Program 1:

Word Scramble—VIC Version

```
10 PRINT"{CLR}":POKE36879,8:PRINT"{RED}
   {7 DOWN}{5 RIGHT}WORD SCRAMBLE":POKE36
   878,15                           :rem 98
20 GOSUB1000:POKE36879,27:PRINT"{CLR}"
                                    :rem 80
25 PRINT"{RED}EACH PLAYER TAKING":PRINT"T
   URNS ENTERS A COMMON"            :rem 114
30 PRINT"{RED}WORD (MAX.10 LETTERS).";
                                    :rem 222
35 PRINT"{RED}THE COMPUTER WILL THEN";:PR
   INT"SCRAMBLE THE WORD AND"       :rem 139
40 PRINT"{RED}PRINT IT."            :rem 169
45 PRINT"{RED}YOU HAVE THREE MINUTES";:PR
   INT"TO FIND IT."                 :rem 233
50 PRINT"{RED}IF FOUND WITHIN THE":PRINT"
   ALLOTTED TIME,YOU WILL";         :rem 64
55 PRINT"{RED}BE GIVEN 50 POINTS.":PRINT"
   EVERY WRONG GUESS THAT";         :rem 221
60 PRINT"{RED}YOU MAKE WILL COST":PRINT"Y
   OU 10 POINTS.{BLU}"              :rem 114
65 PRINT:PRINT:PRINT:PRINT"{3 RIGHT}{RVS}
   {PUR}PRESS SPACE BAR{OFF}"       :rem 246
70 GETA$:IFA$=""THEN70:C=0          :rem 219
80 PRINT"{CLR}{4 DOWN}{GRN}PLAYER # 1'S N
   AME{BLU}":INPUTP$(0)             :rem 200
85 PRINT:PRINT:PRINT"{RED}PLAYER # 2'S NA
   ME{BLU}":INPUTP$(1)              :rem 132
90 PRINT:PRINT"{DOWN}{PUR}"P$(C)",":PRINT
   "{RVS}{RED}ENTER WORD TO BE":PRINT"
   {RVS}SCRAMBLED:{OFF}{BLU}"       :rem 216
92 W$="":INPUTW$:IFW$=""THENPRINT"{UP}";:
   GOTO92                           :rem 27
95 IFLEN(W$)>10THENPRINT"{RVS}{GRN}MORE T
   HAN 10 LETTERS!{OFF}{BLU}{7 UP}":GOTO9
   0                                :rem 254
100 GOSUB200                        :rem 163
110 GOSUB300                        :rem 165
120 T(C)=T(C)+S(C)                  :rem 178
130 GOSUB400:FORI=1TO10:B$(I)="":NEXT
                                    :rem 184
140 GOTO90                          :rem 55
200 FORI=1TOLEN(W$)                 :rem 126
210 A$(I)=MID$(W$,I,1)              :rem 107
220 NEXT                            :rem 211
230 C$="":FORI=1TOLEN(W$)           :rem 163
240 R=INT(RND(1)*LEN(W$)+1)         :rem 248
250 IFB$(R)<>""THEN240              :rem 178
260 B$(R)=A$(I)                     :rem 221
270 NEXT                            :rem 216
271 FORI=1TOLEN(W$):C$=C$+B$(I):NEXT
                                    :rem 111
272 IFC$=W$ANDLEN(W$)<>1THENFORI=1TOLEN(W
   $):B$(I)="":NEXT:GOTO230         :rem 201
275 PRINT"{CLR}{5 DOWN}{RVS}{7 RIGHT}
   {RED}WORD HAS{11 RIGHT}BEEN SCRAMBLED
   {OFF}{BLU}"                      :rem 255
280 PRINT"{6 DOWN}{3 RIGHT}{GRN}PRESS SPA
   CE BAR{9 RIGHT}WHEN READY{BLU}"
                                    :rem 223
285 GETC$:IFC$=""THEN285            :rem 101
290 PRINT"{CLR}{5 DOWN}{7 RIGHT}";:rem 90
295 FORI=1TOLEN(W$):PRINT"{RED}";B$(I);:N
   EXT                              :rem 162
298 RETURN                          :rem 131
300 X=51:S(C)=50                    :rem 81
310 TI$="000000"                    :rem 246
320 PRINT:PRINT:PRINT:PRINT         :rem 119
325 SC=7885:CC=38605                :rem 76
330 FORI=1TOLEN(W$)                 :rem 130
335 POKESC,99:POKECC,2              :rem 75
340 GETC$                           :rem 222
350 PRINT"{HOME}{RVS}"MID$(TI$,4,1)"
   {OFF}MINUTES{2 SPACES}{RVS}"RIGHT$(TI
   $,2)" {OFF}SECONDS"              :rem 95
355 IFTI$="000300"THENGOSUB500:GOTO390
                                    :rem 228
360 IFC$=""THEN340                  :rem 214
365 PRINT"{4 DOWN}"                 :rem 179
370 IFC$=A$(I)THENPRINTTAB(X)A$(I);:POKE3
   6875,200:FORT=1TO100:NEXT:POKE36875,0
   :GOTO380                         :rem 230
375 IFS(C)<10THENGOSUB550:GOTO390 :rem 10
378 IFC$<>A$(I)THENS(C)=S(C)-10:POKE36877
   ,220:FORT=1TO100:NEXT:POKE36877,0:GOT
   O335                             :rem 131
380 X=X+1:SC=SC+1:CC=CC+1:NEXT      :rem 59
390 RETURN                          :rem 124
400 IFC<>1THENC=1:RETURN            :rem 11
410 PRINT"{CLR}{5 DOWN}{7 RIGHT}{RED}
   {RVS}SCORES{OFF}{BLU}"           :rem 199
420 PRINT"{7 RIGHT}[6 T]"           :rem 4
430 PRINT"{DOWN}"P$(0),P$(1)        :rem 144
440 PRINTT(1),T(0)                  :rem 252
450 C=0:RETURN                      :rem 99
500 PRINT"{CLR}{4 DOWN}{3 RIGHT}{RVS}
   {RED}YOUR TIME IS UP.{OFF}{BLU}"
                                    :rem 96
510 PRINT"{2 DOWN}{2 RIGHT}WORD WAS:"W$".
   ":S(C)=0                         :rem 159
520 FORT=1TO5000:NEXT:RETURN        :rem 59
550 PRINT"{RVS}{RED}{2 DOWN}YOU RAN OUT O
   F POINTS.{OFF}{BLU}"             :rem 226
560 PRINT"{2 DOWN}{PUR}WORD WAS:{BLU}"W$"
   ."                               :rem 151
570 FORT=1TO2000:NEXT               :rem 35
580 RETURN                          :rem 125
1000 FORS=250TO235STEP-1:POKE36874,S:POKE
     36878,S-235:FORT=1TO100:NEXTT,S
                                    :rem 188
1010 POKE36874,0:POKE36878,15:RETURN
                                    :rem 126
```

Program 2:
Word Scramble—64 Version

```
1 POKE53280,6:POKE53281,1                :rem 141
5 SN=54272                               :rem 23
6 POKESN+24,15:POKESN+5,17:POKESN+6,240:P
  OKESN,100                              :rem 27
10 PRINT"{CLR}":PRINT"{RED}{9 DOWN}
   {13 RIGHT}WORD SCRAMBLE"              :rem 131
20 GOSUB1000:PRINT"{CLR}"                :rem 65
25 PRINT"{RED}{2 DOWN}EACH PLAYER TAKES A
    TURN ENTERING A{5 SPACES}COMMON ";
                                         :rem 247
30 PRINT"WORD (A MAXIMUM OF 10 LETTERS)."
                                         :rem 103
35 PRINT"{DOWN}THE COMPUTER WILL THEN SCR
   AMBLE THE WORD";                      :rem 162
40 PRINT"AND PRINT IT."                  :rem 96
45 PRINT"{DOWN}YOU HAVE THREE MINUTES TO
   {SPACE}FIND IT."                      :rem 152
50 PRINT"{DOWN}IF FOUND WITHIN THE ALLOTT
   ED TIME, YOU"                         :rem 183
55 PRINT"WILL BE GIVEN 50 POINTS."
                                         :rem 227
60 PRINT"{DOWN}FOR EVERY WRONG GUESS THAT
    YOU MAKE, YOUWILL LOSE 10 POINTS.
   {BLU}"                                :rem 57
65 PRINT"{3 DOWN}{7 RIGHT}{RVS}{PUR}PRESS
    SPACE BAR WHEN READY{OFF}"           :rem 239
70 IFPEEK(197)<>60THEN70                 :rem 131
75 POKE198,0                             :rem 153
80 PRINT"{CLR}{4 DOWN}{GRN}PLAYER # 1'S N
   AME{BLU}":INPUTP$(0)                  :rem 200
85 PRINT"{3 DOWN}{PUR}PLAYER # 2'S NAME
   {BLU}":INPUTP$(1)                     :rem 169
90 PRINT"{HOME}{16 DOWN}{RED}";P$(C);" EN
   TER WORD TO BE SCRAMBLED:{OFF}{BLU}"
                                         :rem 151
92 W$="":INPUTW$:IFW$=""THENPRINT"{UP}";:
   GOTO92                                :rem 27
95 IFLEN(W$)>10THENPRINT"{RVS}{GRN}NO MOR
   E THAN 10 LETTERS{OFF}{BLU}":GOTO90
                                        :rem 131
100 GOSUB200                            :rem 163
110 GOSUB300                            :rem 165
120 T(C)=T(C)+S(C)                      :rem 178
130 GOSUB400:FORI=1TO10:B$(I)="":NEXT
                                        :rem 184
140 GOTO90                               :rem 55
200 FORI=1TOLEN(W$)                     :rem 126
210 A$(I)=MID$(W$,I,1)                  :rem 107
220 NEXT                                :rem 211
230 C$="":FORI=1TOLEN(W$)               :rem 163
240 R=INT(RND(1)*LEN(W$)+1)             :rem 248
250 IFB$(R)<>""THEN240                  :rem 178
260 B$(R)=A$(I)                         :rem 221
270 NEXT                                :rem 216
271 FORI=1TOLEN(W$):C$=C$+B$(I):NEXT
                                        :rem 111
272 IFC$=W$ANDLEN(W$)<>1THENFORI=1TOLEN(W
    $):B$(I)="":NEXT:GOTO230            :rem 201
275 PRINT"{CLR}{5 DOWN}{8 SPACES}{RVS}
    {RED}WORD HAS BEEN SCRAMBLED.{OFF}
    {BLU}"                              :rem 35
280 POKE 198,0:PRINT"{6 DOWN}{7 SPACES}
    {GRN}PRESS SPACE BAR WHEN READY{BLU}"
                                        :rem 234
285 IFPEEK(197)<>60THEN285              :rem 243
290 PRINT"{CLR}{5 DOWN}{15 RIGHT}";
                                        :rem 66
295 FORI=1TOLEN(W$):PRINT"{RED}";B$(I);:N
    EXT                                 :rem 162
298 POKE198,0:RETURN                    :rem 234
300 X=95:S(C)=50                         :rem 89
310 TI$="000000"                        :rem 246
320 PRINT:PRINT:PRINT:PRINT             :rem 119
325 SC=1399:CC=SC+54272                   :rem 5
330 FORI=1TOLEN(W$)                     :rem 130
335 POKESC,99:POKECC,2                   :rem 75
340 GETC$                               :rem 222
350 PRINT"{HOME}{RVS}{9 RIGHT}"MID$(TI$,4
    ,1)" {OFF}MINUTES{2 SACES}{RVS}"RIGH
    T$(TI$,2)" {OFF}SECONDS"            :rem 100
355 IFTI$="000300"THENGOSUB500:GOTO390
                                        :rem 228
360 IFC$=""THEN340                      :rem 214
365 PRINT"{4 DOWN}"                     :rem 179
370 IFC$=A$(I)THENPRINTTAB(X)A$(I);:BY=50
    :LN=50:GOSUB600:GOTO380             :rem 141
375 IFS(C)<10THENGOSUB550:GOTO390       :rem 10
378 IFC$<>A$(I)THENS(C)=S(C)-10:BY=20:LN=
    120:GOSUB600:GOTO335                :rem 79
380 X=X+1:SC=SC+1:CC=CC+1:NEXT          :rem 59
390 RETURN                              :rem 124
400 IFC<>1THENC=1:RETURN                 :rem 11
410 PRINT"{CLR}{5 DOWN}{17 RIGHT}{RED}
    {RVS}SCORES{OFF}{BLU}"              :rem 233
420 PRINT"{17 RIGHT}{6 T}"             :rem 38
430 PRINT"{DOWN}{10 RIGHT}"P$(0);TAB(25);
    P$(1)                               :rem 139
440 PRINT"{9 RIGHT}"T(1);TAB(24);T(0)
                                        :rem 29
450 C=0:RETURN                           :rem 99
500 PRINT"{CLR}{4 DOWN}{12 RIGHT}{RVS}
    {RED}YOUR TIME IS UP{OFF}{BLU}"
                                        :rem 55
510 PRINT"{2 DOWN}{10 RIGHT}WORD WAS "W$"
    .":S(C)=0                           :rem 77
520 FORT=1TO5000:NEXT:RETURN            :rem 59
550 PRINT"{RVS}{RED}{2 DOWN}{9 RIGHT}YOU
    {SPACE}RAN OUT OF POINTS{OFF}{BLU}"
                                        :rem 185
560 PRINT"{2 DOWN}{PUR}{10 RIGHT}WORD WAS
     {BLU}"W$"."                        :rem 127
570 FORT=1TO2000:NEXT                    :rem 35
580 RETURN                              :rem 125
600 POKESN+1,BY:POKESN+4,33:FORQQ=1TOLN:N
    EXT:POKESN+4,32:RETURN              :rem 127
1000 FORBY=50TO20STEP-1:LN=20:GOSUB600:NE
     XT:FORI=1TO500:NEXT                 :rem 73
1010 RETURN                             :rem 162
```

The Beginner's Corner

See article on page 88.

