Super Action Software!

Commodore 64 | PROGRESSIVE PERIPHERALS & SOFTWARE | Disk Programs

(World's Greatest Adventure Game)

CYBERWORLD List $49.95 Sale $34.95

NEW

This five-screen arcade adventure packs the computer with intense graphics and sound! You are a special Cyberleague agent in a universe full of hostile aliens and vicious robots. Joystick and keyboard transport you through 3-D rooms, space barriers, fleets of invaders, and warship-ridden quadrants of space. Over 100 sprites, 8 new character sets, and dozens of mind-boggling sound effects make up this multi-layered adventure. Animation, action, and strategy all combined into a game so extensive that two disk sides are jammed with game programs and data! Reach the ultimate rank of admiral and you may carve a niche in the permanent high-score list. A full-size book quality manual with full-color covers is included to guide you through your most exciting game experience.

NEW

Perplexian Challenger

List $34.95 **Sale $24.95**

The incredibly responsive three-axis joystick control of a space fighter is in your hands. Split screen graphics provide a continuous display of your ship's instrumentation, as well as a three-dimensional, animated view of space.

You, as a pilot, must utilize lightning fast reflexes to destroy invading ships, and avoid their return fire. Simultaneously, you must maneuver your ship to capture space debris that remains from the explosions.

Outstanding graphics features include smooth 3-D rotations, split screens, and the most incredible high-resolution hyperspace sequence ever produced.

Programmed entirely in machine language, this action-strategy game is guaranteed to blow you away.

All the professional features you expect are included: automatic self-demo, high score retention, pause, and provisions for 1 to 4 players. Add to this, features you don't expect like easy-loading, and music during the load. Perplexian Challenger is a game that brings the arcade experience to your home.

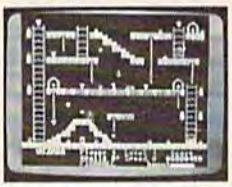
Wizard

List $49.95 **Sale $34.95**

Jump from ropes to ladders, dodge plummeting boulders and duck under deadly arrows in your quest for sparkling diamonds, gleaming bars of gold, and glistening pearls. With joystick in hand you must explore forty dazzling screens, each a new and exciting adventure. Take the key to unlock the doorway to your next spine-tingling level. Each key restores your magical powers, allowing you to cast over ten different spells. With these magic spells you have the power to overcome vicious creatures, terrifying traps, and perilous plunges.

Your Wizard is realistically animated in every possible direction. Dozens of movements are possible — jump over burning fires, shimmy up or down ropes and ladders, even slip down treacherous sliding staircases! Magic portals move your Wizard through midair and protect you from a myriad of fully-animated fiendish monsters. Catch an elevator to the top of the screen and dart through sliding gates in your quest for magic and treasure.

Wizard's fascinating variety of screens are sure to please and entertain, and of course you can build an unlimited number of your own levels using the construction set provided with your game.



Gothmog's Lair

List $44.95 **Sale $34.95**

Real-time adventure excitement at its best. Solve countless puzzles and slay over a dozen monsters by using the huge vocabulary of over 200 words. Two challenging difficulty levels await you with over 80 areas, each fully described in Old English script.

Menacing monsters, kniving villains, tattered code books and treacherous terrain are just a few of the situations you must overcome in your quest for the thirteen priceless treasures. More than seventy objects are invaluable to you in your search for glory and wealth!

A full-size, thoroughly illustrated manual is included. Featuring color front and back, book quality, and a fold-out map, this "extra" further extends the professionalism of this game. The following are quotes from unsolicited testimonials sent to us by adventurers in Gothmog's Lair...

"I have extremely enjoyed Gothmog's Lair, and plan to buy more adventure games..."

Scott Tulman, Memphis, TN

"Gothmog's Lair is the best adventure I've ever played..."

Dennis Manochio, Jr., Saratoga, CA

PROFESSOR List $39.95 Sale $29.95

An in-depth self-tutorial for the Commodore 64 on a two-sided disk. This menu-driven tutorial covers every aspect of your C-64—BASIC, keyboard, sound, music, simple and advanced graphics. Quizzes test your comprehension at the end of many lessons! On-screen illustrations, sound effects and full-color interactive graphics make learning easy and interesting. The PROFESSOR is your ONLY choice for an all-in-one, thorough tutorial about the Commodore 64!

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail!

VISA — MASTER CARD — C.O.D.

No C.O.D. to Canada, APO-FPO.

PROTECTO

ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010

Phone 312/382-5244 to order

MACHINE LANGUAGE FOR BEGINNERS

Richard Mansfield, Senior Editor

Loops And Branches

Branching, looping, and printing messages—these are among the most common computer activities. As you become familiar with machine language, you'll discover how to accomplish everything you can now do in BASIC (and a good deal more). But one of the first things you'll want to know is how to print messages to the screen and to a printer. This month let's explore looping and branching as a method of printing.

In BASIC, it's quite common to set up a loop and then branch out of the loop after a job is done. Here's one such structure which prints DATA statements:

```
10 READ X$
20 IF X$="END"THEN END
30 PRINT X$
40 GOTO 10
50 DATA SEND,THIS,MESSAGE,END
```

Here's the same thing in machine language:

```
10 *= 864
20 .P
30 .S
40 .O
50 ;   READ DATA FROM TABLE AND THEN BRANCH
60 ;
70 LDY #0; INITIALIZE INDEX
80 ;
90 LOOP LDA TABLE,Y
100 BEQ END
110 JSR $FFD2
120 INY
130 JMP LOOP
140 ;
150 END RTS
160 ;
170 TABLE .BYTE "SEND THIS MESSAGE":.BYTE 0
```

There are several things to notice here. We're using an assembler program which accepts BASIC-like programming. The general name for the program above is *source code* which, when an assembler assembles it, becomes a runnable, executable ML program (called *object code).* It's like BASIC because you can use line numbers, make remarks, even have multiple statements on a line separated by colons. The first line must contain the starting address, in this case 864.

Pseudo-Ops And Semicolons

Line 20 is a *pseudo-op* (a false op-code) which tells the assembler that you want a printout of the results of the assembly. (A real op-code, like LDA or JSR, is an instruction which your assembler can turn into ML code. A pseudo-op, by contrast, is a command to the assembler to perform some task which assists you in programming, but will not show up as actual, assembled object code.)

The .S pseudo-op in line 30 causes those assembly results to be listed on the screen during assembly and line 40 causes the object program to be stored in RAM memory.

The semicolons are like BASIC REM statements—anything after a semicolon on a line will be ignored. The first actual ML instruction appears in line 70 and sets up the Y register as a counter so we can take each character out of our message table in turn.

In line 90 we start the loop, using Y as an index to load the Accumulator with a character. If it's equal to 0, that's our signal to branch to END and thus RTS (ReTurn from Subroutine) back to BASIC. Notice that this kind of assembler doesn't rely very much on numbers. We're not branching to a specific address, rather to the label *END*. The assembler will calculate the proper address of the label and replace the word END with the correct number so that the BEQ (Branch if EQual) instruction will operate properly.

Anyway, if we're not through with the message, we JSR $FFD2, which is the routine in the VIC and 64 ROM BASIC which prints whatever character is in the Accumulator at the next available location on screen. Then we raise our index (INcrement Y) to point us to the next character in the table and JMP (jump) back up to the start of the loop. JMP is an unconditional branch. It always branches. The several 6502 ML instructions beginning with the letter B (BNE, BEQ, BCC, BPL, etc.) are all *conditional* branches. Sometimes they send you somewhere and sometimes they don't. They are little tests. In this program, the BEQ (equal to zero?) test will fail repeatedly as we step through the message. Finally, we'll LDA (LoaD the Accumulator) with that 0 on line 170

VIC-20/C-64

MODEM

MFJ-1237

$49.95

Tiny 2¾x2¾x1 in.

300 baud
Direct Connect
Originate/Answer • Full Duplex • Carrier detect LED

World's lowest cost modem. High performance Texas Instrument single chip modem design.