Program 1: Tic-Tac-Toe (64 Version)

```
100 REM TIC-TAC-TOE                     :rem 111
110 GOTO 560                            :rem 102
120 REM X                               :rem 207
130 POKE M,A:POKE M+4,A:POKE M+41,A
                                        :rem 178
140 POKE M+43,A:POKE M+82,A:POKE M+121,A
                                        :rem 170
150 POKE M+123,A:POKE M+160,A:POKE M+164,
    A                                    :rem 14
160 RETURN                              :rem 119
170 REM O                               :rem 203
```

```
180 POKE M,233:POKE M+4,223:POKE M+160,95
    :POKE M+164,105                    :rem 1
190 FOR I=1 TO 3                       :rem 14
200 POKE M+I,A:POKE M+I+160,A          :rem 72
210 POKE M+40*I,A:POKE M+40*I+4,A      :rem 2
220 NEXT I                             :rem 28
230 POKE M+82,32                       :rem 55
240 RETURN                             :rem 118
250 FOR I=M+C TO M+C+160 STEP 40       :rem 127
260 POKE I,CC(N):POKE I+1,CC(N):POKE I+2,
    CC(N):POKE I+3,CC(N):POKE I+4,CC(N)
                                       :rem 227
270 NEXT I                             :rem 33
280 RETURN                             :rem 122
290 REM TONE                           :rem 181
300 POKE F1,42:POKE F2,62              :rem 64
310 POKE W,17                          :rem 174
320 FOR D=1 TO 60:NEXT D:POKE W,0      :rem 16
330 RETURN                             :rem 118
340 X=N:P(E)=X:M=S(E):ON X GOSUB 130,130,
    180                                :rem 164
350 GOSUB 250                          :rem 175
360 FOR I=1 TO 7 STEP 3                :rem 128
370 IF P(I)<>P(I+1) THEN 400           :rem 150
380 IF P(I)<>P(I+2) THEN 400           :rem 152
390 ON P(I)+1 GOTO 400,1550,1550,1550
                                       :rem 49
400 NEXT I                             :rem 28
410 FOR I=1 TO 3                       :rem 9
420 IF P(I)<>P(I+3) THEN 450           :rem 153
430 IF P(I)<>P(I+6) THEN 450           :rem 157
440 ON P(I)+1 GOTO 450,1550,1550,1550
                                       :rem 50
450 NEXT I                             :rem 33
460 IF P(5)<>X THEN 490                :rem 157
470 IF (P(1)=X) AND (P(9)=X) THEN 1550
                                       :rem 111
480 IF (P(3)=X) AND (P(7)=X) THEN 1550
                                       :rem 112
490 FOR I=1 TO 9                       :rem 23
500 IF P(I)=0 THEN 540                 :rem 67
510 NEXT I                             :rem 30
520 PRINT "TIE GAME!"                  :rem 133
530 GOTO 1590                          :rem 160
540 N=ABS(N-4)                         :rem 250
550 RETURN                             :rem 122
560 PRINT "{CLR}{BLU}":POKE 53281,1
                                       :rem 231
570 W=54276:POKE W,0                   :rem 84
580 PRINT TAB(15)"TIC-TAC-TOE"         :rem 246
590 FOR I=1 TO 9:READ S(I):NEXT I      :rem 29
600 DATA 1114,1122,1130,1434,1442      :rem 73
610 DATA 1450,1754,1762,1770           :rem 111
620 PRINT "{2 DOWN}CHOOSE ONE OF THE POSI
    TION NUMBERS."                     :rem 99
630 PRINT "{DOWN}TRY TO GET THREE IN A RO
    W."                                :rem 115
640 POKE 54296,15:F1=54273:F2=54272:C=F2
                                       :rem 125
650 POKE W+1,128:POKE W+2,16           :rem 244
660 A=160:CC(1)=6:CC(3)=10             :rem 72
670 PRINT "{DOWN}ALTERNATE TURNS WITH THE
     COMPUTER."                        :rem 117
680 PRINT "{DOWN}{BLK}X{BLU} GETS FIRST M
    OVE."                              :rem 167
690 Y=3:Z=1:CH=2                       :rem 158
700 PRINT "{2 DOWN}CHOOSE{2 SPACES}{BLK}F
    1{BLU} FOR {BLK}X{BLU}"            :rem 95
710 PRINT TAB(8)"{BLK}F3{BLU} FOR {BLK}O
    {BLU}"                             :rem 214
720 GET E$:IF E$<>"{F1}" AND E$<>"{F3}" T
    HEN 720                            :rem 159
730 IF E$="{F1}"THEN Y=1:Z=3           :rem 146
740 POKE 198,0                         :rem 200
750 PRINT "{2 DOWN}CHOOSE{2 SPACES}{BLK}F
    1{BLU} BEGINNER"                   :rem 192
760 PRINT TAB(8)"{BLK}F3{BLU} INTERMEDIAT
    E"                                 :rem 113
770 GET E$:IF E$<>"{F1}" AND E$<>"{F3}" T
    HEN 770                            :rem 169
780 IF E$="{F1}" THEN CH=1             :rem 197
790 PRINT "{CLR}":CT=INT(14*RND(0))+2
                                       :rem 159
800 FOR I=1 TO 9:P(I)=0:NEXT I         :rem 101
810 FOR I=1080 TO 1960 STEP 40         :rem 226
820 POKE I,A:POKE I+C,CT               :rem 98
830 POKE I+8,A:POKE I+8+C,CT           :rem 41
840 NEXT I                             :rem 36
850 FOR I=1353 TO 1375                 :rem 73
860 POKE I,A:POKE I+C,CT:POKE I+320,A:POK
    E I+320+C,CT                       :rem 232
870 NEXT I                             :rem 39
880 FOR I=1 TO 9:POKE S(I)+82,I+48:rem 17
890 POKE S(I)+82+C,0:NEXT I            :rem 222
900 N=1                                :rem 85
910 IF CH=2 THEN 1040                  :rem 23
920 IF Y=1 THEN 970{5 SPACES}          :rem 192
930 REM COMPUTER'S MOVE                :rem 160
940 E=INT(9*RND(0)+1):IF P(E)<>0 THEN 940
                                       :rem 164
950 GOSUB 340                          :rem 181
960 REM YOUR MOVE                      :rem 9
970 GOSUB 300                          :rem 179
980 GET E$:IF E$=""THEN 980            :rem 109
990 IF E$<"1" OR E$>"9" THEN 980 :rem 224
1000 E=VAL(E$):IF P(E)<>0 THEN 970 :rem 8
1010 GOSUB 340                         :rem 217
1020 ON CH GOTO 940,1060               :rem 180
1030 REM COMPUTER'S MOVE               :rem 200
1040 IF Y=1 THEN 970                   :rem 234
1050 REM CENTER POSITION               :rem 224
1060 IF P(5)=0 THEN E=5:GOTO 1520:rem 186
1070 IF P(5)=X THEN 1140               :rem 183
1080 REM DIAGONALS                     :rem 63
1090 IF P(1)=0 AND P(9)=Z THEN E=1:GOTO 1
    520                                :rem 249
1100 IF P(1)=Z AND P(9)=0 THEN E=9:GOTO 1
    520                                :rem 249
1110 IF P(3)=0 AND P(7)=Z THEN E=3:GOTO 1
    520                                :rem 244
1120 IF P(3)=Z AND P(7)=0 THEN E=7:GOTO 1
    520                                :rem 249
1130 REM COLUMNS                       :rem 202
1140 FOR K=1 TO 3                      :rem 60
1150 IF P(K)=X OR P(K+3)=X OR P(K+6)=X TH
    EN 1200                            :rem 204
1160 IF P(K)+P(K+3)+P(K+6)<>2*Z THEN 1200
                                       :rem 82
1170 IF P(K)=0 THEN E=K:GOTO 1520:rem 232
1180 IF P(K+3)=0 THEN E=K+3:GOTO 1520
                                       :rem 165
1190 E=K+6:GOTO 1520                   :rem 52
1200 NEXT K                            :rem 77
1210 REM ROWS                          :rem 243
1220 FOR K=1 TO 7 STEP 3               :rem 174
1230 IF P(K)=X OR P(K+1)=X OR P(K+2)=X TH
    EN 1280                            :rem 205
1240 IF P(K)+P(K+1)+P(K+2)<>2*Z THEN 1280
                                       :rem 83
1250 IF P(K)=0 THEN E=K:GOTO 1520:rem 231
1260 IF P(K+1)=0 THEN E=K+1:GOTO 1520
                                       :rem 160
```

```
1270 E=K+2:GOTO 1520                    :rem 47
1280 NEXT K                             :rem 85
1290 REM PREVENT COLUMN                 :rem 162
1300 FOR K=1 TO 3                       :rem 58
1310 IF P(K)=Z OR P(K+3)=Z OR P(K+6)=Z TH
     EN 1360                            :rem 215
1320 IF P(K)+P(K+3)+P(K+6)<>2*X THEN 1360
                                        :rem 85
1330 IF P(K)=0 THEN E=K:GOTO 1520:rem 230
1340 IF P(K+3)=0 THEN E=K+3:GOTO 1520
                                        :rem 163
1350 E=K+6:GOTO 1520                    :rem 50
1360 NEXT K                             :rem 84
1370 REM PREVENT ROW                    :rem 203
1380 FOR K=1 TO 7 STEP 3                :rem 181
1390 IF P(K)=Z OR P(K+1)=Z OR P(K+2)=Z TH
     EN 1440                            :rem 216
1400 IF P(K)+P(K+1)+P(K+2)<>2*X THEN 1440
                                        :rem 77
1410 IF P(K)=0 THEN E=K:GOTO 1520:rem 229
1420 IF P(K+1)=0 THEN E=K+1:GOTO 1520
                                        :rem 158
1430 E=K+2:GOTO 1520                    :rem 45
1440 NEXT K                             :rem 83
1450 IF P(5)=Z THEN 1500                :rem 187
1460 IF P(1)=0 AND P(9)=X THEN E=1:GOTO 1
     520                                :rem 248
1470 IF P(1)=X AND P(9)=0 THEN E=9:GOTO 1
     520                                 :rem 1
1480 IF P(3)=0 AND P(7)=X THEN E=3:GOTO 1
     520                                :rem 252
1490 IF P(3)=X AND P(7)=0 THEN E=7:GOTO 1
     520                                 :rem 1
1500 GOTO 940                           :rem 156
1510 REM                                :rem 171
1520 GOSUB 340                          :rem 223
1530 GOTO 970                           :rem 162
1540 REM GAME OVER                        :rem 4
1550 IF X<>Y THEN 1580                  :rem 130
1560 PRINT "YOU WIN!!!"                 :rem 235
1570 GOTO 1590                          :rem 213
1580 PRINT "COMPUTER WINS!!"            :rem 145
1590 FOR I=1 TO 20                      :rem 114
1600 POKE F1,INT(60*RND(0)+30)           :rem 88
1610 POKE F2,INT(200*RND(0)+10)         :rem 132
1620 GOSUB 310                          :rem 221
1630 NEXT I                              :rem 82
1640 PRINT "{22 DOWN}TRY AGAIN? (Y/N)";
                                        :rem 18
1650 GET E$:IF E$="N" THEN 1690          :rem 21
1660 IF E$<>"Y" THEN 1650               :rem 215
1670 CLR                                :rem 175
1680 GOTO 560                           :rem 163
1690 PRINT "{CLR}"                       :rem 52
1700 END                                :rem 159
```

Program 2: Tic-Tac-Toe (VIC Version)

```
1 GOTO28                                :rem 212
2 POKEM,77:POKEM+3,78:POKEM+23,77:POKEM+2
  4,78:POKEM+45,78                      :rem 158
3 POKEM+46,77:POKEM+66,78:POKEM+69,77:RET
  URN                                   :rem 202
4 POKEM,85:POKEM+1,67:POKEM+2,67:POKEM+3,
  73:POKEM+22,66:POKEM+23,32:POKEM+25,93
                                        :rem 222
5 POKEM+44,66:POKEM+47,93:POKEM+66,74:POK
  EM+67,64:POKEM+68,64:POKEM+69,75:RETURN
                                        :rem 111
6 FORI=M+C TOM+C+66 STEP22:POKEI,CC(N):PO
  KEI+1,CC(N):POKEI+2,CC(N)             :rem 181
7 POKEI+3,CC(N):NEXT:RETURN             :rem 241
8 POKEF1,231                            :rem 160
9 FORD=1TO60:NEXT:POKEF1,0:RETURN:rem 170
10 X=N:P(E)=X:M=S(E):ONX GOSUB2,2,4
                                        :rem 69
11 GOSUB6                               :rem 24
12 FORI=1TO7STEP3                       :rem 74
13 IFP(I)<>P(I+1)THEN16                 :rem 51
14 IFP(I)<>P(I+2)THEN16                 :rem 53
15 ONP(I)+1GOTO16,89,89,89              :rem 192
16 NEXT                                 :rem 166
17 FORI=1TO3                            :rem 220
18 IFP(I)<>P(I+3)THEN21                  :rem 54
19 IFP(I)<>P(I+6)THEN21                  :rem 58
20 ONP(I)+1GOTO21,89,89,89              :rem 184
21 NEXT                                 :rem 162
22 IFP(5)<>X THEN25                      :rem 49
23 IF(P(1)=X)AND(P(9)=X)THEN89          :rem 223
24 IF(P(3)=X)AND(P(7)=X)THEN89          :rem 224
25 FORI=1TO9:IFP(I)=0THEN27             :rem 153
26 NEXT:PRINT"TIE GAME!":GOTO92         :rem 173
27 N=ABS(N-4):RETURN                    :rem 228
28 PRINT"{CLR}{BLU}":PRINTTAB(5)"TIC-TAC-
   TOE":FORI=1TO9:READS(I):NEXT         :rem 191
29 DATA7726,7733,7740,7880,7887,7894,8034
   ,8041,8048                           :rem 98
30 PRINT"{2 DOWN}CHOOSE ONE OF THE":PRINT
   "POSITION NUMBERS.":PRINT"{DOWN}GET 3
   {SPACE}IN A ROW."                    :rem 102
31 POKE36878,15:F1=36876:C=30720:CC(1)=6:
   CC(3)=2:Y=3:Z=1:H=2                  :rem 69
32 PRINT"{DOWN}{BLK}X{BLU} GETS FIRST MOV
   E.":PRINT"{DOWN}CHOOSE {BLK}F1{BLU} FO
   R {BLK}X":PRINTTAB(7)"F3{BLU} FOR
   {BLK}0{BLU}"                          :rem 8
33 GETE$:IFE$<>"{F1}"ANDE$<>"{F3}"THEN33
                                        :rem 57
34 IFE$="{F1}"THENY=1:Z=3               :rem 95
35 PRINT"{2 DOWN}CHOOSE {BLK}F1{BLU} BEGI
   NNER":PRINTTAB(7)"{BLK}F3{BLU} INTERME
   DIATE"                               :rem 153
36 GETE$:IFE$<>"{F1}"ANDE$<>"{F3}"THEN36
                                        :rem 63
37 IFE$="{F1}"THENH=1                   :rem 77
38 PRINT"{CLR}":T=INT(6*RND(0))+2:FORI=1T
   O9:P(I)=0:NEXT                       :rem 182
39 FORI=7709TO8129STEP22:POKEI,160:POKEI+
   C,T:POKEI+7,160:POKEI+7+C,T:NEXT
                                        :rem 46
40 FORI=7835TO7854:POKEI,160:POKEI+C,T:PO
   KEI+154,160:POKEI+154+C,T:NEXT       :rem 80
41 FORI=1TO9:POKES(I)+23,I+48:POKES(I)+23
   +C,0:NEXT                            :rem 250
42 N=1:IFH=2THEN51                       :rem 55
43 IFY=1THEN46                          :rem 86
44 E=INT(9*RND(0)+1):IFP(E)<>0THEN44
                                        :rem 58
45 GOSUB10                              :rem 74
46 GOSUB8                               :rem 34
47 GETE$:IFE$=""THEN47                   :rem 1
48 IFE$<"1"ORE$>"9"THEN47               :rem 116
49 E=VAL(E$):IFP(E)<>0THEN46            :rem 126
50 GOSUB10:ONH GOTO44,52                :rem 153
51 IFY=1THEN46                           :rem 85
52 IFP(5)=0THENE=5:GOTO88                :rem 2
53 IFP(5)=X THEN58                      :rem 254
54 IFP(1)=0ANDP(9)=Z THENE=1:GOTO88
                                        :rem 64
55 IFP(1)=Z ANDP(9)=0THENE=9:GOTO88
                                        :rem 73
56 IFP(3)=0ANDP(7)=Z THENE=3:GOTO88
                                        :rem 68
```

```
57 IFP(3)=Z ANDP(7)=ØTHENE=7:GOTO88
                                    :rem 73
58 FORK=1TO3:IF(P(K)=X)+(P(K+3)=X)+(P(K+6
   )=X)THEN63                       :rem 207
59 IFP(K)+P(K+3)+P(K+6)<>2*Z THEN63
                                    :rem 158
60 IF P(K)=ØTHENE=K:GOTO88          :rem 45
61 IFP(K+3)=ØTHENE=K+3:GOTO88       :rem 234
62 E=K+6:GOTO88                     :rem 121
63 NEXT                             :rem 168
64 FORK=1TO7STEP3:IF(P(K)=X)+(P(K+1)=X)+(
   P(K+2)=X)THEN69                  :rem 63
65 IFP(K)+P(K+1)+P(K+2)<>2*Z THEN69
                                    :rem 155
66 IFP(K)=ØTHENE=K:GOTO88           :rem 51
67 IFP(K+1)=ØTHENE=K+1:GOTO88       :rem 236
68 E=K+2:GOTO88                     :rem 123
69 NEXT                             :rem 174
70 FORK=1TO3:IF(P(K)=Z)+(P(K+3)=Z)+(P(K+6
   )=Z)THEN75                       :rem 210
71 IFP(K)+P(K+3)+P(K+6)<>2*X THEN75
                                    :rem 153
72 IFP(K)=ØTHENE=K:GOTO88           :rem 48
73 IFP(K+3)=ØTHENE=K+3:GOTO88       :rem 237
74 E=K+6:GOTO88                     :rem 124
75 NEXT                             :rem 171
76 FORK=1TO7STEP3:IF(P(K)=Z)+(P(K+1)=Z)+(
   P(K+2)=Z)THEN81                  :rem 66
77 IFP(K)+P(K+1)+P(K+2)<>2*X THEN81
                                    :rem 150
78 IFP(K)=ØTHEN E=K:GOTO88          :rem 54
79 IFP(K+1)=ØTHENE=K+1:GOTO88       :rem 239
80 E=K+2:GOTO88                     :rem 117
81 NEXT                             :rem 168
82 IFP(5)=Z THEN87                   :rem 4
83 IFP(1)=ØANDP(9)=X THENE=1:GOTO88
                                    :rem 64
84 IFP(1)=X ANDP(9)=ØTHENE=9:GOTO88
                                    :rem 73
85 IFP(3)=ØANDP(7)=X THENE=3:GOTO88
                                    :rem 68
86 IFP(3)=X ANDP(7)=ØTHENE=7:GOTO88
                                    :rem 73
87 GOTO44                           :rem 16
88 GOSUB1Ø:GOTO46                   :rem 46
89 IFX<>Y THEN91                    :rem 196
90 PRINT"YOU WIN!!!":GOTO92         :rem 102
91 PRINT"COMPUTER WINS!!"            :rem 45
92 FORI=1TO2Ø:POKEF1,INT(2Ø*RND(Ø)+2ØØ):G
   OSUB9:NEXT                       :rem 112
93 PRINT"{2Ø DOWN}TRY AGAIN? (Y/N)";
                                    :rem 145
94 GETE$:IFE$="N"THEN97              :rem 86
95 IFE$<>"Y"THEN94                   :rem 25
96 CLR:GOTO28                        :rem 45
97 PRINT"{CLR}":END                 :rem 229
```

Inside Random Numbers

See article on page 98.

BEFORE TYPING...

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

Random Number Test

```
1 PRINT"{CLR}{DOWN} PRESS {RVS}R{OFF} TO
  {SPACE}RANDOMIZE."                :rem 163
2 GETA$:IFA$=""THEN2                :rem 135
3 IFA$="R"THENX=RND(-TI)            :rem 189
5 POKE56,27:CLR:PRINT"{CLR}{DOWN} PLEASE
  {SPACE}WAIT..."                   :rem 167
7 B=7168:C=7679:FORA=BTOC:POKEA,Ø:NEXT
                                    :rem 105
10 B=885:C=947:FORA=BTOC:READD:POKEA,D:NE
   XT                               :rem 215
20 PRINT"{CLR}{DOWN} {RVS}V{OFF}IC OR 64?
   "                                :rem 167
25 PRINT"{DOWN} PRESS {RVS}V{OFF} FOR VIC
   "                                :rem 153
26 PRINT" {RVS}RETURN{OFF} FOR 64."
                                    :rem 60
30 GETA$:IFA$=""THEN3Ø              :rem 233
35 IFA$<>"V"THEN5Ø                    :rem 4
40 POKE888,39:POKE892,148           :rem 168
50 PRINT"{CLR}{DOWN} PRESS {RVS}F1{OFF} T
   O TEST"                          :rem 101
51 PRINT" RANDOM SEEDS."            :rem 154
60 PRINT"{DOWN} PRESS {RVS}F3{OFF} TO TES
   T"                               :rem 213
61 PRINT" BASIC {RVS}RND{OFF} COMMAND."
                                    :rem 79
70 GETA$:IFA$=""THEN7Ø              :rem 241
80 IFA$="{F1}"THENGOTO1ØØØ          :rem 139
90 IFA$="{F3}"THENGOTO2ØØØ          :rem 142
100 GOTO5Ø                          :rem 47
885 DATA165,197,201,4,240,55,32,151
                                    :rem 243
893 DATA224,165,143,133,251,169,254,133
                                    :rem 191
901 DATA253,169,27,133,254,160,255,24
                                    :rem 86
909 DATA165,253,105,2,133,253,165,254
                                    :rem 86
917 DATA105,Ø,133,254,2ØØ,196,251,2Ø8
                                    :rem 75
925 DATA238,160,Ø,24,177,253,105,1
                                    :rem 184
933 DATA145,253,2ØØ,177,253,105,Ø,145
                                    :rem 76
941 DATA253,201,255,2Ø8,195,96,Ø,Ø
                                    :rem 189
1ØØØ PRINT"{CLR}{DOWN} PRESS 1-5 TO TEST"
                                    :rem 57
1Ø1Ø PRINT" RANDOM SEEDS."          :rem 246
1Ø2Ø PRINT"{2 DOWN} {RVS}PRESS{4 SPACES}T
     EST BYTE:{OFF}"                :rem 149
1Ø3Ø PRINT"{DOWN}{3 SPACES}1{9 SPACES}139
     "                              :rem 116
1Ø4Ø PRINT"{3 SPACES}2{9 SPACES}14Ø"
                                    :rem 93
1Ø5Ø PRINT"{3 SPACES}3{9 SPACES}141"
                                    :rem 96
1Ø6Ø PRINT"{3 SPACES}4{9 SPACES}142"
                                    :rem 99
1Ø7Ø PRINT"{3 SPACES}5{9 SPACES}143"
                                    :rem 102
1Ø8Ø GETA$:IFA$=""THEN1Ø8Ø          :rem 181
1Ø9Ø A=VAL(A$):IFA<1ORA>5THEN1ØØØ:rem 157
11ØØ POKE895,138+A                  :rem 203
1110 PRINT"{CLR}{DOWN} TESTING RANDOM SEE
     D"                             :rem 56
112Ø PRINT" BYTE ";A+138;"{LEFT}.":rem 86
113Ø PRINT"{DOWN} PRESS {RVS}F1{OFF} TO S
     TOP."                          :rem 102
114Ø SYS885:GOTO3ØØØ                :rem 160
2ØØØ POKE945,4:POKE946,169:POKE947,255:PO
```

```
     KE948,133:POKE949,176:POKE950,96
                                    :rem 236
2002 FORA=891TO897:POKEA,234:NEXT:rem 254
2005 PRINT"{CLR}{DOWN} TESTING BASIC
     {RVS}RND{OFF}"                 :rem 68
2010 PRINT" COMMAND."               :rem 193
2020 PRINT"{DOWN} ENTER NUMBER RANGE:"
                                    :rem 148
2030 PRINT" (0 - 255)"              :rem 224
2035 PRINT"{DOWN} ENTER TWO NUMBERS"
                                    :rem 64
2036 PRINT" SEPARATED BY A COMMA.{DOWN}"
                                    :rem 189
2040 INPUTA,B                        :rem 5
2050 IFA<0ORA>255THEN2000           :rem 176
2060 IFB<0ORB>255THEN2000           :rem 179
2070 PRINT"{CLR}{DOWN} TESTING {RVS}RND
     {OFF} COMMAND."                :rem 17
2080 PRINT"{DOWN} RANGE= ";A;"-";B
                                    :rem 251
2090 PRINT"{2 DOWN} PRESS {RVS}F1{OFF} TO
      STOP."                        :rem 125
2100 X=251:Y=176:Z=255:R=(ABS(A-B))+1:POK
     EY,0                           :rem 54
2110 IFA<BTHENL=A                   :rem 11
2120 IFB<ATHENL=B                   :rem 13
2130 POKEX,INT((RND(1)*R)+L):SYS885:IFPEE
     K(Y)=ZTHEN3000                 :rem 94
2140 GOTO2130                       :rem 198
3000 POKE198,0:GOSUB3500            :rem 114
3010 B=7168:C=7678:Q=256:L=20:FORA=BTOCST
     EP2                            :rem 110
3020 PRINTCT,PEEK(A)+Q*PEEK(A+1):N=N+1:CT
     =CT+1:IFN=>LTHENGOSUB3200      :rem 71
3030 NEXT                            :rem 5
3100 PRINT"{2 DOWN}{2 SPACES}(PRESS ANY K
     EY)"                           :rem 102
3110 GETA$:IFA$=""THEN3110          :rem 173
3120 GOTO4000                       :rem 195
3200 N=0                            :rem 128
3210 GETA$:IFA$=""THEN3210          :rem 175
3220 GOSUB3500:RETURN               :rem 41
3500 PRINT"{CLR}NUMBER","COUNT{DOWN}":RET
     URN                            :rem 25
4000 PRINT"{CLR}{DOWN} REQUEST ANALYSIS?"
                                    :rem 5
4002 PRINT"{DOWN} (PRESS {RVS}Y{OFF} OR
     {RVS}N{OFF})"                  :rem 22
4004 GETA$:IFA$=""THEN4004          :rem 179
4006 IFA$<>"Y"THEN5000              :rem 201
4008 PRINT"{CLR}{DOWN} PLEASE WAIT..."
                                    :rem 186
4009 B=7168:C=7678:D=2:TT=0:CT=0:HI=0:LO=
     65535:ZZ=0:G=256               :rem 58
4010 FORA=BTOCSTEPD                 :rem 210
4020 N=PEEK(A)+G*PEEK(A+1)          :rem 183
4030 IFN=0THENZZ=ZZ+1:GOTO4040      :rem 124
4035 GOSUB4500                      :rem 21
4040 NEXT                            :rem 7
4100 PRINT"{CLR}{DOWN} {RVS}RANDOM ANALYS
     IS:"                           :rem 171
4110 PRINT"{DOWN} TOTAL NUMBERS HIT:"
                                    :rem 103
4120 PRINT"{3 SPACES}";CT           :rem 106
4130 PRINT"{DOWN} TOTAL COUNT:"     :rem 241
4140 PRINT"{3 SPACES}";TT           :rem 125
4150 PRINT"{DOWN} AVERAGE COUNT:":rem 106
4160 PRINT"{3 SPACES}";TT/CT         :rem 69
4170 PRINT"{DOWN} LOWEST COUNT:"    :rem 79
4180 PRINT"{3 SPACES}";LO           :rem 116
4190 PRINT"{DOWN} HIGHEST COUNT:":rem 127
4200 PRINT"{3 SPACES}";HI           :rem 99
4210 PRINT"{DOWN} TOTAL ZEROES:"    :rem 63
4220 PRINT"{3 SPACES}";ZZ           :rem 136
4230 CLR:PRINT"{DOWN}{RVS}R{OFF} TO REPEA
     T ANALYSIS."                   :rem 178
4240 GETA$:IFA$=""THEN4240          :rem 183
4400 IFA$="R"THEN3000               :rem 129
4410 GOTO5000                       :rem 199
4500 CT=CT+1:TT=TT+N                :rem 208
4510 IFN>HITHENHI=N                 :rem 193
4520 IFN<LOTHENLO=N                 :rem 212
4530 RETURN                         :rem 172
5000 PRINT"{CLR} PROGRAM ENDED."    :rem 207
```

Power BASIC

See article on page 112.

BEFORE TYPING...

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

Power BASIC: One-Touch Keywords

```
140 IF PEEK(PEEK(56)*256)<>120THENPOKE56,
    PEEK(56)-1:CLR                  :rem 158
150 HI=PEEK(56):BASE=HI*256          :rem 47
160 PRINT"{CLR}PATIENCE..."         :rem 206
170 FOR AD=0 TO 211: READ BY        :rem 153
180 POKE BASE+AD,BY: NEXT AD         :rem 88
190 :                               :rem 212
200 REM RELOCATION ADJUSTMENTS      :rem 184
210 POKE BASE+26,HI: POKE BASE+81,HI
                                     :rem 2
220 POKE BASE+123,HI: POKE BASE+133,HI
                                    :rem 95
230 :                               :rem 207
231 ::IF PEEK(65532)=34 GOTO 240    :rem 135
232 ::POKE BASE+9,72: POKE BASE+48,194
                                    :rem 51
233 ::POKE BASE+52,235: POKE BASE+92,160
                                    :rem 139
234 ::POKE BASE+154,72: POKE BASE+157,224
                                    :rem 193
235 ::POKE BASE+158,234             :rem 230
236 ::                               :rem 15
240 PRINT"{CLR}* ONE-TOUCH KEYWORDS *"
                                    :rem 88
250 PRINT"ON/OFF:{3 SPACES}SYS{RVS}";BASE
                                    :rem 176
260 END                             :rem 111
270 DATA 120,173,143,2,201,32       :rem 127
280 DATA 208,12,169,220,141,143     :rem 239
290 DATA 2,169,235,141,144,2         :rem 94
300 DATA 88,96,169,32,141,143       :rem 155
310 DATA 2,169,0,141,144,2          :rem 237
320 DATA 88,96,165,212,208,117      :rem 206
330 DATA 173,141,2,201,3,176         :rem 83
340 DATA 110,201,0,240,106,169      :rem 175
350 DATA 159,133,245,169,236,133     :rem 49
360 DATA 246,165,215,201,193,144     :rem 40
370 DATA 95,201,219,176,91,56       :rem 160
380 DATA 233,193,174,141,2,224      :rem 194
390 DATA 2,208,3,24,105,26          :rem 245
400 DATA 170,189,159,0,162,0         :rem 92
410 DATA 134,198,170,160,158,132     :rem 40
420 DATA 34,160,192,132,35,160      :rem 187
```

```
430 DATA 0,10,240,16,202,16          :rem 22
440 DATA 12,230,34,208,2,230         :rem 78
450 DATA 35,177,34,16,246,48        :rem 108
460 DATA 241,200,177,34,48,17       :rem 147
470 DATA 8,142,211,0,230,198         :rem 91
480 DATA 166,198,157,119,2,174      :rem 215
490 DATA 211,0,40,208,234,230       :rem 131
500 DATA 198,166,198,41,127,157       :rem 8
510 DATA 119,2,230,198,169,20       :rem 146
520 DATA 141,119,2,76,220,235       :rem 139
530 DATA 76,67,236                  :rem 127
540 :                               :rem 211
550 REM *TOKENS FOR SHIFT KEY       :rem 202
560 :                               :rem 213
570 DATA 153,175,199,135,161,129     :rem 56
580 DATA 141,164,133,137,134,147     :rem 42
590 DATA 202,181,159,151,163,201     :rem 37
600 DATA 196,139,192,149,150,155     :rem 52
610 DATA 191,138                     :rem 20
620 :                               :rem 210
630 REM *TOKENS FOR COMMODORE KEY:rem 240
640 :                               :rem 212
650 DATA 152,176,198,131,128,130     :rem 45
660 DATA 142,169,132,145,140,148     :rem 43
670 DATA 195,187,160,194,166,200     :rem 54
680 DATA 197,167,186,157,165,184     :rem 72
690 DATA 190,158,0                  :rem 121
700 ::                               :rem 11
710 ::REM *CHECKSUM ROUTINE         :rem 147
720 ::                               :rem 13
730 ::FOR AD=0 TO 158 : READ BY      :rem 25
740 ::CHKSUM = CHKSUM + BY : NEXT AD
                                    :rem 166
750 ::IF CHKSUM <> 20347 THEN PRINT "ERRO
    R!"                             :rem 143
```

Scroll 64

See article on page 127 for instructions on use.

```
10 DATA11507,12573,12522,11001
20 A=49152:B=84:C=A+B:FORI=1TO4:D=0:READX
   :FORJ=ATOC:D=D+PEEK(J):NEXT
30 IFD<>XTHENPRINTTAB(19)"ERROR IN BLOCK
   {SPACE}#"I:GOTO50
40 PRINT"BLOCK #"I"IS CORRECT"
50 A=C+1:C=A+B:NEXT:END
60 FORI=1TO4:READA:NEXT:READLO,HI:FORI=LO
   TOHI:READX:POKEI,X:NEXT:END
5000 DATA 49152, 49528
5010 DATA 174,114,193,224,3,144,3,76,117,
     192,188,114,193,140,121,193,174
5020 DATA 118,193,232,202,32,30,193,172,1
     21,193,173,119,193,201,2,208,10
5030 DATA 169,32,72,173,33,208,72,76,50,1
     92,177,90,72,177,92,72,204
5040 DATA 116,193,240,20,200,177,90,72,17
     7,92,136,145,92,104,145,90,200
5050 DATA 204,116,193,208,238,240,18,136,
     177,90,72,177,92,200,145,92,104
5060 DATA 145,90,136,204,115,193,208,238,
     173,119,193,201,0,208,5,104,104
5070 DATA 76,111,192,104,145,92,104,145,9
     0,236,117,193,208,160,96,172,116
5080 DATA 193,200,189,114,193,170,32,30,1
     93,173,120,193,201,2,208,19,136
5090 DATA 169,32,153,122,193,173,33,208,1
     53,162,193,204,115,193,208,239,240
5100 DATA 16,136,177,90,153,122,193,177,9
     2,153,162,193,204,115,193,208,240
5110 DATA 236,117,193,240,37,202,32,30,19
     3,172,116,193,200,136,177,90,72
5120 DATA 177,92,32,48,193,145,92,104,145
     ,90,32,56,193,204,115,193,208
5130 DATA 234,236,117,193,208,221,240,46,
     202,206,118,193,232,32,30,193,172
5140 DATA 116,193,200,136,32,48,193,177,9
     0,72,177,92,32,56,193,145,92
5150 DATA 104,145,90,204,115,193,208,234,
     236,118,193,208,221,238,118,193,232
5160 DATA 32,30,193,173,120,193,201,0,240
     ,20,172,115,193,136,200,185,162
5170 DATA 193,145,92,185,122,193,145,90,2
     04,116,193,208,240,96,189,89,193
5180 DATA 133,91,24,105,212,133,93,189,64
     ,193,133,90,133,92,96,72,152
5190 DATA 24,105,40,168,104,96,72,152,56,
     233,40,168,104,96,0,40,80
5200 DATA 120,160,200,240,24,64,104,144,1
     84,224,8,48,88,128,168,208,248
5210 DATA 32,72,112,152,192,4,4,4,4,4,4,4
     ,5,5,5,5,5
5220 DATA 5,6,6,6,6,6,6,6,7,7,7,7,7,3,0,4
     ,0
5230 DATA 4,1,1
```

Tape Data Files For VIC And 64

See article on page 130.

Program 1:
Files Written From Keyboard

```
200 REM{11 SPACES}* FILES WRITTEN *
                                    :rem 106
210 REM{11 SPACES}* FROM KEYBOARD *
                                     :rem 80
215 CLR                             :rem 121
220 INPUT"{CLR}{5 DOWN}NO. OF ITEMS IN FI
    LE";N                           :rem 114
225 DIMW$(N+1)                      :rem 233
230 INPUT"{DOWN}FILENAME";NAME$:IFNAME$="
    "+1"THEN END                     :rem 44
240 PRINT"{DOWN}ON THE PROMPT,":PRINT"TYP
    E EACH ITEM,":PRINT"FOLLOWED BY {RVS}
    RETURN{OFF}"                     :rem 67
245 OPEN1,1,1,NAME$                  :rem 41
250 FORX=0TON:INPUTW$(X):PRINT#1,W$(X):IF
    W$(X)<>"-1"THEN NEXT              :rem 4
260 CLOSE1:FORX=0TON:PRINTW$(X):NEXT
                                    :rem 122
265 PRINT"HIT ANY KEY"               :rem 36
270 GETA$:IFA$=""THEN 270            :rem 85
280 GOTO200                         :rem 101
290 END                             :rem 114
```

Program 2: Files Written From Data

```
500 REM{10 SPACES}* FILE WRITTEN *:rem 26
510 REM{10 SPACES}*{2 SPACES}FROM DATA
    {3 SPACES}*                      :rem 28
520 REM                             :rem 123
540 DATAW21,DELIGHT,CHAPTER,FARTHER,BUILT
    ,JOYFUL,STORIES                  :rem 11
550 DATABOUGHT,SCARF,FILLED,SAILED,REMAIN
    ,CLOAK,PLACED,DRIVING,FEAST,STRONG,-1
                                    :rem 150
551 DATAW22,FLYING,SOMETIMES,HIGHWAY,SNOW
    ING,CLOSING                      :rem 16
```

```
560 DATABEDTIME,PUSHED,BRUSHES,DREAMING,B
    OOKCASE,PULLED                  :rem 196
570 DATAAIRPLANE,BUYING,SPENDING,SKATED,D
    ECEMBER,-1,+1                   :rem 221
580 READNAME$:R$=CHR$(13)           :rem 161
585 IFNAME$="+1"THENGOTO650          :rem 86
590 OPEN1,1,1,NAME$                  :rem 44
600 READWRD$:PRINT#1,WRD$;R$        :rem 204
620 IFWRD$<>"-1"THEN600              :rem 25
630 CLOSE1                           :rem 64
640 GOTO580                         :rem 112
650 CLOSE1:END                       :rem 83
```

Program 3: File Reader

```
300 REM{11 SPACES}*{4 SPACES}FILE
    {4 SPACES}*                     :rem 235
310 REM{11 SPACES}*{3 SPACES}READER
    {3 SPACES}*                     :rem 127
350 INPUT"{CLR}{8 DOWN}WHAT FILE";NAME$
                                    :rem 91
360 IFNAME$="+1"THEN END             :rem 80
370 OPEN1,1,0,NAME$:N=0:DIMW$(16)   :rem 99
380 INPUT#1,W$(N):N=N+1              :rem 52
390 IFW$(N-1)="-1"THEN410            :rem 70
400 GOTO380                         :rem 104
410 CLOSE1:FORX=0TO(N-1):PRINTW$(X):NEXT
                                    :rem 38
420 PRINT"HIT ANY KEY"               :rem 29
430 GETA$:IFA$=""THEN430             :rem 81
450 RUN350                           :rem 38
460 END                             :rem 113
```