Works for both VIC-20 and Commodore 64. Plugs into user's port. Use with single or multi-line phones. Plugs into telephone base.

300 baud. Direct connect. Originate/answer. Full duplex. Carrier detect LED. Crystal controlled.

Powered by computer. Aluminum enclosure.

Includes Basic listing of Terminal Program. Terminal Program available on tape, $4.95 and cartridge, $19.95. Specify VIC-20 or C-64.

Save VIC-20 Cartridge Programs on tape

MFJ-1256

$39.95

Adapter board lets you save VIC-20 cartridge programs on cassette tape and run them using 8K RAM board. Provides cartridge backup, eliminates plugging and unplugging cartridges and turning VIC-20 on and off.

Includes adapter board that plugs into expansion port and software to save and run cartridge programs on cassette tape. Requires 8K RAM board (not included).

RS-232 Interface for VIC-20/C-64

MFJ-1238

$39.95

Provides RS-232 voltage conversion for VIC-20/C-64 serial port. Use RS-232 printers, modems, speech synthesizers and other RS-232 peripherals. Switch reverses transmit/receive lines for DTE or DCE operation. Use as null modem. Standard 25 pin RS-232 connector. Plugs into user's port. Powered by computer. 2¼x2¼ inches.

VIC-20 Capacitance Meter

Measure 100 pf to 100 Mfd. Includes calibration capacitor, software on tape and hardware interface.

MFJ-1258

$29.95

Order from MFJ and try it. If not delighted, return within 30 days for refund (less shipping).

One year unconditional guarantee.

Order yours today. Call toll free 800-647-1800. Charge VISA, MC. Or mail check, money order. Add $4.00 each for shipping and handling.

CALL TOLL FREE . . . 800-647-1800

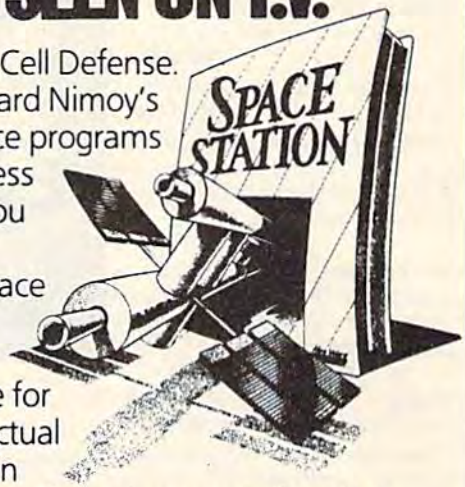
Call 601-323-5869 in MS, outside continental USA.

MFJ ENTERPRISES INCORPORATED

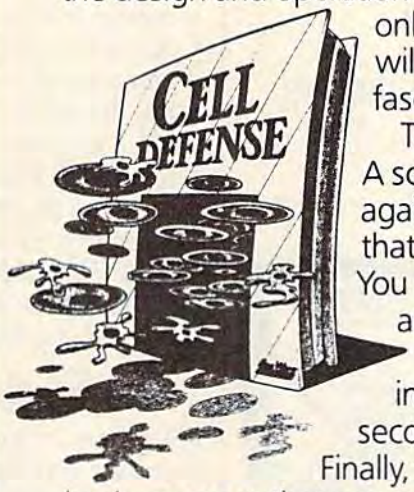
921 Louisville Road, Starkville, MS 39759

WHY LEONARD NIMOY'S NEWEST SCIENCE SERIES WILL NEVER BE SEEN ON T.V.

Project Space Station. Cell Defense. And Calmpute are Leonard Nimoy's favorite computer science programs from HesWare. And unless you have a computer, you may never see them.

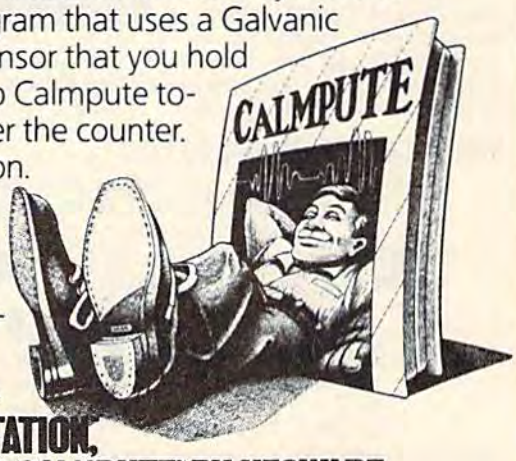


First there's Project Space Station. A program that will put you and your children into deep space for days. It's based on the actual procedures NASA uses in the design and operation of a real space station. Not only will it expand your mind, it will challenge, stimulate, and fascinate you for hours.



Then there's Cell Defense. A science program that pits you against a host of deadly viruses that invade your bloodstream. You must use scientific strategies and basic concepts in biology to protect the cells from invaders. Or in a matter of seconds you'll be dead.

Finally, there's Calmpute. A biofeedback program that accurately monitors your level of stress. And suggests proper exercises to reduce tension. Or increase it, if need be. It's the only medical award winning program that uses a Galvanic Skin Response biosensor that you hold in your hand. Pick up Calmpute today. It's available over the counter. Without a prescription.



HesWare Science Series. They may never be seen on television but the experts give them extremely high ratings.

PROJECT SPACE STATION, CELL DEFENSE™ AND CALMPUTE™ BY HESWARE

Apple II Family and Commodore 64 computers

which signals the end of the message. Then the BEQ test will send us down to END. We'll get to the pseudo-op .BYTE momentarily.

The .P pseudo-op in the program above causes the following information to be printed during the assembly of the source code:

```
50            READ DATA FROM TABLE AND THEN
   BRANCH
60
70 0360 A0 00              LDY #0  INITIALIZE
   INDEX
80
90 0362 B9 6F 03    LOOP   LDA TABLE,Y
100 0365 F0 07             BEQ END
110 0367 20 D2 FF          JSR $FFD2
120 036A C8                INY
130 036B 4C 62 03          JMP LOOP
140
150 036E 60         END    RTS
160
170 036F TABLE .BYTE "SEND THIS MESSAGE
170 0380 .BYTE 0
```

It's similar to the source code, but something's been added. Look at line 70. Now, in addition to the mnemonic LDY and its argument #0, there's also the result of assembling that mnemonic/argument pair: A0 00. The number A0 is hexadecimal (called *hex*, it's a more convenient number system for working in ML). It's the same as 160 decimal. That's a number the 6502 chip understands to mean LDY #. The 00 is hex for 0, the value we want to load into the Y register. So, A0 00 is what will appear, after assembly, in addresses 864–865 in RAM memory (0360 is hex for 864, 0361 is hex for 865). In other words, this is a printout which includes the *object code*, the runnable ML program.

Automatic Assembly

See how in line 130 the address 0362 (in reverse order, as our microprocessor chip wants it) has now replaced the label *LOOP*? If all this is a bit confusing to you at first, get hold of a good assembler and start playing around with it. Much of what we're discussing will be automatically performed for you by the assembler itself.

Also notice the pseudo-op .BYTE in line 170. It allows you to enter literal ASCII code letters (using a quote the way BASIC defines strings) or literal numbers (no quote). This is the fastest way to set up data tables or messages in ML programming. You just give the line a name (*TABLE* in this example), announce a .BYTE series, and then write in whatever data you want. The printout loop technique we're using here signals the end-of-message with the number 0. So, .BYTE 0 sticks in a zero into RAM memory following the letter *e* in the word *message*. (The zero has to be outside of the quotation marks in a separate .BYTE statement since we're after the *numeric* 0, not the *character* 0.)

Before showing how to redirect messages to a printer, let's first explain what all these labels are doing. How can words substitute for numbers? After all, we want to JMP to address 0362 (866 decimal). Why not just write JMP 0362 and be done with it?