3-D Tic-Tac-Toe

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:
3-D Tic-Tac-Toe—64 Version

```
1 PRINT"{CLR}":POKE53281,0:POKE53280,4
                                    :rem 40
2 DIMDI(28):FORI=1TO27:READX1:X2=X2+X1:DI
  (I+1)=X2:NEXTI:DI(1)=0            :rem 82
5 PRINTTAB(9)"{RED}{3 SPACES}{RVS}£
  {6 SPACES}{OFF}{2 SPACES}{RVS}
  {6 SPACES}§*³"                    :rem 193
6 PRINTTAB(9)"{2 SPACES}{RVS}£{8 SPACES}
  {OFF} {RVS}{7 SPACES}§*³"         :rem 166
7 PRINTTAB(9)" {RVS}£{9 SPACES}{OFF}
  {RVS}{8 SPACES}§*³"               :rem 167
8 PRINTTAB(9)"{BLU}{RVS}£{6 SPACES}{RED}
  {4 SPACES}{OFF} {RVS}{3 SPACES}{BLU}
  {6 SPACES}§*³"                    :rem 2
9 PRINTTAB(9)"{RVS}{8 SPACES}{RED}
  {3 SPACES}{OFF} {RVS}{3 SPACES}{BLU}
  {7 SPACES}§*³"                    :rem 59
10 PRINTTAB(9)"{RVS}{2 SPACES}{RED}{OFF}
   £{2 SPACES}{RVS}£{BLU}{2 SPACES}
   {RED}{3 SPACES}{OFF} {RVS}{3 SPACES}
   {BLU}{2 SPACES}{OFF} {RVS}{RED}
   {3 SPACES}{BLU}{2 SPACES}"       :rem 148
11 PRINTTAB(9)"{4 SPACES}{RVS}{RED}£
   {BLU}{2 SPACES}{RED}{3 SPACES}{OFF}
   {RVS}{3 SPACES}{BLU}{2 SPACES}{OFF}
   {RVS}{RED}{3 SPACES}{BLU}{2 SPACES}"
                                    :rem 72
12 PRINTTAB(9)"{3 SPACES}{RVS}{RED}£
   {2 SPACES}{BLU}{2 SPACES}{RED}
   {3 SPACES}{OFF} {RVS}{3 SPACES}{BLU}
   {2 SPACES}{OFF} {RVS}{RED}{3 SPACES}
   {BLU}{2 SPACES}"                 :rem 73
13 PRINTTAB(9)"{3 SPACES}{RVS}{5 SPACES}
   {RED}{3 SPACES}{OFF} {RVS}{3 SPACES}
   {BLU}{2 SPACES}{OFF} {RVS}{RED}
   {3 SPACES}{BLU}{2 SPACES}"       :rem 102
14 PRINTTAB(9)"{3 SPACES}{RVS}{5 SPACES}
   {RED}{3 SPACES}{OFF} {RVS}{3 SPACES}
   {BLU}{2 SPACES}{OFF} {RVS}{RED}
   {3 SPACES}{BLU}{2 SPACES}"       :rem 103
15 PRINTTAB(9)"{2 SPACES}{RVS}{RED}£
   {3 SPACES}{BLU}{2 SPACES}{RED}
   {3 SPACES}{OFF} {RVS}{3 SPACES}{BLU}
   {2 SPACES}{OFF} {RVS}{RED}{3 SPACES}
   {BLU}{2 SPACES}"                 :rem 76
16 PRINTTAB(9)" {RVS}{RED}£{4 SPACES}
   {BLU}{2 SPACES}{RED}{2 SPACES}{OFF}£
   {SPACE}{RVS}{3 SPACES}{BLU}{2 SPACES}
   {RED}{4 SPACES}{BLU}{2 SPACES}":rem 82
17 PRINTTAB(9)"{RVS}{RED}£{5 SPACES}
   {BLU}{2 SPACES}{RED} {OFF}£{2 SPACES}
   {RVS}{3 SPACES}{BLU}{2 SPACES}{RED}
   {4 SPACES}{BLU}{2 SPACES}"       :rem 83
18 PRINTTAB(9)"{RVS}{2 SPACES}{RED}
   {4 SPACES}{BLU}{2 SPACES}{RED}{OFF}£
   {3 SPACES}§*³{RVS}{2 SPACES}{BLU}
   {2 SPACES}{RED}{4 SPACES}{BLU}
   {2 SPACES}"                      :rem 138
19 PRINTTAB(9)"{RVS}{7 SPACES}{OFF}£
   {5 SPACES}{RED}§*³{RVS} {BLU}
   {7 SPACES}{OFF}£"                :rem 80
20 PRINTTAB(9)"§*³{RVS}{5 SPACES}{OFF}£
   {7 SPACES}{RED}§*³{RVS}{BLU}{6 SPACES}
   {OFF}£"                          :rem 39
25 PRINTTAB(9)"{DOWN}{CYN}T I C{RED} CC
   {CYN}T A C{RED} CC{CYN} T O E":rem 107
30 PRINTTAB(12)"{4 DOWN}§3FIREBUTTON TO
   {SPACE}PLAY"                     :rem 116
35 IF(PEEK(56321)AND16)=16AND(PEEK(56320)
   AND16)=16THEN35                  :rem 172
52 PRINT"{CLR}§6";:POKE53281,0:DIMP(27):
   O=1                              :rem 124
53 PH=54272:CU=32:CC=0:S=54272      :rem 176
54 POKES+13,240                      :rem 60
55 P(1)=1119:P(2)=1122:P(3)=1125:P(4)=120
   1:P(5)=1204:P(6)=1207:P(8)=1286
                                    :rem 198
56 P(7)=1283                         :rem 78
57 P(9)=1289:P(10)=1439:P(11)=1442:P(12)=
   1445:P(13)=1521:P(14)=1524:P(15)=1527
                                    :rem 10
58 P(16)=1603:P(17)=1606:P(18)=1609:P(19)
   =1759:P(20)=1762:P(21)=1765      :rem 10
59 P(22)=1841:P(23)=1844:P(24)=1847:P(25)
   =1923:P(26)=1926:P(27)=1929      :rem 20
60 FORD=1TO2:PRINT"PLAYER #";D;:INPUTPL$(
   D):NEXT:CO(1)=4:CO(2)=5          :rem 161
61 FORV=1TO2:PL$(V)=LEFT$(PL$(V),9):NEXT
                                    :rem 170
63 M(2)=87:M(1)=86:L=1              :rem 141
65 INPUT"HOW MANY ROUNDS";RD         :rem 14
70 U(1)=56321:U(2)=56320            :rem 203
90 PRINT"{CLR}{RED}";:FORV=1TO3     :rem 223
100 PRINTTAB(12)"§9 @³"             :rem 177
```

```
110 PRINTTAB(12)"{RVS}{*3}{OFF}{2 SPACES}M
    {2 SPACES}M{2 SPACES}M"          :rem 216
120 PRINTTAB(12)"{*3}{RVS}{*3}{OFF}{@3 M{@3
     M{@3 M"                         :rem 164
130 PRINTTAB(13)"{*3}{RVS}{*3}{OFF}{2 T3M
    {2 T3M{2 T3M"                    :rem 140
140 PRINTTAB(14)"{*3}{RVS}{*3}{OFF}{@3 M{@3
     M{@3 M"                         :rem 168
150 PRINTTAB(15)"{*3}{RVS}{*3}{OFF}{2 T3M
    {2 T3M{2 T3M"                    :rem 144
160 PRINTTAB(16)"{*3}{RVS}{*3}{OFF}
    {2 SPACES}M{2 SPACES}M{2 SPACES}M"
                                     :rem 192
170 PRINTTAB(17)"{*3}{RVS}{9 SPACES}"
                                     :rem 234
175 ONVGOTO176,177,330               :rem 244
176 PRINT"{BLU}";:NEXT               :rem 66
177 PRINT"{YEL}";:NEXT               :rem 194
330 R=1                              :rem 86
340 POKEP(O),160:POKEP(O)+PH,1       :rem 242
350 PRINT"{HOME}{YEL}ROUND:{6}";R;"{HOME}
    {6 DOWN}{6}":PRINTPL$(L);"'S TURN";
                                     :rem 52
351 FORV=1TO12-LEN(PL$(L))-2:PRINT" ";:NE
    XT                               :rem 34
360 J=15-(PEEK(U(L))AND15)           :rem 165
380 IFJ=8THENPP=1:GOTO410            :rem 46
390 IFJ=4THENPP=-1:GOTO420           :rem 89
400 GOTO459                          :rem 111
410 IFO=27THENO=26:NX=CU:NC=CC:GOTO433
                                     :rem 140
415 NX=PEEK(P(O+1)):NC=PEEK(P(O+1)+PH):GO
    TO430                            :rem 214
420 IFO=1THENO=2:NX=CU:NC=CC:GOTO433
                                     :rem 31
425 NX=PEEK(P(O-1)):NC=PEEK(P(O-1)+PH)
                                     :rem 209
430 POKEP(O),CU:POKEP(O)+PH,CC       :rem 72
433 O=O+PP                           :rem 64
435 CU=NX:CC=NC                      :rem 165
440 POKEP(O),160:POKEP(O)+PH,1       :rem 243
441 POKES+11,17:POKES+8,70:POKES+24,15
                                     :rem 205
442 FORD=1TO20:NEXT:POKES+24,0:POKES+11,1
    6:GOTO460                        :rem 67
459 FORD=1TO15:NEXT                  :rem 189
460 POKEP(O),CU:POKEP(O)+PH,CC:FORD=1TO45
    :NEXT:POKEP(O),160:POKEP(O)+PH,1
                                     :rem 56
470 J=PEEK(U(L))AND16:IFJ=16THEN360
                                     :rem 67
475 IFCU=32THEN480                   :rem 52
476 POKES+11,17:POKES+8,14:POKES+24,15
                                     :rem 211
477 FORD=1TO500:NEXT:POKES+24,0:POKES+11,
    16:GOTO360                       :rem 125
480 POKEP(O),M(L):POKEP(O)+PH,CO(L):CU=M(
    L):CC=CO(L)                      :rem 109
481 POKES+11,17:POKES+8,200:POKES+24,15
                                     :rem 252
482 FORD=1TO250:NEXT:POKES+11,16:POKES+24
    ,0                               :rem 111
485 A=0:B=0:C=0                      :rem 34
490 RESTORE:READX:FOR I=1TODI(O)+13:READX
    ,Y:NEXTI                         :rem 155
495 FORI=DI(O)+1TODI(O+1)            :rem 196
500 READ X,Y:IF(PEEK(P(X))=M(L))AND(PEEK(
    P(Y))=M(L))THEN B=X:C=Y:A=O      :rem 25
510 NEXTI                            :rem 30
740 IFA=0THEN923                     :rem 165
745 PRINT"{HOME}{7 DOWN}";PL$(L);" WINS!"
    ;                                :rem 175
746 FORV=1TO12-LEN(PL$(L))-2:PRINT" ";:NE
    XT                               :rem 42
747 SC(L)=SC(L)+1                    :rem 161
750 T=255:POKES+11,17:POKES+24,15:FORX=1T
    O10                              :rem 112
760 H=INT(16*RND(1))                 :rem 85
770 POKEP(A)+PH,H                    :rem 230
780 POKEP(B)+PH,H                    :rem 232
790 POKEP(C)+PH,H                    :rem 234
800 POKES+8,T:FORD=1TO350:NEXT:T=T-26:NEX
    T:POKES+11,16:POKES+24,0:GOTO920
                                     :rem 68
920 FORO=1TO27:POKEP(O),32:POKEP(O)+PH,0:
    NEXT                             :rem 38
923 IFL=1THENL=2:GOTO930             :rem 224
925 L=1                              :rem 90
930 IFA=0THEN350                     :rem 160
940 CU=32:CC=0:NX=0:NC=0:O=1         :rem 128
945 R=R+1:IFR>RDTHEN950              :rem 155
946 GOTO350                          :rem 116
950 PRINT"{CLR}{7 DOWN}{17 SPACES}"
                                     :rem 121
955 PRINT"{9 RIGHT}{3 DOWN}FINAL SCORE"
                                     :rem 146
957 PRINT"{9 RIGHT}{YEL}{11 Y3}{GRN}"
                                     :rem 20
960 PRINT"{2 RIGHT}"PL$(1);":";SC(1)
                                     :rem 248
970 PRINT"{DOWN}{2 RIGHT}"PL$(2);":";SC(2
    )                                :rem 12
980 PRINT"{DOWN}{2 RIGHT}{3}FIREBUTTON TO
     PLAY AGAIN"                     :rem 142
990 IF(PEEK(U(1))AND16)=16THENIF(PEEK(U(2
    ))AND16)=16THEN990               :rem 185
1000 POKE198,0:RUN                   :rem 29
1080 DATA 7,4,7,4,5,4,7,4,7,4,5,4,5,12,5,
     4,5,4,7,4,7,4,5,4,7,4,7         :rem 36
1100 DATA 2,3,4,7,5,9,10,19,13,25,11,21,1
     4,27,1,3,5,8,11,20              :rem 28
1105 DATA 14,26,2,1,6,9,5,7,12,21,15,27,1
     4,25,11,19,5,6                  :rem 112
1110 DATA 14,24,13,22,1,7,1,9,2,8,3,7,4,6
     ,14,23,15,24                    :rem 2
1115 DATA 3,9,14,22,4,5,1,4,3,5,8,9,16,25
     ,13,19,17,27,14,21              :rem 55
1120 DATA 7,9,2,5,17,26,14,20,1,5,3,6,7,8
     ,18,27,15,21,17,25,14,19        :rem 95
1125 DATA 11,12,13,16,14,18,1,19,10,12,14
     ,17,2,20,3,19,1,21              :rem 30
1130 DATA 14,16,15,18,10,11,3,21,10,16,14
     ,15                             :rem 74
1135 DATA 4,22,1,25,7,19,10,18,11,17,12,1
     6,13,15,2,26                    :rem 11
1140 DATA 1,27,3,25,4,24,6,22,7,21,8,20,9
     ,19,12,18                       :rem 127
1145 DATA 13,14,6,24,3,27,9,21,10,13,18,1
     7,12,14,7,25                    :rem 18
1150 DATA 11,14,16,18,8,26,9,25,7,27,12,1
     5,10,14,16,17                   :rem 74
1155 DATA 9,27,20,21,22,25,23,27,1,10,9,1
     4,7,13,3,11                     :rem 218
1160 DATA 23,26,2,11,8,14,19,21,19,20,27,
     24,23,25                        :rem 80
1165 DATA 12,3,9,15,7,14,1,11,19,25,23,24
     ,4,13,6,14                      :rem 174
1170 DATA 19,27,20,26,21,25,22,24,5,14,22
     ,23,21,27                       :rem 127
1175 DATA 6,15,4,14,23,21,26,27,19,22,7,1
     6,1,13,9,17                     :rem 236
```

```
1180 DATA 3,14,20,23,25,27,8,17,2,14,19,2
     3,21,24,25,26                    :rem 69
1185 DATA 9,18,3,15,7,17,1,14         :rem 141
```

Program 2:

3-D Tic-Tac-Toe—VIC Version

Memory expansion (any amount) required.

```
100 PRINT"{CLR}":POKE36879,15:DIMP(27)
                                      :rem 31
110 DIMDI(28):FORI=1TO27:READX1:X2=X2+X1:
    DI(I+1)=X2:NEXTI:DI(1)=0          :rem 178
120 DD=37154:P1=37151:P2=37152        :rem 80
130 PRINT"{RED}{3 SPACES}{RVS}£
    {5 SPACES}{OFF}{2 SPACES}{RVS}
    {5 SPACES}£*£"                    :rem 191
140 PRINT"{2 SPACES}{RVS}£{7 SPACES}
    {OFF} {RVS}{6 SPACES}£*£"         :rem 164
150 PRINT" {RVS}£{8 SPACES}{OFF} {RVS}
    {7 SPACES}£*£"                    :rem 165
160 PRINT"{BLU}{RVS}£{6 SPACES}{RED}
    {3 SPACES}{OFF} {RVS}{2 SPACES}{BLU}
    {6 SPACES}£*£"                    :rem 0
170 PRINT"{RVS}{8 SPACES}{RED}{2 SPACES}
    {OFF} {RVS}{2 SPACES}{BLU}{7 SPACES}
    £*£"                              :rem 57
180 PRINT"{RVS}{2 SPACES}{RED}{OFF}£
    {2 SPACES}{RVS}£{BLU}{2 SPACES}{RED}
    {2 SPACES}{OFF} {RVS}{2 SPACES}{BLU}
    {2 SPACES}{OFF}{2 SPACES}{RVS}{RED}
    {2 SPACES}{BLU}{2 SPACES}"        :rem 107
190 PRINT"{4 SPACES}{RVS}{RED}£ {BLU}
    {2 SPACES}{RED}{2 SPACES}{OFF} {RVS}
    {2 SPACES}{BLU}{2 SPACES}{OFF}
    {2 SPACES}{RVS}{RED}{2 SPACES}{BLU}
    {2 SPACES}"                       :rem 31
200 PRINT"{3 SPACES}{RVS}{RED}£
    {2 SPACES}{BLU}{2 SPACES}{RED}
    {2 SPACES}{OFF} {RVS}{2 SPACES}{BLU}
    {2 SPACES}{OFF}{2 SPACES}{RVS}{RED}
    {2 SPACES}{BLU}{2 SPACES}"        :rem 23
210 PRINT"{3 SPACES}{RVS}{5 SPACES}{RED}
    {2 SPACES}{OFF} {RVS}{2 SPACES}{BLU}
    {2 SPACES}{OFF}{2 SPACES}{RVS}{RED}
    {2 SPACES}{BLU}{2 SPACES}"        :rem 52
220 PRINT"{3 SPACES}{RVS}{5 SPACES}{RED}
    {2 SPACES}{OFF} {RVS}{2 SPACES}{BLU}
    {2 SPACES}{OFF}{2 SPACES}{RVS}{RED}
    {2 SPACES}{BLU}{2 SPACES}"        :rem 53
230 PRINT"{2 SPACES}{RVS}{RED}£
    {3 SPACES}{BLU}{2 SPACES}{RED}
    {2 SPACES}{OFF} {RVS}{2 SPACES}{BLU}
    {2 SPACES}{OFF}{2 SPACES}{RVS}{RED}
    {2 SPACES}{BLU}{2 SPACES}"        :rem 26
240 PRINT" {RVS}{RED}£{4 SPACES}{BLU}
    {2 SPACES}{RED}{2 SPACES}{OFF} {RVS}
    {2 SPACES}{BLU}{2 SPACES}{RED}
    {4 SPACES}{BLU}{2 SPACES}"        :rem 119
250 PRINT"{RVS}{RED}£{5 SPACES}{BLU}
    {2 SPACES}{RED} {OFF}£ {RVS}
    {2 SPACES}{BLU}{2 SPACES}{RED}
    {4 SPACES}{BLU}{2 SPACES}"        :rem 33
260 PRINT"{RVS}{2 SPACES}{RED}{4 SPACES}
    {BLU}{2 SPACES}{RED}{OFF}£{2 SPACES}
    {RVS}{2 SPACES}{BLU}{2 SPACES}{RED}
    {4 SPACES}{BLU}{2 SPACES}"        :rem 121
270 PRINT"{RVS}{7 SPACES}{OFF}£
    {3 SPACES}{RED}£*£{RVS} {BLU}
    {7 SPACES}{OFF}£"                 :rem 30
280 PRINT"£*£{RVS}{5 SPACES}{OFF}£
    {5 SPACES}{RED}£*£{RVS}{BLU}
    {6 SPACES}{OFF}£"                 :rem 254
290 PRINT"{DOWN}{CYN}T I C{RED} C {CYN}T
    {SPACE}A C{RED} C{CYN} T O E":rem 184
300 PRINTTAB(2)"{DOWN}£3£FIREBUTTON TO PL
    AY"                               :rem 64
310 POKEDD,255:P=PEEK(P1):IF -((PAND32)=0
    )<>1 THEN 310                     :rem 204
320 PRINT"{CLR}£6£";:POKE36879,10:O=1
                                      :rem 203
325 H=256*PEEK(648):PH=30720:IFPEEK(648)=
    16THENPH=33792                    :rem 154
330 CU=32:CC=0:S=36874:VO=36878        :rem 255
340 POKES+13,240                      :rem 106
350 P(1)=31+H:P(2)=34+H:P(3)=37+H:P(4)=77
    +H:P(5)=80+H:P(6)=83+H:P(7)=123+H
                                      :rem 163
360 P(8)=126+H:P(9)=129+H:P(10)=207+H
                                      :rem 160
370 P(11)=210+H:P(12)=213+H:P(13)=253+H:P
    (14)=256+H:P(15)=259+H            :rem 11
380 P(16)=299+H:P(17)=302+H:P(18)=305+H:P
    (19)=383+H:P(20)=386+H:P(21)=389+H
                                      :rem 190
390 P(22)=429+H:P(23)=432+H:P(24)=435+H:P
    (25)=475+H:P(26)=478+H:P(27)=481+H
                                      :rem 191
400 FORD=1TO2:PRINT"PLAYER #";D;:INPUTPL$
    (D):NEXT:CO(1)=4:CO(2)=5          :rem 207
410 FORV=1TO2:PL$(V)=LEFT$(PL$(V),8):NEXT
                                      :rem 215
420 M(2)=87:M(1)=86:L=1               :rem 186
430 INPUT"HOW MANY ROUNDS";RD         :rem 58
450 PRINT"{CLR}{RED}";:FORV=1TO3      :rem 15
460 PRINT"{7 SPACES}£9 @£"            :rem 47
470 PRINTTAB(6)"{RVS}£*£{OFF}{2 SPACES}M
    {2 SPACES}M{2 SPACES}M"           :rem 180
480 PRINTTAB(6)"£*£{RVS}£*£{OFF}£@£ M£@£
    {SPACE}M£@£ M"                    :rem 128
490 PRINTTAB(7)"£*£{RVS}£*£{OFF}£2 T£M
    £2 T£M£2 T£M"                     :rem 104
500 PRINTTAB(8)"£*£{RVS}£*£{OFF}£@£ M£@£
    {SPACE}M£@£ M"                    :rem 123
510 PRINTTAB(9)"£*£{RVS}£*£{OFF}£2 T£M
    £2 T£M£2 T£M"                     :rem 99
520 PRINTTAB(10)"£*£{RVS}£*£{OFF}
    {2 SPACES}M{2 SPACES}M{2 SPACES}M"
                                      :rem 186
530 PRINTTAB(11)"£*£{RVS}{9 SPACES}{OFF}"
    ;                                 :rem 177
540 ONVGOTO550,560,570                :rem 238
550 PRINT"{BLU}";:NEXT                :rem 62
560 PRINT"{GRN}";:NEXT                :rem 62
570 R=1                               :rem 92
580 POKEP(O),160:POKEP(O)+PH,2        :rem 249
590 PRINT"{HOME}{4 DOWN}{YEL}ROUND:£6£";R
                                      :rem 76
600 PRINT"{HOME}{6 DOWN}";:FORV=1TO11:PRI
    NT" ";:NEXTV                      :rem 26
605 PRINT"{HOME}{5 DOWN}":PRINTPL$(L);"'S
    ":PRINT"TURN"                     :rem 69
610 POKEDD,127:P=PEEK(P2)AND128:J1=-(P=0)
                                      :rem 110
620 POKEDD,255:P=PEEK(P1):J2=-((PAND16)=0
    )                                 :rem 142
630 IFJ1=1THENPP=1:GOTO660            :rem 93
640 IFJ2=1THENPP=-1:GOTO680           :rem 142
650 GOTO760                           :rem 113
660 IFO=27THENO=26:NX=CU:NC=CC:GOTO710
                                      :rem 145
670 NX=PEEK(P(O+1)):NC=PEEK(P(O+1)+PH):GO
    TO700                             :rem 217
680 IFO=1THENO=2:NX=CU:NC=CC:GOTO710
                                      :rem 37
```

```
690 NX=PEEK(P(O-1)):NC=PEEK(P(O-1)+PH)
                                    :rem 213
700 POKEP(O),CU:POKEP(O)+PH,CC      :rem 72
710 O=O+PP                          :rem 62
720 CU=NX:CC=NC                     :rem 162
730 POKEP(O),160:POKEP(O)+PH,1      :rem 245
740 POKEVO,15:POKES,170             :rem 129
750 FORD=1TO20:NEXT:POKEVO,0:GOTO770
                                    :rem 46
760 FORD=1TO15:NEXT                 :rem 184
770 POKEP(O),CU:POKEP(O)+PH,CC:FORD=1TO45
    :NEXT:POKEP(O),160:POKEP(O)+PH,1
                                    :rem 60
780 POKEDD,255:P=PEEK(P1):J= -((PAND32)=0
    ):IFJ<>1THEN610                 :rem 229
790 IFCU=32THEN820                  :rem 50
800 POKEVO,15:POKES,160             :rem 125
810 FORD=1TO500:NEXT:POKEVO,0:GOTO610
                                    :rem 87
820 POKEP(O),M(L):POKEP(O)+PH,CO(L):CU=M(
    L):CC=CO(L)                     :rem 107
830 POKEVO,15:POKES,200             :rem 123
840 FORD=1TO250:NEXT:POKEVO,0       :rem 82
850 A=0:B=0:C=0                     :rem 30
860 RESTORE:READX:FOR I=1TODI(O)+13:READX
    ,Y:NEXTI                        :rem 156
870 FORI=DI(O)+1TODI(O+1)           :rem 193
880 READ X,Y:IF(PEEK(P(X))=M(L))AND(PEEK(
    P(Y))=M(L))THEN B=X:C=Y:A=O     :rem 36
890 NEXTI                           :rem 41
900 IFA=0THEN1010                   :rem 199
910 PRINT"{HOME}{14 DOWN}{10 SPACES}"
                                    :rem 108
920 PRINT"{HOME}{14 DOWN}";PL$(L):PRINT"W
    INS!";                          :rem 173
930 SC(L)=SC(L)+1                   :rem 155
940 T=255:POKEVO,15:POKES,133::FORX=1TO10
                                    :rem 14
950 H=INT(16*RND(1))                :rem 86
960 POKEP(A)+PH,H                   :rem 231
970 POKEP(B)+PH,H                   :rem 233
980 POKEP(C)+PH,H                   :rem 235
990 POKES,T:FORD=1TO350:NEXT:T=T-26:NEXT:
    POKES,160:POKEVO,0:GOTO1000     :rem 117
1000 FORO=1TO27:POKEP(O),32:POKEP(O)+PH,0
     :NEXT                          :rem 76
1010 IFL=1THENL=2:GOTO1030          :rem 44
1020 L=1                            :rem 125
1030 IFA=0THEN590                   :rem 206
1040 CU=32:CC=0:NX=0:NC=0:O=1       :rem 168
1050 R=R+1:IFR>RDTHEN1070           :rem 233
1060 PRINT"{HOME}{14 DOWN}{11 SPACES}":PR
     INT"{5 SPACES}":GOTO590        :rem 181
1070 PRINT"{CLR}{7 DOWN}{17 SPACES}"
                                    :rem 163
1080 PRINT"{3 DOWN}FINAL SCORE"     :rem 179
1090 PRINT"{YEL}{11 Y}{GRN}"        :rem 52
1100 PRINTPL$(1);":";SC(1)          :rem 157
1110 PRINTPL$(2);":";SC(2)          :rem 160
1120 PRINT"{DOWN}FIREBUTTON TO PLAY
     {4 SPACES}AGAIN,(E) TO END"    :rem 29
1125 POKEDD,255:P=PEEK(P1)          :rem 6
1128 GET XX$:IF XX$="E" THEN END :rem 233
1129 IF -((PAND32)=0)<>1 THEN 1125
                                    :rem 195
1130 POKE198,0:RUN                  :rem 33
1140 DATA 7,4,7,4,5,4,7,4,7,4,5,4,5,12,5,
     4,5,4,7,4,7,4,5,4,7,4,7        :rem 33
1150 DATA 2,3,4,7,5,9,10,19,13,25,11,21,1
     4,27,1,3,5,8,11,20             :rem 33
1160 DATA 14,26,2,1,6,9,5,7,12,21,15,27,1
     4,25,11,19,5,6                 :rem 113
1170 DATA 14,24,13,22,1,7,1,9,2,8,3,7,4,6
     ,14,23,15,24                   :rem 8
1180 DATA 3,9,14,22,4,5,1,4,3,5,8,9,16,25
     ,13,19,17,27,14,21             :rem 57
1190 DATA 7,9,2,5,17,26,14,20,1,5,3,6,7,8
     ,18,27,15,21,17,25,14,19       :rem 102
1200 DATA 11,12,13,16,14,18,1,19,10,12,14
     ,17,2,20,3,19,1,21             :rem 24
1210 DATA 14,16,15,18,10,11,3,21,10,16,14
     ,15                            :rem 73
1220 DATA 4,22,1,25,7,19,10,18,11,17,12,1
     6,13,15,2,26                   :rem 6
1230 DATA 1,27,3,25,4,24,6,22,7,21,8,20,9
     ,19,12,18                      :rem 127
1240 DATA 13,14,6,24,3,27,9,21,10,13,18,1
     7,12,14,7,25                   :rem 14
1250 DATA 11,14,16,18,8,26,9,25,7,27,12,1
     5,10,14,16,17                  :rem 75
1260 DATA 9,27,20,21,22,25,23,27,1,10,9,1
     4,7,13,3,11                    :rem 215
1270 DATA 23,26,2,11,8,14,19,21,19,20,27,
     24,23,25                       :rem 82
1280 DATA 12,3,9,15,7,14,1,11,19,25,23,24
     ,4,13,6,14                     :rem 172
1290 DATA 19,27,20,26,21,25,22,24,5,14,22
     ,23,21,27                      :rem 130
1300 DATA 6,15,4,14,23,21,26,27,19,22,7,1
     6,1,13,9,17                    :rem 226
1310 DATA 3,14,20,23,25,27,8,17,2,14,19,2
     3,21,24,25,26                  :rem 64
1320 DATA 9,18,3,15,7,17,1,14       :rem 132
```