One significant advantage of using labels is that you can then freely modify your program without having to change all of the specific references. For example, suppose you write the following:

```
864 LDA 15
866 BEQ 869
868 INY
869 RTS
```

This would work fine because that branch to 869 is correct. But what if you later modified this program by inserting another INY? Or what if you deleted something between the branch and its target? The branch would still be to address 869 but that would be wrong. If, instead, you give the RTS a label:

```
864 LDA 15
866 BEQ FINISH
868 INY
869 FINISH RTS
```

you can change the other parts of this program as much as you want and the assembler will always make sure that the BEQ is correctly sent to the address of the RTS.

Other Advantages

With labels you can quickly modify programs in other ways, too. Let's imagine that you write a large game program and you frequently reset the background color. You have used #6 as your color in many places within the program. You've got LDA #6 sprinkled all over the place. If you later change your mind and want to use #5 instead of #6, you would have to locate every place where #6 appeared and change it to #5. If you had simply assigned a label at the start of the program: COLOR1 = 6 and then always used LDA #COLOR1, you could just change that first label assignment to COLOR1 = 5. All references to COLOR1 throughout the program would then automatically change as well.

These and other advantages of labels all contribute to a BASIC-like environment which can make sophisticated ML programming efficient and comfortable for the programmer. But let's now turn to the way you can communicate with your printer in ML. There are two things to do. Open a file (OPEN 4,4 is the BASIC equivalent) and then print a character to file #4 (PRINT#4,A$ in BASIC).

To open the channel of communication to

the printer, you use the following protocol:

```
10 OPEN4 LDA #4; FILE NUMBER
20 LDX #4;       DEVICE NUMBER
30 LDY #0;       SECONDARY ADDRESS
40 JSR $FFBA;    KERNAL SETLFS(SET UP LOG
   ICAL FILE)
50 LDA #0;       LENGTH OF NAME
60 JSR $FFBD;    KERNAL SETNAM (SET NAME)
70 JSR $FFC0;    KERNAL OPEN (OPEN A LOGI
   CAL FILE)
80 JSR $FFCC;    KERNAL CLRCHN (CLEAR I/O
    CHANNELS)
90 RTS
```

This is a subroutine called OPEN4 which can be invoked in your ML program by a JSR OPEN4. It uses the Commodore Kernal routines, which are common to the VIC and 64 (these same locations, with different device numbers, are used to open communications to a disk drive or tape drive). When it returns, it will have reset normal I/O (input/output) conditions for you in line 80. Normal I/O specifies that the screen is the output target and the keyboard is the input source. These conditions prevail unless the computer is notified otherwise. That's where the "chkout" and "chkin" routines come in. They are like PRINT# and INPUT# in BASIC, redirecting the output or input of a given character or symbol from the defaults to other, previously opened, files.

Now, whenever you want to send a character to the printer, you can invoke a PRINT# by LDX #4: JSR 65481, thus opening a channel of communication to the printer (file number four). It's similar to BASIC's CMD. Then, JSR 65490 is the general purpose print routine which sends the character in the accumulator to whatever device lies at the end of the currently open channel. Finally, you close the channel to the printer by JSR 65484 which restores normal I/O conditions.

Once the chkout routine is called, you can use the print routine (65490) to send individual bytes to the printer, one after another. But to be safe, we're going to clear the channel after each byte we send. Below is a subroutine called PRINTER which prints the character in the accumulator to the printer. You must call this subroutine for *each* character you want to print.

Note that you must save the character in the accumulator until you need it. Those ROM routines we JSR to will not preserve the value in the accumulator for you. To accomplish this, we've previously defined a variable called A which can temporarily hold the value of the accumulator until we need it.

Here's the entire PRINTER subroutine:

```
10 PRINTER STA A; SAVE ACCUMULATOR VALUE
20 JSR 65484;     CLEAR CHANNELS (RESTORE
   NORMAL I/O)
30 LDX #4;        PREPARE DEVICE #4
40 JSR 65481;     BY OPENING A CHANNEL TO
   THAT (PREVIOUSLY OPENED) FILE
50 LDA A;         RECOVER ACCUMULATOR VALUE
   TO BE PRINTED
60 JSR 65490;     PRINT
70 JSR 65484;     CLEAR CHANNELS AGAIN
80 RTS;           RETURN
```

Take another look at the first example above which prints PRINT THIS MESSAGE to the screen. To make it print to the printer instead, just replace the JSR $FFD2 in line 110 with a JSR PRINTER (our new subroutine). To have it print on screen and printer simultaneously, leave in the JSR $FFD2 and just add a new line right below it:

```
115 JSR PRINTER
```

Whenever you JSR to routines in ROM, it's a good idea to first save the numbers in the Accumulator and the X and Y registers if you will be wanting to work with these numbers further. ML programmers make heavy use of the registers, and the ROM routines like to use them as well. You cannot be sure that, after a JSR to ROM, you'll get back the same numbers in the registers. Luckily, the general print routine at 65490 ($FFD2) is *non-destructive*. It preserves the A, X, and Y registers. However, some other ROM routines will not offer this courtesy.

Notice that we're using the Y register (in the first example above) to index our printed message. To be on the safe side, set up a variable, called Y or something, and STY Y just as you STA A upon entry to the PRINTER subroutine. If you make a habit of saving important registers before ROM JSRs, you'll avoid one major source of ML program bugs.

To pull it all together, here are the steps you take to print a character to the printer. First you would JSR OPEN4 (to open a file to the printer), then load the character you want printed into the Accumulator, and finally JSR PRINTER to print the character. This seems like a struggle to print only one character, but once you've set up the subroutine, you simply put any character into the Accumulator and JSR.

At the end of an ML program, you'll want to close files and shut down communications to peripherals. Here's the way to close down our file to the printer:

```
500 JSR 65484;      SHUT DOWN PRINTER,
   GRACEFULLY.
510 LDX #4
520 JSR 65481;      OPEN PRINTER CHANNEL
530 LDA #13;        LOAD A CARRIAGE RETURN
   CHARACTER
540 JSR $FFD2;      PRINT IT
550 JSR 65484;      CLEAR ALL CHANNELS AGAIN
   (RESTORE NORMAL I/O)
560 LDA #4
570 JSR $FFC3;      CLOSE THE FILE
```

COMMODORE 64 GETS AWAY FROM BASICs
with

THE *Pascal Compiler* FOR THE COMMODORE 64™

Limbic Systems, Inc. introduces the PASCAL COMPILER from Oxford Computer Systems (Software), Ltd., developers of *PETSPEED -- the BASIC compiler recommended by Commodore.

THE PASCAL COMPILER OFFERS --

- **EASE OF OPERATION**
  Pascal, a programming language, simplifies development of software for customized applications
- **POWER**
  to write programs efficiently and effectively
- **TEACHING TOOL**
  Pascal is the preferred language to teach programming skills
- **EFFICIENCY**
  debugging time is minimized -- the major task in program development
- **FLEXIBILITY**
  utilities are provided as an aid in programming development

THE PASCAL COMPILER IS REASONABLY PRICED!

Limbic Systems Inc.

560 San Antonio Road, Suite 202 Palo Alto, CA 94306
(415) 424-0168

Commodore 64 is a trademark of Commodore Electronics, Inc.
**PETSPEED is a trademark of Oxford Computer Systems (Software), Ltd.*

SOME OF THIS YEAR'S BEST HIGH-DIVE PERFORMANCES WILL TAKE PLACE ON A KITCHEN TABLE.

Perform a double sumersault dive. Lift 600 pounds. Run the 500 yard dash. Shoot a bow and arrow. All on your kitchen table. All with HesGames.™ With HesGames' six action packed sports games on one disk you can break all sorts of records without ever really breaking a sweat. With very colorful, realistic characters, cheering crowds, a judged performance, quick instant replay capability, and even a way to save all your best performances, you can experience the thrill of victory or the agony of defeat. All on your kitchen table. Try HesGames now, and we'll give you your very own free HesGames t-shirt.

HESGAMES BY HESWARE
For the Commodore 64. Coming soon for the Apple II.

Teaching Your Computer English

Michael A. Long

Would you like to talk to your computer? With the string handling techniques described in this article, it may seem like you are. For the VIC and 64.

The basic idea of an adventure game like *Zork* is that you play a character in a story. The computer describes the surroundings, and you decide what to do—search the room, listen at the door, fight the dragon, drink a magic potion, and so on. Your choices determine the direction of the story, which is why adventure games are sometimes called interactive fiction.

The heart of such a game is the *parser routine*, which splits apart the sentences you type and matches the words against a vocabulary list. Parsing is a computer's way of diagramming sentences.

Some adventure games include lists of acceptable but hard-to-understand commands. For instance, you may have to use G for get, + for up, and − for down. However, these should not be your only options when operating text adventures or other interactive programs. There are several techniques which allow the user to type in English commands such as *go north* or *open door*. One of the most useful of these techniques looks only at the first few letters of each word. Let's study the program for a simple six-room adventure game to demonstrate how it works.

Although the program will run on any VIC or 64, it will almost completely fill an unexpanded VIC, so type the program in carefully and without extra spaces if you're using a VIC without expansion.

If you have the inclination (and enough memory), you might want to add new commands, new treasure, and new rooms to this mini-adventure game.

Normal English Commands

This program demonstrates a technique used by many adventure programmers. It allows the user to input a command in normal English sentences. The only restriction is that the user must type in the verb first and the object (noun or compass direction) last. Only a few of the letters in the first and last word are actually read by the computer.

Lines 12–17 take care of screen printing and the input prompt. The user types in a sentence, and the computer stores it in variable A$. Lines 18 and 19 read through A$ using a FOR-NEXT loop and the MID$ function, looking for the space between the first and second words. Then the computer puts the two letters following that space into B$. At that point it is finished with the last part of the sentence and can discard it, a feat accomplished in line 20 by setting A$ equal to the first two letters of the first word in the sentence.

But the computer still can't use the values because they're letters rather than numbers. The variable V$, established in line 21, holds the first two letters of all the verbs you may use in the program, and line 24 does the actual numeric conversion. It sets up a FOR-NEXT loop in which the variable X increments from one to the length of V$. The computer uses the MID$ function to look at the two letters following X until it finds a match for the value in A$. When it finds a match, it then sets V equal to the number of the count, plus one, and then divides V by two. At that point you have the number of the verb you typed in.

Line 24 does the same thing, except that it is looking for a match in N$ (contained in line 22) for the B$ string. Lines 71–75 contain a list of all the nouns (N$).

You now have V equal to the number of the verb and N equal to the number of the noun. Suddenly, we're working with numbers, which are easier for your computer to manipulate.

Line 25 uses an ON-GOTO statement to send the program to different sections of the listing depending on which verb was used. For example, if you used the verb GO, the program would be sent to line 26. From there all you have to do is set up the logic needed to execute the command you want.

Line 69 is a DATA statement which holds the information for array M, the movement map for moving from room to room in the adventure. Lines 80–81 contain the information for R$, which holds the room names in the

program. Lines 77–78 are DATA statements for array L, the locations of all the objects in the adventure.

One more note. If you have two verbs with the same beginning letters (like PUT and PULL), you can do one of two things to ensure proper selection of the verb you want. You can read more than the first two letters of the verb, but you'll sacrifice some speed (and use a lot more memory).

A better alternative is to direct both commands to the same location in the program. When those verbs are used, the computer searches through the appropriate program lines and reads only the commands you put in for that situation. That way, you only have to store the first two letters in V$ one time, and you don't have to use another GOTO command with your ON statement either.

You can use a similar method to handle two verbs that mean the same thing but are spelled differently. For example, both *get* and *take* could send you to the same part of the program if you repeat the GOTO number in the ON statement.

Here's a list of verbs used in our example program:

GO = Go
GE = Get
PU = Put and Pull
TA = Take
HI = Hit
IN = Inventory
LO = Look

If you'd rather not type in this program, send $3, a *blank* cassette (no disks, please), and a self-addressed stamped mailer, and I'll be glad to make you a copy.

Michael A. Long
6640-B 105th St.
Ewa Beach, HI 96706

See program listing on page 171.

Program Variables

A$ = Inputed sentence, and first two letters of verb
B$ = First two letters of noun
V$ = Holds first two letters of all the verbs allowed
N$ = Holds first two letters of all nouns used
N$0 = Holds list of all the nouns for use by the program
M0 = Two-dimensional array for use as movement map
L0 = Location for all objects (nouns)
L = The room you are currently in
L5 = Holds screen and border memory location for VIC; used to change screen colors when changing rooms.
R$0 = Room names
Z = Used for Get statements
V = Numeric value of the verb
N = Numeric value of the noun

Hi-Res Screen Dump

Gregg Peele, Assistant Programming Supervisor

Have you ever created a hi-res picture or graph and then tried to reproduce it on your printer? This program allows you to do just that. The VIC version requires a Super Expander cartridge. Both VIC and 64 versions are compatible with the Commodore 1525 or MPS-801 printers (but not the 1526).

Both the VIC-20 and Commodore 64 allow you to create high-resolution graphics images on the video screen. With the VIC or 64 Super Expander cartridge or another hi-res program, it's easy to produce detailed artistic creations. However, most of these programs don't provide a method of printing out these artistic endeavors once you've finished them. Unless you leave your computer turned on indefinitely, your creation is short-lived.

Both versions of "Hi-Res Screen Dump" work with a Commodore 1525 or compatible printer. (Note that the new 1526 printer from Commodore is *not* compatible with the 1525, and will not work with this program.)

Bit Transfer

Hi-Res Screen Dump is designed to transfer the bit information from screen memory to the printer. Since the 1525 printer can only accept seven bits of data at a time in graphics mode (the high bit must always be set), the eight-bit bytes in screen memory must be split into odd units before they are sent to the printer. Transferring the information from screen to printer is further complicated since the location of screen memory bytes must also be calculated, and hi-res screens for the 64 can be moved to several different areas of memory.

This program reads data from the screen one bit at a time starting from the lower leftmost corner of the screen. After seven bits, the program moves to the leftmost bit of the next row up and prints seven more bits, continuing up the screen. After the leftmost seven-bit column has been printed, the program starts at the eighth bit over from the bottom left corner and continues cycling from bottom to top until the entire screen has been read. Each seven bits are combined to form the byte to be sent out to the printer. Since the program reads from the left bottom side of the screen to the right top side, the printout is a 90-degree-turned reproduction of the screen image.

Both versions of Hi-Res Screen Dump are written in machine language. A BASIC loader (the first several lines of the program) puts the machine language (in the form of DATA statements) into the appropriate locations in memory. The BASIC loader also prompts you for the width of the printout, and the VIC version protects the machine language at the top of free memory. To operate the program correctly, *you must load and run Hi-Res Screen Dump before you load the program which creates the hi-res image.*

Selecting A Width And Making A Printout

In both versions, you can select either a single-width or double-width printout by POKEing a 1 (for single width) or a 2 (for double width) into location 2 (i.e., POKE 2,1 or POKE 2,2). This location is changed by your selection of width when you are prompted in the BASIC program, but can be changed at any time. A SYS to location 3584 (for the VIC) or 52224 (for the 64) will initiate a printout of the hi-res screen. You can issue this SYS in direct mode if you have a design on the screen, or add it to a hi-res drawing program if you make sure the machine language is loaded into memory before the SYS is encountered. Also, be sure that the printer is turned on before giving the SYS.