Shape Match For VIC And 64

VIC users must have at least 8K memory expansion.
See article on page 84.

```
50 REM **{11 SPACES}INITIALIZE{7 SPACES}*
   *                                :rem 227
52 DIM CL(20,40)                    :rem 19
53 PRINT"{CLR}{BLU}":SYS65517:WI=PEEK(781
   ):IFWI=22THENWI=0:POKE36879,238:SS=2:P
   H=.5                             :rem 163
54 IF WI=0THENAA$="VIC":BB$="VIC":GOTO60
                                    :rem 91
55 POKE53280,1:POKE53281,15:SS=12:WI=8:PH
   =1                               :rem 5
57 AA$="64{DOWN}{22 LEFT}":BB$="64"
                                    :rem 89
60 IF WI=0 THEN CL(4,18)=1:CL(4,31)=2:CL(
   10,18)=3:CL(10,31)=4:GOTO80      :rem 108
63 FORA=6TO12STEP6:FORB=17TO33STEP16:S=64
   6:POKES,SS                       :rem 40
70 T=T+1:CL(A,B)=T:NEXTB,A          :rem 42
80 PRINT"{6 DOWN}";:PRINTTAB(WI+5)"{BLK}S
   HAPE MATCH"                      :rem 123
190 GOSUB3000:REM **{2 SPACES}INSTRUCTION
    S{3 SPACES}**                   :rem 88
200 GOSUB1000:REM ** PRINT BOARD **
                                    :rem 142
210 FORT=1TO1000:NEXT               :rem 25
300 REM ******************************
                                    :rem 141
302 REM **{8 SPACES}SELECT SHAPES
    {6 SPACES}**                    :rem 165
305 REM ******************************
                                    :rem 146
```

```
306 K=RND(-TI)                          :rem 32
310 K=INT(RND(0)*4)+1:IFK=KKTHEN310
                                        :rem 33
320 REM                                 :rem 121
340 FORT=1TO3:GOSUB2000:BO$=SH$(T):A=4+T:
    B=18:PRINT"{HOME}":GOSUB1170:NEXT T
                                        :rem 29
345 GOSUB1180:REM **{6 SPACES}DELAY
    {6 SPACES}**                        :rem 27
348 IFWI=0THENFORT=1TO3:GOSUB2100:BO$=SH$
    (T):A=4+T:B=32:PRINT"{HOME}"  :rem 157
349 IFWI=0THENGOSUB1170:NEXTT:GOTO355
                                        :rem 145
350 FORT=1TO3:GOSUB2100:BO$=SH$(T):A=4+T:
    B=28:PRINT"{HOME}":GOSUB1170:NEXT T
                                        :rem 32
355 GOSUB1180                           :rem 231
360 FORT=1TO3:GOSUB2200:BO$=SH$(T):A=10+T
    :B=18:PRINT"{HOME}":GOSUB1170:NEXT T
                                        :rem 78
365 GOSUB1180                           :rem 232
370 FORT=1TO3:GOSUB2300:BO$=SH$(T):A=10+T
    :B=28:PRINT"{HOME}":GOSUB1170:NEXT T
                                        :rem 81
375 GOSUB1180                           :rem 233
380 FORT=1TO3                           :rem 26
383 ONKGOSUB2000,2100,2200,2300   :rem 152
385 BO$=SH$(T):A= 8+T:B=3:PRINT"{HOME}":G
    OSUB1170:NEXT T:KK=K                :rem 23
500 REM                                 :rem 121
550 REM END                             :rem 85
600 REM ********************************
                                        :rem 144
602 REM **{10 SPACES}MOVE CURSOR
    {6 SPACES}**                        :rem 57
604 REM ********************************
                                        :rem 148
605 PRINT"{HOME}":C=1:FORNT=1TO10:rem 159
610 FORT=1TO15:BO$=CHR$(63):IFT/2=INT(T/2
    )THEN BO$=CHR$(18)+CHR$(63)+CHR$(146)
                                        :rem 182
613 IF WI=0THEN ON C GOTO 660,673,683,687
                                        :rem 132
615 ONC GOTO670,675,680,685             :rem 185
620 GOSUB1170:PRINT"{HOME}":FORTT=1TO 75:
    {8 SPACES}NEXTTT:GOTO700      :rem 112
625 NEXT T                              :rem 48
630 BO$=CHR$(32):GOSUB1170:PRINT"{HOME}":
    {8 SPACES}FORTT=1TO 75:NEXTTT:C=C+1
                                        :rem 163
640 IFC>4THENC=1                        :rem 190
650 NEXTNT:GOTO600:REM **{2 SPACES}REPEAT
    {2 SPACES}**                        :rem 12
660 A=4:B=18:L=1:GOTO620                :rem 111
670 A=6:B=17:L=1:GOTO620                :rem 113
673 A=10:B=18:L=2:GOTO620               :rem 161
675 A=12:B=17:L=2:GOTO620               :rem 164
680 A=6:B=33:L=3:GOTO620                :rem 114
683 A=4:B=31:L=3:GOTO620                :rem 113
685 A=12:B=33:L=4:GOTO620               :rem 165
687 A=10:B=31:L=4:GOTO620               :rem 163
700 REM ********************************
                                        :rem 145
702 REM **{5 SPACES}CHECK FOR RESPONSE
    {4 SPACES}**                        :rem 217
704 REM ********************************
                                        :rem 149
710 GETA$:IFA$=""THEN 625               :rem 88
715 IF A$="Q"THEN PRINT"{CLR}":END :rem 7
720 IFCL(A,B)=K THEN800:REM CORRECT
                                        :rem 54
740 GOTO850: REM **{2 SPACES}WRONG
    {4 SPACES}**                        :rem 196
800 REM ********************************
                                        :rem 146
802 REM **{9 SPACES}CORRECT ANSWER
    {4 SPACES}**                        :rem 8
804 REM ********************************
                                        :rem 150
810 BO$="{RVS}{WHT}YES! THAT'S RIGHT!":A=
    18:GOSUB1174:PRINT"{HOME}"          :rem 8
820 BO$="{5}{RVS}{BLK} PRESS ANY KEY
    {WHT}{OFF}":A=20:GOSUB1174:PRINT"
    {HOME}"                             :rem 190
825 GETA$:IFA$=""THEN825                :rem 97
830 FORA=1TO23:BO$="{40 SPACES}":B=0:GOSU
    B1170                               :rem 21
835 PRINT"{HOME}":NEXT                  :rem 253
840 GOTO200:REM **{2 SPACES}RE-PRINT BOAR
    D{2 SPACES}**                       :rem 230
850 REM ********************************
                                        :rem 151
852 REM **{11 SPACES}WRONG ANSWER
    {4 SPACES}**                        :rem 136
854 REM ********************************
                                        :rem 155
855 BO$=CHR$(32):GOSUB1170:PRINT"{HOME}"
                                        :rem 236
860 BO$="{RVS}{5}{BLK}SORRY...TRY AGAIN
    ...{WHT}{OFF}":{7 SPACES}A=18:B=10:GO
    SUB1174:PRINT"{HOME}"         :rem 151
865 FORW=1TO1000:NEXT                   :rem 44
870 BO$="{22 SPACES}":A=18:{6 SPACES}B=10
    :GOSUB1174:PRINT"{HOME}"      :rem 181
875 C=C+1:IFC>4THENC=1                  :rem 33
880 GOTO 625                            :rem 118
999 REM{2 SPACES}************************
    ******                              :rem 123
1000 PRINT"{CLR}{BLU}"                  :rem 68
1015 IFWI=0THENBO$=" {RVS}{A}*****{R}****
    *{S}":A=3:B=15:GOSUB1170:PRINT"{HOME}
    ":GOTO1030                          :rem 76
1020 A=3:B=15:BO$="{RVS}{A}*********{R}**
    *******{S}{OFF}":GOSUB1170:PRINT"{HO
    ME}"                                :rem 216
1025 REM                                :rem 172
1030 FORT=1TO5:A=3+T:B=15               :rem 206
1035 IF WI=0THENBO$=" {RVS}-{OFF}
    {5 SPACES}{RVS}-{OFF}{5 SPACES}{RVS}
    -{OFF}":GOSUB1170:PRINT"{HOME}":GOTO
    1045                                :rem 43
1040 BO$="{RVS}B{OFF}{9 SPACES}{RVS}B
    {OFF}{9 SPACES}{RVS}B{OFF}":GOSUB117
    0:PRINT"{HOME}"                     :rem 206
1045 NEXT T                             :rem 93
1047 IFWI=0THENA=9:B=15:BO$=" {RVS}{Q}***
    **+*****{W}{OFF}":GOSUB1170:PRINT"{H
    OME}":GOTO1060                      :rem 21
1050 A=9:B=15:BO$="{RVS}{Q}*********+****
    *****{W}{OFF}":GOSUB1170:PRINT"{HOME
    }"                                  :rem 10
1060 FORT=1TO5:A=9+T:B=15               :rem 215
1065 IFWI=0THENBO$=" {RVS}-{OFF}
    {5 SPACES}{RVS}-{OFF}{5 SPACES}{RVS}
    -{OFF}":GOSUB1170:PRINT"{HOME}":GOTO
    1073                                :rem 47
1070 BO$="{RVS}B{OFF}{9 SPACES}{RVS}B
    {OFF}{9 SPACES}{RVS}B{OFF}":GOSUB117
    0:PRINT"{HOME}"                     :rem 209
1073 NEXTT                              :rem 94
```

```
1078 A=15:B=15:IFWI=0THENBO$=" {RVS}{Z}**
     ***{E}*****{X}{OFF}":GOSUB1170:PRINT"
     {HOME}":GOTO1090                :rem 43
1080 BO$="{RVS}{Z}*********{E}*********
     {X}{OFF}":GOSUB1170:PRINT"{HOME}"
                                     :rem 223
1090 BO$=T$:A=20:B=13:GOSUB1170:PRINT"
     {HOME}"                         :rem 12
1093 PRINTTAB(WI)"{2 SPACES}TYPE (Q) TO Q
     UIT.{HOME}"                     :rem 113
1095 RETURN                          :rem 175
1100 REM ************************+***
                                     :rem 147
1110 REM **{11 SPACES}PRINT{10 SPACES}**
                                     :rem 220
1111 REM ****************************
                                     :rem 148
1170 FORI=1TOA:PRINT"{DOWN}";:NEXT:PRINTT
     AB(B*PH)BO$:RETURN              :rem 221
1174 FORI=1TOA:PRINT"{DOWN}";:NEXT:PRINTT
     AB(WI)BO$:RETURN                :rem 125
1175 REM                             :rem 178
1180 FORX=1TO500:NEXT:RETURN          :rem 66
2000 REM                             :rem 166
2015 SH$(1)="{RED} {RVS}{3 U}" :rem 224
2020 SH$(2)=" {RVS}{3 SPACES}"       :rem 153
2025 SH$(3)=" {RVS}{3 SPACES}{WHT}":RETUR
     N                               :rem 190
2100 REM                             :rem 167
2115 SH$(1)="{PUR}{RVS}{RIGHT} {RIGHT}"
                                     :rem 115
2120 SH$(2)="{RVS}{3 SPACES}"        :rem 154
2125 SH$(3)="{RVS}{RIGHT} {RIGHT}{WHT}":R
     ETURN                           :rem 249
2200 REM                             :rem 168
2215 SH$(1)="{BLK}{RVS}{3 RIGHT}£"
                                      :rem 46
2220 SH$(2)="{RVS}{2 RIGHT}£ "       :rem 126
2225 SH$(3)="{RVS}{RIGHT}£{2 SPACES}
     {WHT}":RETURN                   :rem 134
2300 REM                             :rem 169
2315 SH$(1)="{BLU}{RVS}{2 RIGHT}£{*}
     {RIGHT}"                        :rem 157
2320 SH$(2)="{RVS}{RIGHT}£{2 SPACES}
     {*}"                            :rem 65
2325 IFWI=8THENSH$(3)="{RVS}£{4 SPACES}
     {*}":RETURN                     :rem 23
2330 SH$(3)="":RETURN                :rem 166
3000 PRINT"{4 DOWN}";:PRINTTAB(WI)"
     {2 SPACES}INSTRUCTIONS (Y/N) ?"
                                       :rem 1
3020 GETA$:IFA$=""THEN3020           :rem 173
3030 IFA$="N"THEN3200                :rem 125
3040 IFA$="Y"THENPRINT"{CLR}":GOTO 3100
                                      :rem 95
3050 GOTO3020                        :rem 198
3100 PRINTTAB(WI)"IN SHAPE MATCH,THE "AA$
     ;:PRINTTAB(WI)"WILL PRINT A SHAPE ON
     "                               :rem 167
3110 PRINTTAB(WI)"THE LEFT SIDE OF THE":P
     RINTTAB(WI)"SCREEN. THE STUDENT"
                                     :rem 206
3115 PRINTTAB(WI)"WILL FIND 4 DIFFERENT":
     PRINTTAB(WI)"SHAPES ON THE RIGHT"
                                      :rem 26
3120 PRINTTAB(WI)"SIDE OF THE SCREEN,":PR
     INTTAB(WI)"ONE OF WHICH WILL"
                                     :rem 219
3130 PRINTTAB(WI)"MATCH THE SHAPE ON":PRI
     NTTAB(WI)"THE LEFT."            :rem 201
3140 PRINTTAB(WI)"WHEN THE {RVS}?{OFF} IS
     ":PRINTTAB(WI)"NEXT TO THE SHAPE HE"
                                     :rem 135
3150 PRINTTAB(WI)"OR SHE THINKS IS":PRINT
     TAB(WI)"CORRECT, THE STUDENT"
                                     :rem 105
3155 PRINTTAB(WI)"SHOULD PRESS ANY KEY."
                                     :rem 194
3156 PRINTTAB(WI)"THE ";BB$;" WILL EVALUA
     TE"                             :rem 58
3160 PRINTTAB(WI)"THE ANSWER AND LET":PRI
     NTTAB(WI)"THE STUDENT KNOW HOW"
                                     :rem 212
3165 PRINTTAB(WI)"HE DID. TYPE {RVS}Q
     {OFF} TO":PRINTTAB(WI)"END THE PROGR
     AM.{DOWN}"                      :rem 176
3190 PRINTTAB(WI)"PRESS ANY KEY TO PLAY";
                                     :rem 216
3195 GETA$:IFA$=""THEN3195           :rem 199
3196 PRINT"{CLR}{WHT}"CHR$(142)       :rem 37
3200 RETURN                          :rem 165
```

The Frantic Fisherman

See special instructions in article on page 58.