ALL THE BEST PRICES

commodore

M-801 Dot Matrix Parallel	$219.00
MCS 801 Color Printer	$499.00
1520 Color Printer/Plotter	$129.00
1530 Datasette	$69.99
1541 Disk Drive	$249.00
1600 Vic Modem	$59.99
1610 Vic Term 40	$59.99
1650 Auto Modem	$89.99
1702 Color Monitor	$249.00
DPS Daisywheel Printer	$459.00
Desk Organizer Lock	$49.99
1311 Joystick...each	$4.99
1312 Paddles	$11.99
1110 VIC 8K	$42.99
1111 VIC 16K	$69.99
IEEE/RS-232 Interface	$42.99
1211 Super Expander	$53.99

MCD

SD 1 Disk Drive	$359.00
SD 2 Disk Drive	$589.00

CARDCO

Light Pen	$32.99
3 Slot VIC Expansion Interface	$32.00
6 Slot Expansion Interface	$79.99
Cassette Interface	$29.99
Parallel Printer Interface	$49.99
Parallel Interface w/Graphics	$69.99

PRINTERS

C. ITOH

Gorilla Banana	$149.00
Prowriter 8510P	$379.00
Prowriter 1550P	$599.00
A10 (18 cps) Son of Starwriter	$569.00
F10-40 Starwritter	$999.00
F10-55 Printmaster	$1349.00

SX-64 PORTABLE

$839

Call for CBM 4 plus

Commodore 64	**$199**
VIC 20	**CALL**

COMREX

ComWriter II Letter Quality	$449.00

EPSON

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

JUKI

6100	$469.00

MANNESMAN TALLY

160L	$699.00
180L	$799.00
Spirit 80	$289.00

NEC

8027 Dot Matrix	$389.00

OKIDATA

82, 83, 84, 92, 93, 2350, 2410	$CALL

OLYMPIA

Compact 2	$479.00
Compact RO	$609.00
ESW 3000	$1449.00

SMITH CORONA

TP 1000	$449.00

SILVER REED

500 Letter Quality	$449.00
550 Letter Quality	$549.00
770 Letter Quality	$869.00

TOSHIBA

1340	$629.00
1351	$1449.00

TRANSTAR

120P	$469.00
130P	$649.00
315 Color	$459.00

MONITORS

AMDEK

300 Green	$149.00
300 Amber	$159.00
Color I	$259.00
Color I Plus	$289.00
New Color 300	CALL

BMC

1201 (12" Green)	$88.99
1201 Plus (12" Green Hi Res)	$98.99
9191 Plus Color	$249.00

GORILLA

122" Green	$88.99
12" Amber	$95.99

NEC

JB 1201 Green	$149.99
JB 1205 Amber	$159.99
JB 1215 Color	$259.00

SAKATA

SC-100 Color	$269.00
SG-1000 Green	$129.00
SG-1000 Amber	$139.00

TAXAN

100 12" Green	$125.00
105 12" Amber	$119.00

USI

Pi 1, 9" Green	$99.99
Pi 2, 12" Green	$119.99
Pi 3, 12" Amber	$129.00
Pi 4, 9" Amber	$119.99
1400 Color	$269.99

ZENITH

ZVM122 Amber	$99.99
ZVM123 Green	$89.99

SOFTWARE

ATARISOFT (C-64/VIC)

Dig Dug	$34.99
Defender	$34.99
Robotron 2084	$34.99
PacMan	$34.99
Donkey Kong	$34.99
Centipede	$34.99
Stargate	$34.99

BRODERBUND (64)

Choplifter	$29.99
David's Midnight Magic	$22.99
Bank Street Writer	$49.99

COMMODORE (64)

C-64 Reference Guide	$18.99
EasyCalc	$65.00
Easy Finance I, II, III, IV	$19.99
Easy Mail	$19.99
Easy Script	$39.99

CREATIVE SOFTWARE (VIC)

Apple Panic	$19.99
Black Hole	$22.99
Home Inventory (C-64/VIC)	$11.00

CBS (64)

Addition & Subtraction	$16.00
Linear Equation	$16.99
Multiplication & Division	$16.99
Quadratic Equations	$16.99

CONTINENTAL SOFTWARE (64)

The Home Accountant	$49.99

CSA (64)

PractiCalc	$39.99
PractiFile	$39.99

DESIGNWARE (64)

Cryto Club	$29.99
Trap-a-Zoid	$29.99

DYNATECH (64)

Codewriter	$75.99

ELECTRONIC ARTS (64)

Pinball Construction	$29.99
Cut & Paste	$39.99
Hard Hat Mack	$75.99

EPYX (6-64/VIC)

Temple of Apshai	$29.99
Upper Reaches of Apshai	$16.99
Jumpman Junior	$29.99

HES (64)

Tri math	$22.99
The Pit	$27.99
Ghost Manor	$15.99
Pool Challenge	$19.99
Hes Mon	$29.99
Hes Writer	$35.99

INFOCOM (64)

Zork I, II, III	$27.99
Deadline	$29.99
Witness	$29.99

PROFESSIONAL SOFTWARE (64)

Word Pro 64 plus Spell	$59.99

SEGA (64)

Star Trek	$29.99
Congo Bongo	$29.99

SPINNAKER (64)

Snooper Troops 1 or 2	$29.99
Delta Drawing	$29.99
Kids on Keys	$29.99

SCREENPLAY (64)

Wylde	$22.99
Kaiv	$22.99
Pogo Joe	$20.90

SUB LOGIC (64)

Flight Simulator II	$40.99

SYNAPSE (64)

Zaxxon	$29.99
Protector II	$23.99
Blue Max	$24.99

TIMEWORKS (64)

Wall Street Manager	$19.99
Word Writer	$39.99
Data Manager	$19.99

COMPUTER MAIL ORDER

VISA — MasterCard

west
800-648-3311
In NV call (702)588-5654, Dept. 115
Order Status Number: 588-5654
P.O.Box 6689
Stateline, NV 89449

canada
Ontario/Quebec 800-268-3974
Other Provinces800-268-4559
In Toronto call (416)828-0866, Dept. 115
Order Status Number: 828-0866
2506 Dunwin Drive, Unit 3B
Mississauga, Ontario, Canada L5L1T1

east
800-233-8950
In PA call (717)327-9575, Dept. 115
Order Status Number: 327-9576
Customer Service Number: 327-1450
477 E. 3rd St., Williamsport, PA 17701

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

Configuring The VIC

The VIC version of Hi-Res Screen Dump (Program 1) is designed to be used with the Super Expander cartridge. The GRAPHIC 2 command provided by the Super Expander sets up a 3200-byte high-resolution screen beginning at location 4096 ($1000), while maintaining the normal text screen beginning at location 7680 ($1E00). With the 3K of additional RAM provided by the Super Expander, BASIC program storage begins at location 1024 ($0400). Line 5 of the loader program reserves two pages (512 bytes) of memory for the machine language, just below the area to be used by the hi-res screen. Thus, the machine language starts at location 3584 ($0E00), and locations 1024-3583 ($0400-$0DFF) are available to BASIC programs.

However, using the VIC Super Expander alone leaves only 2-1/2K free for your BASIC programs. If you have an expander or motherboard that will allow you to add 8K or more of extra RAM in *addition* to the Super Expander, then you'll have much more room for BASIC programs. The machine language will still load into the 3K RAM area provided in the Super Expander, and SYS 3584 will still initiate the dump. This area is now untouched by BASIC, so be sure to remove line 5 before you attempt to run Program 1 in this configuration; otherwise you will get an ?OUT OF MEMORY message. The hi-res screen set up by the GRAPHIC 2 command is still 3200 bytes beginning at location 4096 ($1000), but BASIC programs will now start at location 8192 ($2000) and continue to the end of installed memory.

The 64 Version

The machine language for the 64 version (Program 2) resides at the top of the 64's free block of RAM above location 49152 ($C000). This makes it compatible with the *Super Expander 64*, but also means that it cannot be used with the 64 DOS wedge program, as both occupy the same area of memory. The program is designed to print the hi-res screen that is currently visible. If you want a screen dump when you are not in hi-res mode, POKE location 900 with the high byte of the starting address of the hi-res screen and SYS to location 52224+32. This alternate SYS bypasses the routine which determines the location of the hi-res screen. For example, if your hi-res screen starts at location 57344 ($E000)—as in "Screen-80" from the September issue—you would initiate the screen dump with:

```
POKE 900,(57344/256): SYS 52256
```

See program listings on page 180.

Robert Sims, Assistant Editor

All The Fun That's Fit To Print

For even the most dedicated computer owner, there comes a time in the serious and brow-furrowing process of learning to telecommunicate when what is needed most is a good, hearty laugh.

Most of the bulletin boards and services on the communications networks do not focus on humor (the Artsig on CompuServe and almost any special interest group on Delphi excepted). It is often the local bulletin board which is the repository of fun and of the absurd.

A Perpetual Party

Experienced modem owners use their local bulletin boards as a kind of perpetual party of the mind, a conversational free-for-all with everybody dropping in and out willy-nilly. Occasionally, a sysop will see the sense in all this nonsense and try to bring a touch of order into the chaos. A good example of this organized hilarity is *The Modem Times*, an electronic magazine originating in Colorado Springs. The magazine, brainchild of editors Jennifer Petkus and James Bates, is one of a kind in electronic publishing.

The editors don't seem to be sure of exactly what they've wrought—a bulletin board, a literary magazine, or a no-holds-barred forum of opinion for their subscribers. If they ever adopt a slogan for their masthead, it might well be "All the fun that's fit to print."

The Jocular Vein

The *Times* is structured much like a conventional print newspaper or magazine, with a table of contents, editorial comments, letters to the editor, features, and fiction from very obscure authors. There the similarity ends.

From the ersatz history of the publication (in which it is revealed that ancient astronauts founded the magazine along with the Egyptian and Incan civilizations), to the apologia for its editorial stance ("If you think our editorial page is offensive, send $20 and some suggestions"), *The Modem Times* is dedicated to the proposition that there is not a serious bone in the entire body politic.

A Comic's Dream

The Modem Times and its electronic kin add a new dimension—participatory zaniness. For the first time in the history of the funnybone, audience and performer are essentially indistinguishable.

Any subscriber can rise from the ranks to become an instant type-in comedian, uploading a joke and leaving before the audience begins to laugh or throw overripe vegetation. It's the best of all possible worlds for a comedian: You get your chance in the spotlight, but you're never there when you bomb.

The Modem Times is not all jocularity, of course. It provides a creative outlet for writers of short fiction and poetry. The magazine even has a section, "The Modem Times Won't Be Bought," set aside just for soap-box speeches and tirades on any subject. Here subscribers are encouraged to engage in polemic to their hearts' content.

The Laugh Stops Here

Unfortunately, navigating the magazine is not as much fun as its contents. The software that runs the magazine is adapted from an ordinary bulletin board, with a long list of commands to memorize, and a main menu containing 24 options.

A help file and explanation of commands and options are readily available on-line, and the commands will be familiar to experienced bulletin board users. But the beginner will do well

to first download and print out the help file (Page 13) before venturing further. It's difficult to maintain a literary frame of mind if you have to keep jumping to the instructions between pages.

A Common Problem

This navigation problem is not unique to *The Modem Times*, of course. Whenever a program is written to allow a specific computer to communicate with a variety of terminal programs on different computer models, it is necessary to use the ASCII control codes and character set so the computers will have a common language.

Two of the ASCII control codes are standard in almost all telecommunications situations. CTRL-S, a CHR$(19), and CTRL-Q, a CHR$(17), are recognized by most terminal software to mean "stop transmitting" and "resume transmitting," respectively.

Such standard codes can be transmitted automatically. But the meanings of other commands and control codes vary widely among bulletin boards and terminal programs. Consequently, the user must learn the necessary commands or codes and transmit them manually.

More than any other aspect of home telecommunications, electronic publishing points out the need for integrated software. In its infancy, we expect to make the best of what's available. But as electronic publishing matures, we can expect to find fewer technical distractions.

Forgettable Software

The reader is after entertainment, not programming tips or technical information. The software serves no purpose but to connect the reader and the magazine, and it should be transparent. In addition, it should be comprehensive. Instead of targeting the lowest common denominator, ASCII, electronic publishing software needs to be sophisticated enough to exploit the best sound and graphics features of each computer model by exchanging codes with the terminal program to discover the caller's computer type and automatically set transmission rate, display width, and other parameters.

In this ideal world of the possible future, each subscriber will receive a dedicated terminal program which will require no more technical expertise than the ability to connect the modem and load and run the program.

But until that millenium arrives, we'll have to be content to log onto *The Modem Times* and deliver a tirade on the subject.

The Modem Times
15 N. 14th Street
Colorado Springs, CO 80904
(303) 471-2126

SIMPLE ANSWERS TO COMMON QUESTIONS

Tom R. Halfhill
Staff Editor

Q A

Each month, COMPUTE!'s GAZETTE *tackles some questions commonly asked by new Commodore 64/VIC-20 users and by people shopping for their first home computer.*

Q. ***I'm confused! Your instructions concerning powering up the 1541 disk drive indicate that the computer should be turned on first ["Simple Answers To Common Questions," August 1984]. My* VIC-1541 User's Manual *(published 1982) at the top of page six says the following: "It is important that you turn on the devices in the correct order. The computer should always be turned on* last. *As long as the computer is the last one to be turned on, everything will be OK." Which is correct?***

A. We've received a couple of similar letters quoting from that paragraph in the Commodore manual. However, there should be no cause for confusion. In practice, the order in which the computer and disk drive are switched on almost never matters.

This is particularly true of one-drive systems without a printer. Occasionally we've heard of problems when certain printers are part of the daisy chain (the chain of peripherals plugged into each other). Sometimes the computer locks up when such a system is turned on in the wrong order. The Commodore 1526 printer had this problem, but it was temporarily recalled by Commodore because of internal bugs. Most printers cause no difficulties when hooked into the daisy chain.

We've also heard that multi-drive Commodore 64 systems can be sensitive to the power-on sequence. (Of course, keep in mind that any multi-drive system will lock up if the device numbers of the additional disk drives aren't changed; otherwise all the drives will contend for the computer's attention at once.) Interestingly, about a year ago Commodore issued a technical bulletin to dealers recommending the proper way to switch on various Commodore 64 systems. The bulletin stated the exact opposite of what the *VIC-1541 User's Manual* says—the computer should be turned on *first*, not last. Here are the power-on sequences that the bulletin recommended for various systems with 1541 disk drives and Commodore printers:

1) **64, 1541, 1525E.**
2) **64, 1541, 1541.**
3) **64, 1541, 1541 or 1525E** (only one or the other may be on).
4) **64, 1541, 1541, 1526.**

But again, let us repeat: With one-drive systems, in our experience it really makes no difference which device is turned on first.

Why? Partly because of the unique way Commodore handles its Disk Operating System (DOS). Briefly, DOS is a program which allows a computer to interact with a disk drive. Without DOS, a computer and disk drive couldn't communicate. Therefore, all computers hooked up to disk drives require some type of DOS.

Commodore's DOS is permanently stored inside the disk drive on a Read Only Memory (ROM) chip—which is why the 1541 is sometimes called an *intelligent drive*. But with Apple, Atari, IBM, and almost all other systems, DOS is stored as a program file on a floppy disk. DOS must be loaded from this disk into the computer's memory each time the system is turned on. This process is called *booting up*, because in a sense the computer is pulling itself up by its own bootstraps.

For example, with Atari computers you must turn on the disk drive before the computer and wait a few seconds for the drive to initialize, just like a 1541. But next you must insert a disk that contains the DOS program—called a *DOS disk* or *system disk*—and only then

switch on the computer. The first thing the computer does when it "wakes up" is load the DOS program into memory. Then it's ready. The procedure for booting up an Apple or IBM system is very similar, except you don't have to switch on the disk drive first because it's powered by the computer.

There are three main advantages to Commodore's method of storing DOS on a ROM chip inside the disk drive instead of loading it off disk. First, you don't have to wait around for the system to boot up. You just switch it on and it's ready to go. Second, you don't have to fool around with system disks. Because DOS can take up a fair amount of space on a disk, many people leave it off most of their disks to make more room for other files. But then they have to shuffle through piles of disks to find a system disk whenever they have to boot up. And third, loading DOS into the computer consumes valuable Random Access Memory (RAM). This can amount to about 10K on an Apple, 5–11K on an Atari, and 24K on an IBM PC/PCjr. Naturally, this leaves less room for your other programs.