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:

The Frantic Fisherman—Redefined Characters, VIC Version

```
1 PRINT"{CLR}{3 DOWN}{2 SPACES}FRANTIC
  {2 SPACES}FISHERMAN"               :rem 108
2 PRINT"{4 DOWN} LOADING CHARACTERS.."
                                     :rem 129
10 FORT=7168TO7168+62*8-1:READA:POKET,A:N
   EXT                               :rem 181
20 DATA14,62,254,62,14,2,6,6        :rem 131
30 DATA14,14,14,22,22,22,54,54      :rem 222
40 DATA118,118,246,254,246,246,246,246
                                     :rem 141
50 DATA0,0,0,1,1,3,7,7               :rem 70
60 DATA246,246,246,254,246,246,246,246
                                     :rem 147
70 DATA15,31,63,127,255,255,255,255
                                     :rem 242
80 DATA0,0,0,0,0,3,7,31             :rem 116
90 DATA246,246,6,254,254,6,6,15     :rem 45
100 DATA255,255,8,255,255,0,0,0     :rem 23
110 DATA127,127,64,255,255,0,0,0    :rem 70
120 DATA255,255,127,127,63,31,15,7
                                     :rem 183
130 DATA170,255,85,0,255,85,0,170:rem 132
140 DATA255,255,254,254,252,248,240,224
                                     :rem 182
150 DATA6,12,24,240,192,0,0,0       :rem 165
160 DATA0,0,0,192,240,24,12,6       :rem 166
170 DATA96,48,24,15,3,0,0,0          :rem 80
180 DATA0,0,0,3,15,24,48,96          :rem 81
190 DATA40,40,170,60,28,252,12,60:rem 122
200 DATA12,40,40,248,248,40,20,40:rem 113
210 DATA40,40,170,60,52,63,48,60    :rem 73
```

```
220 DATA48,40,40,47,47,40,20,40    :rem 22
230 DATA32,48,190,187,252,255,190,128
                                   :rem 88
240 DATA4,12,125,221,63,255,125,1:rem 118
250 DATA0,0,0,0,0,0,7,195          :rem 219
260 DATA0,0,0,0,1,13,109,255       :rem 113
270 DATA0,0,0,0,128,224,248,248    :rem 22
280 DATA0,0,0,0,1,15,15,15         :rem 11
290 DATA7,3,15,63,255,255,255,255:rem 146
300 DATA199,255,255,255,255,255,255,255
                                   :rem 200
310 DATA255,255,255,255,255,255,255,255
                                   :rem 194
320 DATA0,192,240,240,224,248,252,248
                                   :rem 72
330 DATA255,7,31,127,3,24,31,31    :rem 22
340 DATA0,0,0,0,0,0,0,0            :rem 101
350 DATA255,255,255,255,255,127,255,255,0
    ,0,0,0,0,0,0,0                 :rem 164
360 DATA255,255,255,255,252,255,255,224
                                   :rem 192
370 DATA255,192,240,252,0,0,128,0:rem 122
380 DATA0,0,63,0,0,0,0,0           :rem 162
390 DATA3,3,193,1,1,1,0,0          :rem 224
400 DATA0,0,0,0,0,3,12,0           :rem 152
410 DATA0,0,0,48,192,0,0,0          :rem 11
420 DATA127,127,63,63,31,15,7,3    :rem 30
430 DATA1,2,4,8,16,32,64,128       :rem 133
440 DATA0,0,0,0,4,4,8,8            :rem 126
450 DATA255,127,31,7,0,0,0,4       :rem 124
460 DATA16,16,32,32,0,0,0,0         :rem 64
470 DATA4,4,4,4,4,4,4,0            :rem 133
475 DATA255,199,189,207,245,143,255,255
                                   :rem 211
480 DATA255,129,145,169,169,169,145,255
                                   :rem 210
490 DATA255,129,145,177,145,145,185,255
                                   :rem 202
500 DATA255,129,153,165,137,145,189,255
                                   :rem 195
510 DATA255,129,185,137,153,137,185,255
                                   :rem 195
520 DATA255,129,153,169,189,137,137,255
                                   :rem 202
530 DATA255,129,189,161,185,133,185,255
                                   :rem 199
540 DATA255,129,157,161,185,165,153,255
                                   :rem 195
550 DATA255,129,189,133,137,145,145,255
                                   :rem 196
560 DATA255,129,153,165,153,165,153,255
                                   :rem 192
570 DATA255,129,153,165,157,133,185,255
                                   :rem 197
580 DATA20,58,28,119,8,54,8,54       :rem 1
590 DATA8,8,28,20,58,62,62,28      :rem 207
600 DATA28,62,127,73,8,8,40,16     :rem 245
610 DATA16,2,32,136,80,42,116,56    :rem 78
620 REM LOWER MEMORY 512 BYTES     :rem 253
630 POKE52,PEEK(52)-2:POKE56,PEEK(56)-2
                                   :rem 215
```

Program 2:

The Frantic Fisherman—Main Program, VIC Version

```
10 CL$="{WHT}WXY{5 LEFT}{DOWN}Z[ff]↑
   {5 LEFT}{DOWN}←!#$"             :rem 213
20 DEF FNRN(X)=INT(RND(1)*X)       :rem 111
30 V=36878:NO=V-1:S=V-2:S2=V-3:S3=V-4:CO=
   30720:EG=2000:TT=22:T6=256:Z=32:DE=29
                                   :rem 63
40 GOTO7000                        :rem 100
100 GL=3:SC=.                      :rem 207
105 EG=2000                         :rem 33
110 POKE36869,255                  :rem 153
120 PRINT"{CLR}{11 RIGHT}{18 DOWN}{GRN}@
    {LEFT}{DOWN}{WHT}A{2 LEFT}{DOWN}CB
    {3 LEFT}{DOWN}FED{3 LEFT}{DOWN}IHG
    {4 LEFT}{DOWN}{RED}JKKKL";     :rem 84
130 POKE646,10:PRINT"{4 LEFT}KKK":PRINT"
    {CYN}]]]]]]]]]]]]]]]]]]]]{LEFT}
    {INST}]{HOME}{3 DOWN}":POKEV,15+16*9
                                   :rem 181
140 POKE36879,238                  :rem 158
150 PRINT"{HOME}{6 DOWN}{2 RIGHT}"CL$"
    {7 UP}{4 RIGHT}"CL$"{3 DOWN}{3 RIGHT}
    "CL$;                          :rem 235
160 PRINT"{YEL}{8 UP}%&]]{4 LEFT}{DOWN}'(
    )]{3 LEFT}{DOWN}*+,{2 LEFT}{DOWN}-."
                                   :rem 252
170 PRINT"{HOME}{BLK}/"SC;:POKE646,8:PRIN
    T"{HOME}{DOWN}";:IF GL>1THENFORT=1TOG
    L-1:PRINT"S";:NEXT             :rem 44
180 LO=8128:POKELO+CO,10:POKELO,18:POKELO
    -TT,17:POKELO-TT+CO,10:POKELO-1,15
                                   :rem 175
190 POKELO-1+CO,.:CL=LO-1:POKE8132+CO,10:
    POKE8110+CO,10:POKE8133+CO,.   :rem 37
200 TY=FNRN(2)+1:ONTYGOTO210,300 :rem 158
210 X=FNRN(2)+1:ONXGOTO220,230   :rem 248
220 BC=8142:EC=8149:SP=1:DD=21:GOTO240
                                   :rem 176
230 BC=8163:EC=8155:SP=-1:DD=22  :rem 214
240 FORDL=BCTOECSTEPSP:POKEDL,DD:POKEDL+C
    O,.                            :rem 150
250 GOSUB1000:POKEDL,Z:NEXT:IFSD<>196THEN
    3000                            :rem 60
260 SD=.:GOTO200                   :rem 159
300 NU=.:Y=59:C=6:X=FNRN(2)+1:ONXGOTO310,
    320                            :rem 110
310 B=7776:E=8084:GOTO330          :rem 135
320 B=7758:E=8088                  :rem 131
330 FORDL=BTOESTEPTT:POKEDL+CO,C:POKEDL,Y
    :GOSUB1000                     :rem 118
340 POKEDL,Z:NEXT:IFSP<>T6THEN3000
                                   :rem 183
350 SP=.:GOTO200                   :rem 171
1000 CK=PEEK(197):IFCK=64THENFORR=0TODE:N
     EXT:RETURN                     :rem 53
1010 IFCK=29ANDLO=8132THEN1500    :rem 245
1020 IFCK=37ANDLO=8128THEN1750      :rem 1
1030 IFCK=ZTHENONTYGOTO2000,3500 :rem 224
1040 RETURN                        :rem 165
1500 POKELO,Z:POKELO-TT,Z:POKECL,Z:LO=812
     8:CL=LO-1                     :rem 215
1510 POKELO,18:POKELO-TT,17:POKECL,15:FOR
     SD=130TO150STEP2:POKES2,SD:NEXT:POKE
     S2,.                          :rem 191
1520 RETURN                        :rem 168
1750 POKELO,Z:POKELO-TT,Z:POKECL,Z:LO=813
     2:CL=LO+1                     :rem 215
1760 POKELO,20:POKELO-TT,19:POKECL,13
                                   :rem 29
1770 FORSD=150TO130STEP-2:POKES2,SD:NEXT:
     POKES2,.:RETURN               :rem 126
2000 POKECL,PEEK(CL)+1:FORSD=250TO200STEP
     -10:POKENO,SD:NEXT:IFPEEK(CL+TT)=DDT
     HEN2100                       :rem 50
2010 POKECL,PEEK(CL)-1:POKENO,.:RETURN
                                   :rem 138
2100 SC=SC+75:PRINT"{HOME}{BLK}/"SC:GOSUB
     4000                          :rem 180
```

```
2110 POKEDL,58:FORSD=254TO198STEP-2:POKED
     L+CO,FNRN(8):POKENO,SD:NEXT :rem 109
2120 POKENO,.:GOTO2010            :rem 33
3000 GL=GL-1:FORT=130TO254STEP2:POKES,T:P
     OKENO,T:POKEV,15+FNRN(16)*16:NEXT
                                  :rem 170
3010 POKELO-TT,218:FORT=15TO0STEP-.2:POKE
     S,.:POKENO,160:POKEV,T+FNRN(16)*16:N
     EXT                          :rem 29
3020 IFGL=.THEN7000               :rem 72
3030 POKENO,.:GOTO120             :rem 242
3500 IFNU>2THENRETURN             :rem 121
3510 NU=NU+1:POKELO-44,60:POKELO-44+CO,4:
     FORSD=150TO180STEP10:POKES2,SD:NEXT:
     POKES2,.                     :rem 95
3520 IFPEEK(LO-66)=59THEN3600     :rem 166
3530 POKE(LO-44),Z:RETURN         :rem 27
3600 SC=SC+50:PRINT"{HOME}{BLK}/"SC:GOSUB
     4000:POKELO-66,61            :rem 227
3610 FORSP=200TO254STEP2:POKES,SP:NEXT:PO
     KES,.                        :rem 233
3620 POKELO-66,Z:RETURN           :rem 206
4000 IFSC>=EGTHENGL=GL+1:DE=DE-4:EG=EG+20
     00:POKE7700+GL,19:POKE7700+CO+GL,8:G
     OTO4020                      :rem 30
4010 RETURN                       :rem 165
4020 FORT=130TO230STEP10:FORR=T+10TOTSTEP
     -1:POKES,T:NEXTR,T:POKES,.:RETURN
                                  :rem 127
7000 POKE36869,240:PRINTCHR$(8):IFSC>HSTH
     ENHS=SC                      :rem 238
7010 POKE36879,8:POKE646,10:PRINT"{CLR}
     {3 SPACES}{A}**{S}":PRINT"{3 SPACES}
     -{A}*{X}"                    :rem 147
7020 PRINT"{3 SPACES}-{Z}{R}*{R}*{R}**{R}
     **{2 R}*{S}{6 SPACES}-{A}{W}{A}{W}
     {D}-{A}{S}{Q}{S}{A}{W}-{A}{X}
     {6 SPACES}-----{SHIFT-SPACE}--------
     {Z}{S}"                      :rem 2
7030 PRINT"{A}**+{X}{Q}{W}{Z}{2 E}{X}{Z}
     {X}{Z}{X}{Z}{E}*{X}":PRINT"-{A}*{X}
     {SPACE}--"                   :rem 11
7040 PRINT"-{Z}{2 R}*{W}{Z}*{R}*{R}*{R}**
     {R}*{R}**{S} -{A}{W}-*{W}{A}{S}-{V}-
     {A}{W}{2 SPACES}-{D}-{A}{S}-":rem 60
7050 PRINT"---{Q}*----*{W}----- ---- {Z}
     {X}{Z}{E}*{E}{X}{Z}{E}*{E}{X}{Z}
     {5 E}{X}{Z}{X}"              :rem 22
7060 PRINT"{DOWN}{RED}{2 SPACES}LAST SCOR
     E:"SC:PRINT"{DOWN}{GRN}{2 SPACES}HIG
     H SCORE:"HS                  :rem 246
7070 PRINT"{PUR}{DOWN}{2 SPACES}HIT A KEY
      TO PLAY"                    :rem 52
7080 PRINT"{RVS}{WHT}{7 SPACES}CONTROLS
     {7 SPACES}{OFF}{PUR}{4 SPACES}<-LEFT
     ":PRINT"{GRN}{4 SPACES}>-RIGHT"
                                  :rem 255
7090 PRINT"{RVS}{BLU}SPACE{OFF}-CLUB OR U
     MBRELLA":POKENO,.            :rem 120
7100 POKE36878,(FNRN(14)+2)*16:IFPEEK(197
     )=64THEN7100                 :rem 218
7110 GOTO100                      :rem 147
```

Program 3:

The Frantic Fisherman—64 Version

```
4 POKE56,60:CLR                   :rem 123
5 GOSUB 8000                      :rem 125
10 POKE 53280,0:POKE 53281,0      :rem 182
20 PRINT "{CLR}{N}{5}{DOWN} {A}*********
   {S}{3 SPACES}PRESS ANY KEY TO BEGIN"
                                  :rem 129
30 PRINT " -{9 SPACES}-"          :rem 238
40 PRINT "{SHIFT-SPACE}-{2 SPACES}{A}****
   **{X}{5 SPACES}HIGH SCORE:";HS :rem 75
50 PRINT "{SHIFT-SPACE}-{2 SPACES}{Z}****
   **{S}{5 SPACES}YOUR SCORE:";SC:rem 100
60 PRINT "{SHIFT-SPACE}-{9 SPACES}-"
                                  :rem 145
70 PRINT "{SHIFT-SPACE}-{2 SPACES}{A}****
   **+*****{R}*****{R}*****{R}*{R}****{S}
   "                              :rem 54
80 PRINT "{SHIFT-SPACE}-{2 SPACES}-
   {6 SPACES}-{5 SPACES}-{5 SPACES}-
   {5 SPACES}- -{4 SPACES}-"      :rem 193
90 PRINT "{SHIFT-SPACE}-{2 SPACES}- {A}**
   **{W}{2 SPACES}{U}{2 SPACES}- {A}*{S}
   {SPACE}{Q}*{S} {A}*{W} - {A}**{X}"
                                  :rem 176
100 PRINT "{SHIFT-SPACE}-{2 SPACES}- -
    {4 SPACES}- {A}*{S} - - - - - - - - -
    {Z}**{S}"                     :rem 87
110 PRINT " -{2 SPACES}- -{4 SPACES}- - -
     - - - - - - - -{4 SPACES}-"  :rem 86
120 PRINT "{A}{E}**{E}*{E}***{S}{Z}*{X}
    {Z}*{E}*{X} {Z}*{X} {Z}*{X} {Z}*{E}**
    **{X}"                        :rem 73
130 PRINT "-{9 SPACES}- USE SPACE TO RAIS
    E UMBRELLA"                   :rem 227
140 PRINT "-{2 SPACES}{A}******{X}
    {6 SPACES}OR FEND OFF SHARK"  :rem 66
150 PRINT "-{2 SPACES}{Z}******{S}
    {3 SPACES}USE < AND > TO MOVE FROM"
                                  :rem 231
160 PRINT "-{9 SPACES}-{9 SPACES}LEFT TO
    {SPACE}RIGHT"                 :rem 110
170 PRINT "-{2 SPACES}{A}*{R}****+*{S}
    {A}*{R}****{R}****{R}****{R}***{R}***
    {S}";                         :rem 4
180 PRINT "-{2 SPACES}- - {A}**{W} - - -
    {SPACE}{A}**{W}{4 SPACES}-{4 SPACES}-
     {O} -{3 SPACES}-";           :rem 198
190 PRINT "-{2 SPACES}- - {Z}**{W} {Z}*
    {X} - {Z}*{S}- {A}**{W} -- -{A}*{S}-
    {A}*{S}-";                    :rem 140
200 PRINT "-{2 SPACES}- {Q}**{S} - {A}*
    {S} - {A}*{X}- -{2 SPACES}- -- -- ---
     --";                         :rem 146
210 PRINT "-{2 SPACES}- {Q}**{X} - - - -
    {SPACE}{Z}**{W} -{2 SPACES}- -- -- --
    - --";                        :rem 20
220 PRINT "{Z}**{E}*{E}****{E}*{X} {Z}*
    {E}****{E}*{X}{2 SPACES}{Z}*{2 E}*{E}
    {X} {Z}{E}{X} {Z}{X}";        :rem 235
230 GETA$:IFA$=""THEN230          :rem 77
240 POKE53281,14:POKE53280,6      :rem 38
245 PRINT"{CLR}"CHR$(142)"{BLK}SCORE:
    {19 SPACES}FISHERMEN:"        :rem 77
250 PRINT"{2 DOWN}{WHT}{13 SPACES}{D}
    {RVS}{4 SPACES}{OFF}{F}"      :rem 154
260 PRINT"{12 SPACES}{C}{RVS}{9 SPACES}
    {OFF}{3 I}{F}{5 SPACES}{D}{RVS}
    {2 SPACES}{OFF}{F}"           :rem 117
270 PRINT"{4 SPACES}{D}{RVS}{4 SPACES}
    {OFF}{F}{3 SPACES}{C}{RVS}{13 SPACES}
    {OFF}{V}{2 SPACES}{D}{RVS}{7 SPACES}
    {OFF}{F}"                     :rem 158
280 PRINT"{2 SPACES}{D}{RVS}{7 SPACES}
    {OFF}{F}{5 SPACES}{C}{RVS}{8 SPACES}
    {OFF}{V}{3 SPACES}{C}{RVS}{7 SPACES}
    {OFF}{V}"                     :rem 178
290 PRINT" {C}{RVS}{12 SPACES}{OFF}{V}
    {4 SPACES}{C}{RVS}{4 SPACES}{OFF}{V}
```

```
    {7 SPACES}{C}{RVS}{3 SPACES}{OFF}{V}"
                                        :rem 198
300 PRINT"{5 SPACES}{C}{RVS}{7 SPACES}
    {OFF}{V}"                           :rem 130
310 PRINT"{7 SPACES}{C}{RVS}{3 I}{OFF}"
                                        :rem 171
311 PRINT"{5 DOWN}{22 SPACES}{RVS}£{OFF}
    {W}"                                :rem 187
312 PRINT"{21 SPACES}{RVS}£ {OFF}{W}"
                                        :rem 103
313 PRINT"{20 SPACES}{RVS}£{2 SPACES}
    {OFF}{W}"                           :rem 104
314 PRINT"{19 SPACES}{RVS}£{3 SPACES}
    {OFF}{W}"                           :rem 105
315 PRINT"{18 SPACES}{RVS}£{4 SPACES}
    {OFF}{W}"                           :rem 106
316 PRINT"{17 SPACES}{RVS}£{5 SPACES}
    {OFF}{W}"                           :rem 107
317 PRINT"{16 SPACES}{RVS}£{6 SPACES}
    {OFF}{W}"                           :rem 108
318 PRINT"{16 SPACES}{Z}{6 E}{W}":rem 243
319 PRINT"{14 SPACES}{2}{*}{RVS}
    {10 SPACES}{OFF}£"                  :rem 47
320 PRINT"{RVS}{BLU}{39 SPACES}{OFF}{BLK}
    ";                                  :rem 244
330 POKE2023,160:POKE2023+54272,6 :rem 16
340 SYS49152                            :rem 155
350 PRINT"{HOME}{BLK}{12 DOWN}{12 RIGHT}P
    RESS RETURN KEY"                    :rem 138
360 GETA$:IFA$<>CHR$(13)THEN360     :rem 4
370 S1=PEEK(829):S2=PEEK(830):S3=PEEK(831
    )                                   :rem 144
380 SC=INT(S1/16)*10+(S1AND15)+INT(S2/16)
    *1000+(S2AND15)*100                 :rem 234
390 SC=SC+INT(S3/16)*100000+(S3AND15)*100
    00                                  :rem 41
400 IF SC>HS THEN HS=SC                 :rem 47
410 GOTO10                              :rem 47
1000 DATA192,000,000,224,000,000,112,000
                                        :rem 167
1001 DATA000,056,000,000,028,000,000,014
                                        :rem 170
1002 DATA000,000,007,000,000,003,128,000
                                        :rem 166
1003 DATA001,128,000,000,000,000,000,000
                                        :rem 158
1004 DATA000,000,000,000,000,000,000,000
                                        :rem 147
1005 DATA000,000,000,000,000,000,000,000
                                        :rem 148
1006 DATA000,000,000,000,000,000,000,000
                                        :rem 149
1007 DATA000,000,000,000,000,000,000,000
                                        :rem 150
1008 DATA001,128,000,003,128,000,007,000
                                        :rem 184
1009 DATA000,014,000,000,028,000,000,056
                                        :rem 178
1010 DATA000,000,112,000,000,224,000,000
                                        :rem 156
1011 DATA192,000,000,000,000,000,000,000
                                        :rem 157
1012 DATA000,000,000,000,000,000,000,000
                                        :rem 146
1013 DATA000,000,000,000,000,000,000,000
                                        :rem 147
1014 DATA000,000,000,000,000,000,000,000
                                        :rem 148
1015 DATA000,000,000,000,000,000,000,053
                                        :rem 157
1016 DATA007,000,000,063,224,000,255,248
                                        :rem 200
1017 DATA000,002,000,000,002,000,000,002
                                        :rem 157
1018 DATA000,000,002,000,000,002,000,000
                                        :rem 156
1019 DATA002,000,000,018,000,000,012,000
                                        :rem 167
1020 DATA000,000,000,000,000,000,000,000
                                        :rem 145
1021 DATA000,000,000,000,000,000,000,000
                                        :rem 146
1022 DATA000,000,000,000,000,000,000,000
                                        :rem 147
1023 DATA000,000,000,000,000,000,000,000
                                        :rem 148
1024 DATA016,000,000,056,000,000,124,000
                                        :rem 174
1025 DATA000,254,000,000,158,000,000,206
                                        :rem 183
1026 DATA000,000,124,000,000,000,000,000
                                        :rem 158
1027 DATA000,000,000,000,000,000,000,000
                                        :rem 152
1028 DATA000,000,000,000,000,000,000,000
                                        :rem 153
1029 DATA000,000,000,000,000,000,000,000
                                        :rem 154
1030 DATA000,000,000,000,000,000,000,000
                                        :rem 146
1031 DATA000,000,000,000,000,000,000,000
                                        :rem 147
1032 DATA000,000,000,000,000,000,000,000
                                        :rem 148
1033 DATA000,000,000,000,000,000,000,000
                                        :rem 149
1034 DATA004,000,000,012,000,000,028,000
                                        :rem 167
1035 DATA000,060,001,007,255,195,014,127
                                        :rem 207
1036 DATA255,031,255,255,127,255,255,056
                                        :rem 237
1037 DATA127,255,003,255,195,000,000,001
                                        :rem 206
1038 DATA000,000,000,000,000,000,000,000
                                        :rem 154
1039 DATA000,000,000,000,000,000,000,183
                                        :rem 167
1040 DATA000,000,000,000,000,000,000,000
                                        :rem 147
1041 DATA000,000,000,000,000,000,000,000
                                        :rem 148
1042 DATA032,000,000,048,000,000,056,000
                                        :rem 177
1043 DATA128,060,000,195,255,224,255,254
                                        :rem 225
1044 DATA112,255,255,248,255,255,254,255
                                        :rem 240
1045 DATA254,028,195,255,192,128,000,000
                                        :rem 223
1046 DATA000,000,000,000,000,000,000,000
                                        :rem 153
1047 DATA000,000,000,000,000,000,000,183
                                        :rem 166
1048 DATA000,000,000,000,000,000,000,000
                                        :rem 155
1049 DATA000,000,255,000,000,255,000,003
                                        :rem 183
1050 DATA255,192,000,085,000,000,089,000
                                        :rem 202
1051 DATA000,085,064,000,090,000,000,085
                                        :rem 194
1052 DATA000,000,255,000,003,255,192,003
                                        :rem 192
```

```
1053 DATA245,080,003,255,192,003,255,192
                                   :rem 224
1054 DATA003,255,192,000,000,000,000,000
                                   :rem 179
1055 DATA000,000,000,000,000,000,000,000
                                   :rem 153
1056 DATA000,000,000,000,000,000,000,000
                                   :rem 154
1057 DATA000,000,255,000,000,255,000,003
                                   :rem 182
1058 DATA255,192,000,085,000,000,101,000
                                   :rem 195
1059 DATA001,085,000,000,165,000,000,085
                                   :rem 196
1060 DATA000,000,255,000,003,255,192,005
                                   :rem 193
1061 DATA095,192,003,255,192,003,255,192
                                   :rem 230
1062 DATA003,255,192,000,000,000,000,000
                                   :rem 178
1063 DATA000,000,000,000,000,000,000,000
                                   :rem 152
1064 DATA 256                      :rem 130
8000 PRINT"{N}{CLR}{12 DOWN}{RIGHT}LOADIN
     G SPRITES AND MACHINE LANGUAGE"
                                   :rem 87
8010 PRINT"{10 RIGHT}{4 DOWN}PLEASE BE PA
     TIENT..."                     :rem 96
9000 I=248*64                      :rem 129
9010 READ A:IF A=256 THEN 9100      :rem 4
9020 POKE I,A:I=I+1:CK=CK+A:GOTO 9010
                                   :rem 81
9100 IF CK<>19128 THEN PRINT"ERROR IN DAT
     A (LINES 1000-1064)":STOP     :rem 38
10000 I=49152:CK=0                 :rem 177
10010 READ A:IF A=256 THEN 10100   :rem 84
10020 POKE I,A:I=I+1:CK=CK+A:GOTO 10010
                                   :rem 161
10100 IF CK<>139243 THEN PRINT"ERROR IN D
      ATA (LINES 49152-50346)":STOP
                                   :rem 250
10200 RETURN                       :rem 211
49152 DATA 169,3,141,64,3,169      :rem 161
49158 DATA 7,141,21,208,169,217     :rem 5
49164 DATA 141,1,208,169,1,141     :rem 198
49170 DATA 28,208,169,10,141,37     :rem 0
49176 DATA 208,169,7,141,38,208    :rem 13
49182 DATA 169,0,141,39,208,32     :rem 207
49188 DATA 60,193,169,25,141,60    :rem 10
49194 DATA 3,169,250,141,250,7     :rem 207
49200 DATA 169,209,141,5,208,169   :rem 53
49206 DATA 2,141,41,208,169,44     :rem 202
49212 DATA 32,238,193,32,156,195   :rem 55
49218 DATA 169,0,141,61,3,141      :rem 148
49224 DATA 62,3,141,63,3,32        :rem 45
49230 DATA 174,195,32,141,196,169:rem 110
49236 DATA 32,141,71,3,169,0       :rem 100
49242 DATA 141,72,3,173,60,3        :rem 97
49248 DATA 141,67,3,32,4,196       :rem 112
49254 DATA 32,238,193,206,67,3     :rem 213
49260 DATA 208,245,141,4,212,32    :rem 244
49266 DATA 150,192,173,64,3,201    :rem 254
49272 DATA 0,208,226,169,0,133     :rem 201
49278 DATA 198,169,0,141,21,208     :rem 9
49284 DATA 169,0,141,4,212,169     :rem 208
49290 DATA 0,162,0,157,0,208        :rem 94
49296 DATA 232,224,17,208,248,96    :rem 66
49302 DATA 173,30,208,141,69,3     :rem 200
49308 DATA 173,65,3,201,1,240      :rem 145
49314 DATA 87,173,66,3,201,0       :rem 103
49320 DATA 240,46,173,69,3,41      :rem 154
49326 DATA 6,201,6,208,11,169      :rem 153
49332 DATA 117,32,31,195,32,179      :rem 2
49338 DATA 194,76,46,193,173,2     :rem 224
49344 DATA 208,201,225,208,3,76    :rem 254
49350 DATA 37,193,206,2,208,173      :rem 1
49356 DATA 2,208,201,255,208,5     :rem 203
49362 DATA 169,0,141,16,208,96     :rem 212
49368 DATA 173,2,208,201,115,144    :rem 46
49374 DATA 3,76,37,193,173,69      :rem 178
49380 DATA 3,41,6,201,6,208         :rem 47
49386 DATA 11,169,117,32,31,195      :rem 7
49392 DATA 32,179,194,76,46,193     :rem 25
49398 DATA 238,2,208,96,173,3      :rem 173
49404 DATA 208,201,227,144,3,76    :rem 252
49410 DATA 37,193,173,69,3,41      :rem 161
49416 DATA 6,201,6,208,11,169      :rem 153
49422 DATA 80,32,31,195,32,202     :rem 196
49428 DATA 194,76,46,193,173,69     :rem 29
49434 DATA 3,41,3,201,3,240         :rem 37
49440 DATA 4,238,3,208,96,32       :rem 109
49446 DATA 151,194,206,64,3,32     :rem 208
49452 DATA 156,195,162,30,32,106    :rem 49
49458 DATA 195,202,208,250,165,162
                                   :rem 157
49464 DATA 201,192,144,38,169,0      :rem 4
49470 DATA 141,2,208,169,229,141    :rem 51
49476 DATA 3,208,169,253,141,249    :rem 64
49482 DATA 7,169,0,141,40,208      :rem 156
49488 DATA 141,16,208,141,65,3     :rem 210
49494 DATA 141,66,3,169,0,141      :rem 159
49500 DATA 27,208,173,30,208,96      :rem 2
49506 DATA 201,128,144,44,169,80    :rem 53
49512 DATA 141,2,208,169,229,141    :rem 48
49518 DATA 3,208,169,252,141,249    :rem 60
49524 DATA 7,169,0,141,40,208      :rem 153
49530 DATA 169,2,141,16,208,169      :rem 4
49536 DATA 0,141,65,3,169,1         :rem 54
49542 DATA 141,66,3,169,0,141      :rem 153
49548 DATA 27,208,173,30,208,96     :rem 14
49554 DATA 201,64,144,44,169,139    :rem 60
49560 DATA 141,2,208,169,100,141    :rem 39
49566 DATA 3,208,169,251,141,249    :rem 62
49572 DATA 7,169,6,141,40,208      :rem 162
49578 DATA 169,0,141,16,208,169     :rem 14
49584 DATA 1,141,65,3,169,0         :rem 57
49590 DATA 141,66,3,169,2,141      :rem 158
49596 DATA 27,208,173,30,208,96     :rem 17
49602 DATA 169,218,141,2,208,169    :rem 56
49608 DATA 100,141,3,208,169,251    :rem 45
49614 DATA 141,249,7,169,6,141     :rem 214
49620 DATA 40,208,169,0,141,16     :rem 198
49626 DATA 208,169,1,141,65,3      :rem 160
49632 DATA 169,1,141,66,3,169      :rem 164
49638 DATA 2,141,27,208,173,30     :rem 205
49644 DATA 208,96,201,44,208,17      :rem 7
49650 DATA 169,132,141,0,208,169    :rem 52
49656 DATA 255,141,248,7,32,227     :rem 11
49662 DATA 194,32,112,195,96,201    :rem 57
49668 DATA 46,208,17,169,212,141    :rem 62
49674 DATA 0,208,169,254,141,248    :rem 61
49680 DATA 7,32,227,194,32,112     :rem 207
49686 DATA 195,96,201,32,208,113    :rem 60
49692 DATA 173,68,3,201,0,208      :rem 157
49698 DATA 115,173,70,3,201,0      :rem 153
49704 DATA 208,108,173,65,3,201    :rem 253
49710 DATA 0,240,38,169,2,141      :rem 148
49716 DATA 41,208,169,209,141,5      :rem 6
49722 DATA 208,169,250,141,250,7    :rem 51
49728 DATA 173,0,208,201,132,208    :rem 44
49734 DATA 8,169,138,141,4,208     :rem 217
49740 DATA 76,132,194,169,218,141:rem 112
```

```
49746 DATA 4,208,76,132,194,169   :rem 19
49752 DATA 0,141,41,208,173,0     :rem 145
49758 DATA 208,201,132,208,18,169:rem 109
49764 DATA 127,141,4,208,169,230  :rem 55
49770 DATA 141,5,208,169,249,141  :rem 59
49776 DATA 250,7,76,132,194,169   :rem 22
49782 DATA 232,141,4,208,169,230  :rem 52
49788 DATA 141,5,208,169,248,141  :rem 67
49794 DATA 250,7,32,1,195,169     :rem 170
49800 DATA 200,141,68,3,96,201    :rem 198
49806 DATA 95,208,5,169,0,141     :rem 164
49812 DATA 64,3,96,32,141,196     :rem 165
49818 DATA 169,33,141,4,212,162     :rem 2
49824 DATA 255,142,1,212,142,37   :rem 251
49830 DATA 208,32,106,195,202,208 :rem 97
49836 DATA 244,169,10,141,37,208  :rem 57
49842 DATA 96,32,141,196,169,129  :rem 71
49848 DATA 141,4,212,162,255,142  :rem 50
49854 DATA 1,212,142,40,208,32    :rem 196
49860 DATA 106,195,202,208,244,96:rem 110
49866 DATA 32,141,196,169,129,141:rem 116
49872 DATA 4,212,162,0,142,1      :rem 94
49878 DATA 212,142,40,208,32,106  :rem 48
49884 DATA 195,232,224,50,208,242:rem 108
49890 DATA 96,169,33,141,4,212    :rem 216
49896 DATA 162,15,142,1,212,32    :rem 203
49902 DATA 106,195,32,106,195,202 :rem 99
49908 DATA 224,5,208,242,169,0    :rem 211
49914 DATA 141,4,212,32,106,195   :rem 251
49920 DATA 96,169,33,141,4,212    :rem 210
49926 DATA 162,5,142,1,212,32     :rem 148
49932 DATA 106,195,32,106,195,232:rem 105
49938 DATA 224,20,208,242,169,0     :rem 3
49944 DATA 141,4,212,32,106,195   :rem 254
49950 DATA 96,248,24,109,61,3     :rem 168
49956 DATA 141,61,3,169,0,109     :rem 161
49962 DATA 62,3,141,62,3,169      :rem 112
49968 DATA 0,109,63,3,141,63      :rem 111
49974 DATA 3,216,32,174,195,56    :rem 222
49980 DATA 173,62,3,237,71,3      :rem 113
49986 DATA 141,69,3,173,63,3      :rem 121
49992 DATA 237,72,3,13,69,3       :rem 69
49998 DATA 144,25,169,32,248,24   :rem 22
50004 DATA 109,71,3,141,71,3      :rem 85
50010 DATA 169,0,109,72,3,141     :rem 136
50016 DATA 72,3,216,238,64,3      :rem 97
50022 DATA 32,156,195,96,160,0    :rem 197
50028 DATA 200,208,253,96,169,0   :rem 250
50034 DATA 141,41,208,173,0,208   :rem 236
50040 DATA 201,132,208,16,169,248 :rem 87
50046 DATA 141,250,7,169,127,141  :rem 41
50052 DATA 4,208,169,222,141,5    :rem 193
50058 DATA 208,96,169,249,141,250:rem 109
50064 DATA 7,169,232,141,4,208    :rem 199
50070 DATA 169,222,141,5,208,96   :rem 252
50076 DATA 162,0,160,35,24,32     :rem 139
50082 DATA 240,255,173,64,3,24    :rem 197
50088 DATA 105,48,32,210,255,96     :rem 0
50094 DATA 162,0,160,6,32,240     :rem 137
50100 DATA 255,173,63,3,41,240    :rem 186
50106 DATA 74,74,74,74,24,105     :rem 154
50112 DATA 48,32,210,255,173,63   :rem 243
50118 DATA 3,41,15,24,105,48      :rem 91
50124 DATA 32,210,255,173,62,3    :rem 188
50130 DATA 41,240,74,74,74,74     :rem 150
50136 DATA 24,105,48,32,210,255   :rem 241
50142 DATA 173,62,3,41,15,24      :rem 89
50148 DATA 105,48,32,210,255,173  :rem 41
50154 DATA 61,3,41,240,74,74      :rem 96
50160 DATA 74,74,24,105,48,32     :rem 149
50166 DATA 210,255,173,61,3,41    :rem 193
50172 DATA 15,24,105,48,32,210    :rem 187
50178 DATA 255,96,32,16,196,72    :rem 219
50184 DATA 32,71,196,32,125,196     :rem 2
50190 DATA 104,96,32,228,255,201  :rem 41
50196 DATA 0,208,3,76,70,196      :rem 108
50202 DATA 201,133,208,7,169,25   :rem 241
50208 DATA 141,60,3,169,133,201   :rem 238
50214 DATA 134,208,7,169,18,141   :rem 250
50220 DATA 60,3,169,134,201,135   :rem 236
50226 DATA 208,7,169,13,141,60    :rem 198
50232 DATA 3,169,135,201,136,208  :rem 37
50238 DATA 7,169,9,141,60,3       :rem 55
50244 DATA 169,136,96,173,68,3    :rem 218
50250 DATA 240,5,206,68,3,240     :rem 140
50256 DATA 21,173,70,3,201,0      :rem 83
50262 DATA 240,3,206,70,3,162     :rem 137
50268 DATA 90,202,208,253,169,0   :rem 252
50274 DATA 141,4,212,96,173,4     :rem 149
50280 DATA 208,201,0,240,5,169    :rem 189
50286 DATA 0,141,4,208,32,112     :rem 136
50292 DATA 195,169,255,141,70,3     :rem 3
50298 DATA 76,81,196,173,141,2    :rem 215
50304 DATA 41,1,201,1,208,6       :rem 28
50310 DATA 32,16,196,76,125,196     :rem 0
50316 DATA 96,162,0,169,0,157     :rem 154
50322 DATA 0,212,232,224,25,208   :rem 231
50328 DATA 248,169,15,141,24,212  :rem 45
50334 DATA 169,16,141,5,212,169   :rem 252
50340 DATA 240,141,6,212,169,100  :rem 26
50346 DATA 141,0,212,96,256       :rem 51
```