As usual, though, there are tradeoffs. Permanently storing DOS on a ROM chip has some disadvantages. The main drawback is that the manufacturer can't revise DOS without making a new ROM chip. Let's say a serious bug is discovered in Commodore DOS. To fix it, Commodore would have to halt production on the old ROM chips, modify the DOS program, burn a new master chip, start production on the revised ROMs, and finally substitute them for the old ROMs on the assembly line. Not only would this take weeks and cost thousands of dollars, it could also trigger a supply shortage while the drives were out of production, resulting in lost revenue and unhappy customers. Besides that, people who already bought a disk drive would still be stuck with the faulty version of DOS. To fix it, they'd have to get the revised ROM chip, take apart the disk drive, pry out the old ROM, and replace it with the new one. Consequently, bugs which crop up in Commodore DOS usually go uncorrected—at least until the next model disk drive is introduced.

When DOS is stored on disk, the process is a little easier. The manufacturer simply revises the DOS program, prepares a fresh master disk, and saves the new DOS onto all system disks duplicated from that day onward. Updates and improvements to DOS can be made just as easily. To avoid confusion, the revision number of DOS is advanced a notch.

HORIZONS: 64

Charles Brannon
Program Editor

Following my review of Simon's BASIC in this column, several readers responded. Not all agreed with some of my points. I stated that some of the commands are a little unnecessary, such as CENTRE A$ instead of PRINT TAB(19-LEN(A$)/2);A$. Many readers are glad to have these extra commands. For example, LEFT and RIGHT, which scroll the screen left and right, seem pretty specialized, useful only for screen scrolling, but some readers have used these commands in new, ingenious ways. Some folks even convert BASIC programs from the GAZETTE into Simon's BASIC for speed and special effects.

However, some readers also lament that they cannot share their Simon's BASIC programs with friends who lack the cartridge. Unfortunately, it's unavoidable. With enough machine language, though, you can convert a Simon's BASIC program into a BASIC program everyone can use. We prefer this approach, since it makes the program available to the widest possible audience. There is still no indication that there are enough owners of Simon's BASIC to justify the publishing of programs in Simon's BASIC—programs that would be useless to the majority of our readers. This may change, though, so we'll keep you posted.

It turns out that some versions of Simon's BASIC do not work on some models of the 64. Reader John Walker bought Simon's BASIC and found that it would not run on his machine, but worked fine on a friend's. He went to a local computer store and was told to PRINT PEEK(65408). This returns a number that you can use to distinguish between the various 64 Kernal ROM versions. He was told if PEEK(65408) returns a 0 or a 3, then Simon's BASIC will work on your machine, but a 170 is bad news for Simon's BASIC enthusiasts. Mr. Walker wrote Commodore and received a new Simon's BASIC cartridge that works fine on his computer. Commodore itself said that the problem is in the cartridge, not which version of the 64 you own. And Commodore seems to have fixed the problem.

Kernal Knowledge

We called Commodore and found that there are at least three versions of the Commodore operating system, a.k.a. the Kernal. The third and latest version has been around since the beginning of 1984. The SX-64 has a fourth, somewhat more radically changed, Kernal ROM. The genealogy of the 64 Kernal ROM is covered in detail in "Commodore 64 ROM Generations," by Jim Butterfield, in the July 1984 issue of COMPUTE!. We'll paraphrase some of it here for those of you who missed that issue. The differences between ROM versions are minor. In revision one, color memory is always filled with white when you clear the screen. Revision two fills color memory with the current background color. What this means is that if you POKE to screen memory without also POKEing a color byte (for example, press RUN/STOP–RESTORE and then POKE 1024,1), you will see a white character on revision one, but on version two the character is invisible. This was an attempt to cosmetically eliminate the sparkle problem, which plagued the earlier machines.

On the SX-64 portable, when the screen is cleared, color memory is filled to the current cursor color, found at location 646. Programs which do not explicitly set color memory, or take advantage of the effect upon screen memory when you clear the screen, can give strange effects when running on a different version of ROM. It doesn't make sense to count on certain undocumented side effects resulting from Kernal calls, since there is no guarantee that these side effects will be preserved on future ROM generations.

The other significant change is that when you load a program from tape on a revision one 64, the computer waits forever after displaying "FOUND program name" until you press a key. The screen blanks, then the load proceeds. On revision two, the 64 waits a few seconds for you to read what it has found, then it goes ahead and loads the program.

When Commodore first marketed the 64, there were separate ROMs for the U.S. and European markets. In the United States, the NTSC standard is used for television broadcasts. Since the RF modulator in your 64 is essentially a tiny television station, the 64 has to conform to the local television standard.

A Universal Operating System

The European television standard, called PAL, has 625 television scan lines, versus 525 for NTSC. This results in a better quality picture. A NTSC screen has a 60 Hz rate (60 frames a second), while PAL uses a 50 Hz rate. Since the computer has to synchronize itself with the television display, the clock crystal runs at a different speed on European 64s than it does on U.S. models. The ROMs must also synchronize themselves with the proper speed, so that the software-supported RS-232 interface will run at the proper baud rate.

Instead of producing separate ROM sets, Commodore programmed a "universal" Kernal that determines which TV standard the hardware conforms to. A raster scan is set to interrupt on scan line 622. Since there is no scan line 622 on U.S. televisions, the interrupt doesn't happen, and the 64 assumes the NTSC standard. The ROMs then make the necessary software adjustments. Jim Butterfield warns that you can't count on the raster register to hold a zero value, since on this ROM it has already been used.

The various Kernal ROMs can cause some programs to be incompatible with other 64s, but it is easy to write a program that is ROM independent. To paraphrase the Commodore representative we talked to, "We don't like to emphasize the differences between Kernal ROMs. You get users who can't format a disk and then blame it on which Kernal they have." The point is well taken. As a programmer, many times I've wanted to blame the hardware for a bug in my program, but sooner or later, I find my mistake. Blaming the hardware should always be the last resort. Too many people have returned portable radios for repair when the batteries have been installed incorrectly.

Problems With CP/M, Too

With that in mind, there does seem to be a hard ware-related problem with both Simon's BASIC and the Commodore CP/M cartridge. A large New York users group, which has "undertaken the mammoth project" of converting the best public-domain CP/M software to 1541 format, has suggested that CP/M will not work on some newer model 64s. They are requesting feedback from CP/M owners to help identify the problem. If you want to aid them, they need to know: Do you have a working 64 CP/M system? If not, describe the problem. What number does PEEK(65408) return? What is your 64's serial number? Are the back slots silver or copper? If you want further information on their CP/M project, or wish to aid them in identifying the CP/M problem, write to:

NYC VIC-20/C64 User Group (CP/M SIG)
c/o NYACC
P.O. Box 106
Church Street Station
New York, NY 10008

Reader/contributor Art Hunkins has reported that the sound quality in the most recent 64s is much improved. For example, the release part of the sound envelope now cuts off cleanly, without an annoying sound residue. Volume changes are smoother. Commodore explains it in terms of manufacturing. The SID chip, a Very Large Scale Integrated Chip (VLSI), has a much higher circuit density than LSI chips such as the 6502. After a SID chip is produced, it is tested and graded for quality. When the SID chip was high in demand and low in availability, the tolerance for error was raised, letting more marginal chips squeak through. Now that the SID chip is cheap and plentiful, Commodore can afford to be more picky, and only the cream of the crop get into production 64s. Naturally, the sound quality is better.

Commodore 64 video quality is also at an all-time high. Sparkle is just a bad memory. Sprites are clean and sharp, and colors are bright. There is still RF interference on some televisions, but color smearing on ordinary TVs is much less of a problem.

Pascal For The 64

From the people who developed the PETSpeed BASIC compiler comes *Oxford Pascal*. PETSpeed was (and is) extremely impressive. It can compile any BASIC program, and fully supports integer math. PETSpeed is truly fast, one of the few optimizing compilers available on 6502 systems. We haven't seen the 64 version of PETSpeed, but have worked with the PET version. As might be expected, *Oxford Pascal* is an equally professional product.