Bug-Swatter:
Modifications And Corrections

• Reader Clifford Tener has discovered a minor error in "Poker" (March). The VIC and 64 versions see an ace as a high card, which eliminates the possibility of small straights (A-2-3-4-5). To correct this, make the following changes:

```
2110 YY=0:IFPT(4)-PT(3)=1THENIFPT(3)-PT(2
     )=1THENIFPT(2)-PT(1)=1THENYY=1

2115 IFYY=1THENIF(PT(5)-PT(4)=1)OR(PT(1)+
     PT(5)-15=1)THENSS=1
```

• Program 4 of "How To Use Arrays" (February) contains misplaced characters in 5180 and 5270. In each of these lines, replace the first double quote with a number sign (#):

```
5180 INPUT#4,HW(N)
5270 PRINT#1,"NAME","SCORE"
```

• Program 4 of "Making Calendars" (April) crashes when printing calendars for the years following 2200 A.D. Readers who like to plan 200 years in advance should make the following change to line 1247:

```
1247 IF(Y=2200ANDM0>3)OR(Y>2200)THEND1=D1
     -1:IFD1=0THEND1=7
```

• The Commodore 64 version of "React" (February) runs as listed, but does not correctly read the forward diagonals of the joystick. To fix it, change line 640. JS(5) should be –41 and JS(9) should be –39. Thanks to Paul T. Dawson for discovering this error.

• Reader Scott Campbell finds it more convenient to use the space bar rather than the M key to represent zero in "Numeric Keypad" (April). Pressing M with a thumb is rather awkward. To make the switch, change the 77 in line 520 to 32 (64 version).

PRODUCT MART

TAKE YOUR COMMODORE 64 TO THE MOVIES

The MOVIES Disc

Use your 64 as a movie guide. Display vital information on many of the latest movies on pay TV and for rent or sale on videocassettes and video-discs. Quickly call up capsule reviews and ratings. Find information fast on major stars, directors, Academy Awards, MPAA ratings, box office bestsellers and more. Helps you decide before you buy, rent or watch.

2-disc set for '83 and '82 movies, only $19.95 postpaid.
Send check or money order to:

CINEMAN SYNDICATE
Suite 501, 7 Charles Court
Middletown, NY 10940

NY Res. add appropriate taxes. Canadian and other foreign customers: U.S. funds only

Commodore 64 is a registered trademark of Commodore Electronics Ltd.

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)

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

20	64		20	64	
☐		Programmer's Aid[1]	☐	☐	Graphic printer (1525 & MPS801)[1]
☐		Vicmon[1]	☐		UMI Wordcraft 20
☐		Super Expander[1]	☐		HES Vic Forth
☐		Vic Typewriter[1]	☐	☐	HES Writer
☐		Victerm 1[1]		☐	Wordpro 3 plus
	☐	Term 64[1]		☐	Easy Script[1]
☐	☐	Quick Brown Fox	☐	☐	Basic
	☐	Hesmon 64		☐	Paper Clip
	☐	Pilot (Commodore)		☐	Script 64
	☐	Calc Result (advanced)	☐	☐	PractiCalc 64/plus
	☐	Calc Result (easy)		☐	Easy Calc[1]

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

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

HOME FINANCE PACKAGE

A Powerful Collection For
The Commodore 64

- Budget Planning
- Checkbook Mgmt.
- Loan Analysis
- Investment Planning
- Mortgage Analysis
- IRA Mgmt.
- Retirement Planning
- Savings Mgmt.

ONLY **$14.95** Diskette
or **$13.95** Cassette

Send Check or Money Order To:

SMS
P.O. Box E
Nampa, Idaho 83653

Software Marketing Service

osirisan™

A 6502 Machine Language Assembler for the SERIOUS Users of the VIC-20 and C-64

Professional Features

- High Speed Operation
- Choice of I/O Devices
- Full Listing Control
- Disassembly Utility
- File Chaining
- Produces Industry-Standard Object Code

INCLUDES
FREDITOR
A full screen, menu-driven text editor with:
- Scrolling
- Error Messages
- Block Operations
- Search/Replace Functions

(Requires 8K RAM)

Specify **Disk** or **Cassette** **34.95**

DOCUMENTATION ONLY . . . $10
(Refundable with order)
(California Residents — Add 6% Tax)

OSIRIS P.O. BOX 11004 SANTA ROSA, CA 95406 (707) 576-0808

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 unexp. (T) or (D)	$19.95	$14.95
PAK ALIEN (maze) 100% ML	VIC unexp. (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.

WIZARD'S DOMINION

ONLY THE BRAVEST DARE TO ENTER!! ARM YOUR CHARACTER WITH WEAPONS AND MAGIC, THEN FIND GOLD AND FIGHT OGRES AND GAIN NEW MAGICAL POWERS.

- *3-D PERSPECTIVE*
- *SUPERB GRAPHICS*
- *COMPLEX BATTLES*
- *LOTS OF MAGIC*
- *THOUSANDS OF CAVES*

COMMODORE 64

TI 99/4A (EXTENDED BASIC)

CASSETTE $19.95 DISK $21.95

DEALER INQUIRIES WELCOME

INSTRUCTIONS INCLUDED. To order send check or money order plus $1.50 shipping/handling to:

American Software
Design & Distribution Co.
P.O. Box 246 Dept. G-6
Cottage Grove, MN 55016

CONVERSE WITH YOUR COMPUTER

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

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

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

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

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

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

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

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

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

VISA MasterCard

COMMODORE SOFTWARE AFICIONADOS

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

For details, write and send a sample review to:

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

"THE REUNION"©

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

"THE REUNION" simultaneously interfaces your *VIC 20 and *C/64 (including *Datasette, modem, etc.) to your disk drive and/or printer **providing 2 computer systems.** Use either instantly.

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

SAVE and LOAD VIC and C/64 programs on same disk or tape. Ends switching disk/tape and cables.

*Simple two minute installation. Full year warranty.

Model A-interfaces VIC 20/C-64/disk/printer. Model B-interfaces VIC 20/C-64 and Datasette.

Special introductory price - $29.95 ea. (U.S. $).

Please add $2.00 for shipping (Canadian: $4.00). Florida resident 5% tax. Send check or money order to: HyTech
P.O. Box 466
Bay Pines, FL 33504

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

FREE GAME FOR YOUR COMMODORE 64

That's right! If you'll help cover shipping costs, KIDware will send you a free game along with our latest program brochure. The cassette-based game, FOUR-IN-A-ROW, lets 2 players compete at trying to line up four pieces on a playing grid. Its fun for the whole family!

KIDware specializes in KID-oriented software (Cassettes and Disks) for the Commodore 64. We offer a wide range of fun, educational programs for kids 1 - 16 years of age, with emphasis in the younger years. Our prices are the lowest and our quality and service the best----you get same day shipping on all orders!

To receive your game tape and program brochure, send $1.50 (for shipping) to KIDware. If you only want a brochure, simply write and ask.

KIDware
p.o. box 1664
idaho falls,
idaho 83401

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

PROTO-64

The only prototyping board for the COMMODORE 64™

- **Plugs directly into the expansion port**
- **44 contacts - 22 per side**
- **.100 inch spacing pad per hole**

ORDER NOW
INTRODUCTORY OFFER
available in two sizes

4.5"×4" $12.95
6.5"×4.5" . . . $15.95

TM is a trademark of Commodore Electronics Ltd.

Add $1.00 shipping
CO res. add 6% tax

Boreas Products
P.O. Box 16961
Co. Springs, Co. 80935
(303) 593-1274
Dealer Inquiries Invited

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.

ADVERTISERS INDEX

Reader Service Number/Advertiser	Page
102 Aardvark Action Software	139
103 Academy Software	36
104 Access Software Inc.	40,41
105 Advanced Ideas	19
A.I.D. Corp	191
106 American Software Design & Distribution Co.	191
Aries Marketing Co.	190
107 Artificial Intelligence Research Group	191
Assembly Technology	147
Atari, Inc.	27
108 Avalon Hill Game Company	7
109 Batteries Included	39
110 Batteries Included	93
Big Bytes	142
The Book Company	191
Boreas Products	191
Boston Educational Computing, Inc.	117
Brantford Educational Services	122
Bytes and Bits	191
111 Bytes & Pieces, Inc.	115
112 Cadmean Corp.	134
CapitalComp, inc.	65
113 Cardco, Inc.	IBC
Century Micro Products	73
Cheatsheet Products	189
Chromazone Software	190
Cineman Syndicate	189
Cloneware	189
CMS Software	190
Columbia Software	118
Commodore Computers	BC
CompuServe	62
114 CompuServe	63
Computer Mail Order	143
115 ComputerMat	122
Computer Place	104
116 Continental Software	45
117 Covox Inc.	189
Creative Software	4
118 davidson & associates	69
Datasoft, Inc.	2,3
Dazco	190
Dennison	29
dilithium Press	25
119 Diversified Manufacturing	147
D & L Computers	190
Dow Jones News/Retrieval	101
120 Eastern House	113
121 Eastern House	116
122 Educomp	132
Elcomp Publishing, Inc.	87
Electronic Arts	43
Electronic Arts	51
Epyx	53
Epyx	55
French Silk	14
123 Futurehouse	75
Genesis Computer Corporation	141
124 GOSUB of Slidell, Inc.	137
125 Handic Software Inc.	15
HyTech	191
INMAC	129
Jamestown Software	149
Jason-Ranheim	134
126 J & H Computers	190
John Henry Software	142
John Wiley & Sons, Inc.	126
Joy of Programming	190
KIDware	191
Kiwisoft Programs	147
127 Lynn Computer Service	104
Metaphase Software	24
128 MFJ Enterprises Incorporated	101
129 MicroProse Software	85
130 Micro Sci Corp.	65
131 Micro Sci Corp.	67
Micro Software International, Inc.	119
132 Micro Ware	48
133 Micro Ware	73
Micro World Electronics, Inc.	103
134 Micro Worx	131
135 Mirage Concepts, Inc.	17
136 Oakwood Computer Products	67
Ohio Computer Services, Inc.	77
137 Orange Micro Inc.	91
138 Orbyte Software	37
139 Osiris	189
140 Panther Computer Corporation	IFC
Parallel Systems	60
Parsec Research	97
141 PB Systems	147
142 PC Gallery	134
143 Precision Software, Inc.	1
Prentice-Hall	35
144 Professional Software, Inc.	9
Pro-Line Software	95
145 Protecto Enterprizes	106,107
146 Protecto Enterprizes	108,109
147 Protecto Enterprizes	110,111
148 Public Domain, Inc.	190
Quicksilva Inc.	61
Reston Computer Group	13
149 Richvale Telecommunications	71
150 Rockware Data	113
Scarborough Systems, Inc.	11
Sierra On-Line, Inc.	79
151 The 64 Club	147
152 SJB Distributors Inc.	142
SJB Distributors Inc.	145
SM Software Inc.	56
SM Software Inc.	56
153 Softext Inc.	30,31
154 Soft-Guide	188
Softlaw Corporation	105
155 SoftPeople Inc.	47
156 Software Discounters of America	103
Software Marketing Service	189
Software Masters	117
Software Plus	149
157 Software Shopper	149
158 SoftWare Warehouse Outlet	145
159 Spectrum Software	190
Spinnaker	21
Spinnaker	23
Starpoint Software	97
160 subLOGIC Corporation	57
161 Such A Deal	121
Sunshine Software	191
162 Susie Software	188
Synergy Software, Inc.	96
163 Systems Management Associates	99
164 Systems Management Associates	123
165 Tenex Computer Marketing Systems	83
3G Company, Inc.	60
166 Timeworks, Inc.	89
167 Totl Software, Inc.	132
Tussey Mountain Software	149
Tymac Incorporated	59
Ultrabyte	128
VIN Systems (U.S.A.)	147
Virginia Micro Systems	145
York 10	141

COMPUTE! Books	80,81
COMPUTE!'s GAZETTE Disk	33
COMPUTE!'s GAZETTE Subscriber Services	187
COMPUTE!'s GAZETTE Subscription	49

"Commodore-ready", . . . and ready for you NOW!

"Cardcorder" DC/1, Data Cassette Recorder/Player

Introducing the "CARDCORDER", Model DC/1, the Computer Cassette that is "Commodore-ready," designed for storage and retrieval of computer data efficiently, economically; with consistent performance. Yet, this fine CARDCO product is priced lower than any similar product with special quality features.

Includes standard connector which is "Commodore-ready"; LED "save" indicator light which confirms data recording on to the tape; handles up to 120 minutes (60 minutes on each side) of any standard tape including existing pre-recorded commercial as well as personal data tapes intended for use with Commodore Personal Computers; ready to go . . . just plug it in and record efficiently.

CARDCO's "CARDCORDER" COMPUTER CASSETTE is a quality data cassette recorder/player in an attractive polystyrene case, with all the standard cassette functions: record . . . play . . . rewind . . . fast forward . . . stop and eject . . . pause. A solid-state designed product of the finest components with auto-stop.

The "CARDCORDER" DC/1 carries a 90 day warranty to original owners.

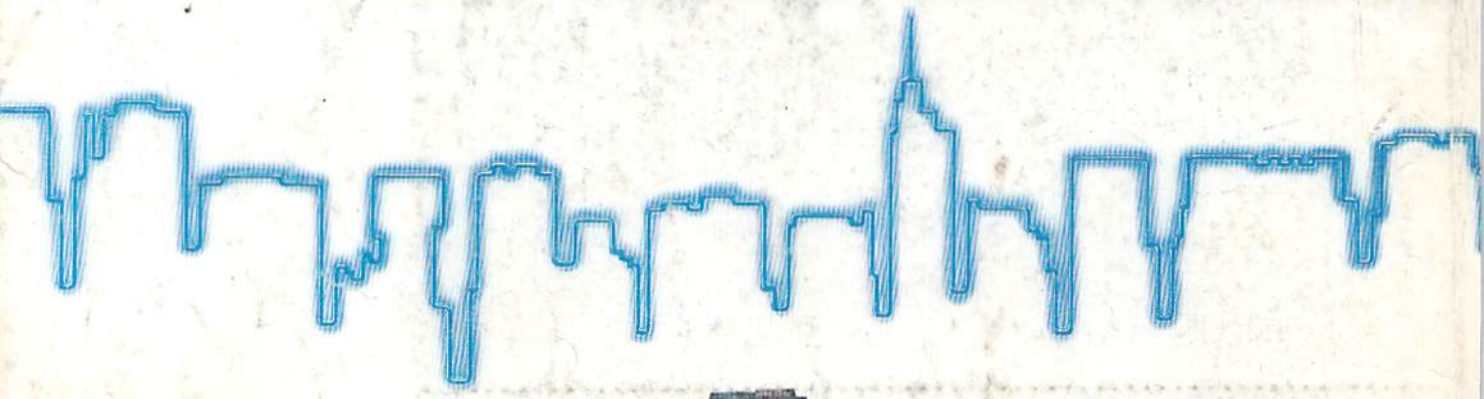
All CARDCO products are available at your local dealers.

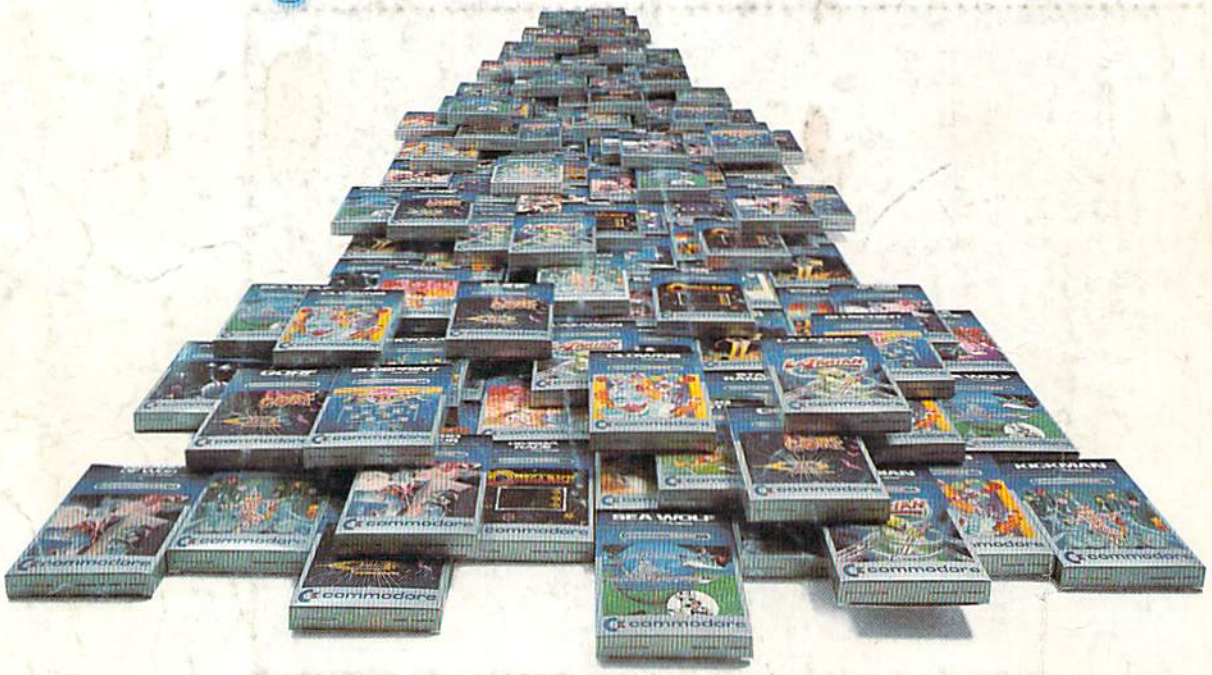
cardco, inc.

313 Mathewson Wichita, Kansas 67214 (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