It's not easy to implement a powerful Pascal system on a microcomputer, but *Oxford Pascal* is a good, usable language. When I first encountered Pascal, I was suspicious. Pascal, with its indented statements and mandatory semicolons, smacked of rigidity. It seemed to be a language that was not intended for programming, but for teaching programming.

But Pascal does not enforce "pretty printing"

or flowcharting. In fact, Pascal is very similar to BASIC in many ways. It has excellent control structures, which give a programmer more, not less, flexibility when programming. I've always agreed that GOTO 100 was fairly meaningless unless you looked at line 100. With Pascal, you never need to use GOTO. It's not so much that GOTO is illegal—it's available even in a structured language like Pascal, but with so many luxurious features like IF..THEN..ELSE, REPEAT..UNTIL, WHILE, and CASE, you truly never want to use GOTO again.

Pascal programs are inherently easy to read, thanks in part to these control structures. Which one of these examples do you prefer?

BASIC: FOR I=100 TO 1 STEP-1:PRINT I:NEXT
Pascal: FOR I:=100 DOWNTO 1 DO WRITELN (I);

Modulation

Another powerful feature of Pascal is that you can write programs in modules. (By the way, modular programming and structured programming are not necessarily the same thing.) Each module is a procedure, which you call by name, rather than the cryptic GOSUB 5128. You can pass *values* as parameters to these procedures, rather than using variables. Variables within a procedure can be local. A local variable within a procedure can have the same name as a variable in the main program, or in another procedure, but there is no debugging nightmare of renaming variables used in two different parts of the program.

Procedures also make designing a large program easier. Instead of trying to write and debug a huge, continuous program, Pascal lets you write and test modules separately, then bring them together to form a program.

Compiling Vs. Interpreting

Unlike BASIC, which is interpreted line-by-line as it is executed, a Pascal program must be compiled into machine language or pseudo-code (a high speed interpreted "virtual" machine language) before you can run it. When BASIC is running a program, it looks at each character or command, interpreting, checking for errors, and making decisions all along the way. With a compiler, some of these decisions (such as how much memory to reserve for an array, or the actual address where a GOTO should jump to) are solved during the compilation process. A compiler also translates the program into a faster, simpler, more efficient language (called P-code in most versions of Pascal), which is then executed by a high-speed interpreter. A *native code* compiler translates your program directly into machine language. This machine language program can then be loaded and run just like any machine language program. The difference in speed can be phenomenal.

Since Pascal is designed to be easy to compile, unlike BASIC, there are some concessions you must make for the sake of fast, efficient compilation. Every time you want to use a variable, you must declare its name and type (e.g., integer, floating point, character). The end of a line is not a statement terminator, since you can carry some statements across many program lines. So even if you only have one statement on a line, you must separate it from the next line by putting a semicolon at the end of the statement. Semicolons are the most confusing part of Pascal. I wish someone would write a compiler clever enough to obviate them.

BASIC was designed to be easy to learn and accessible to the masses (that's me and you). Pascal seems to have been designed *by* programmers *for* programmers. Because it lends itself to structured programming, it has become ideal for teaching programming. A teacher can find it hard to grade dozens of wildly different BASIC programs. Pascal is sometimes blamed for encouraging conformity, but this is really a product of the educational programming environment. As it turns out, large companies which hire Pascal programmers out of college also prefer programmers who write neat, well-documented, structured programs.

But we aren't programming for IBM, are we? Why would you want to program with Pascal on a microcomputer? One good reason is if you are taking Pascal in school. If you have a powerful microcomputer Pascal, you can write and debug your programs at home, instead of having to wait for a terminal on the university's huge timesharing system. But most persuasive of all is that the microcomputer environment lets Pascal be more interactive than it is on these behemoth machines.

For example, it always seemed cumbersome to write a Pascal program with a line editor, save it, compile it, link it, then run it. BASIC is so much easier—just type RUN. Well, *Oxford Pascal* lets you do the same thing. No more write-compile-link-execute. Just type RUN, and your program in memory is quickly compiled and run. *Oxford Pascal* makes Pascal as "friendly" as BASIC in this regard. Another bonus from compiling, aside from the higher speed you get from an interpreter, is that some programming errors will be detected before the program is executed, saving you hours of debugging a flawed program.

In this interactive mode, *Oxford Pascal* does

not have the memory to fully support all the advanced features of Pascal, but it has a separate disk-based compiler that supports every Pascal feature you could ask for. Only the speed and memory of a microcomputer limit what *Oxford Pascal* can do.

Since Pascal is compiled, I expected it to run extremely fast. However, although *Oxford Pascal* is zippy, it's not as fast as some BASIC compilers I've used. Pascal is a higher level language than BASIC, though, so you sacrifice some speed for programming power.

The *Oxford Pascal* manual is short and tutorial. It isn't sufficient to learn Pascal with, but is a good place to start, and serves as a reference while you're learning. There are some errors in the text and example programs, but not enough to invalidate the manual. An errata sheet clears things up. BASIC is not the end-all of programming languages, and if programming turns you on, you owe it to yourself to find a language you really feel comfortable with. I still prefer machine language for the kind of programming I do, but there's a valid need for Pascal on microcomputers.

Oxford Computer Systems (Software) Ltd.
Hensington Road, Woodstock Oxford
OX7 1 JR England
Distributed in the U.S by:
Limbic Systems, Inc.
560 San Antonio Road
Suite 202
Palo Alto, CA 94306
$49.95 (disk)

At press time, we learned that Commodore announced a discontinuation of the MCS-801 color dot-matrix printer, which was covered in this column last month. It appears that dealers never received it. ■

Dan Carmichael, Assistant Editor

Simulating Hi-Res Animation Part 2

Last month we reviewed the basics of creating custom characters. This month we'll show you how to simulate high-resolution animation using them.

When using custom characters to animate in the normal fashion, action can be jumpy. This is because characters are usually moved one screen position, eight pixels, at a time. To illustrate this, enter and RUN the program below. You'll see a vertical bar (a custom character) move across the screen. (Note: All programming examples in this article should be used with an unexpanded VIC.)

```
5 PRINT"{CLR} PLEASE WAIT...":A=7168:B=76
  79:S=7680:C=38400:CE=38911:SE=8191
                                    :rem 117
10 FORX=ATOB:POKEX,0:NEXT:FORX=CTOCE:POKE
   X,6:NEXT                         :rem 89
20 READB:IFB=999THEN50              :rem 71
30 POKEA,B:A=A+1:GOTO20             :rem 107
50 POKE36869,255                    :rem 108
70 POKES,0:FORA=STOSE:POKEA,0:POKEA-1,32:
   FORT=1TO50:NEXTT:NEXTA           :rem 131
100 POKE36869,240                   :rem 146
63010 DATA128,128,128,128,128,128,128,128
                                    :rem 32
63090 DATA 999                      :rem 199
```

The secret of smooth animation is to move the character one row of pixels at a time. To see how this is done, enter and RUN this program, which moves the same vertical bar across the screen, but with one pixel increments. Notice how much smoother the animation is.

```
5 PRINT"{CLR} PLEASE WAIT...":A=7168:B=76
  79:S=7680:C=38400:CE=38911:SE=8191
                                    :rem 117
10 FORX=ATOB:POKEX,0:NEXT:FORX=CTOCE:POKE
   X,6:NEXT                         :rem 89
20 READB:IFB=999THEN50              :rem 71
30 POKEA,B:A=A+1:GOTO20             :rem 107
50 POKE36869,255                    :rem 108
200 PRINT"{HOME}{11 SPACES}"        :rem 118
210 FORA=STOSE:FORB=0TO7:POKEA,B:FORT=1TO
    10:NEXTT:NEXTB:POKEA,32:NEXTA:rem 172
500 POKE36869,240                   :rem 150
1000 DATA128,128,128,128,128,128,128,128
                                    :rem 231
2000 DATA64,64,064,064,064,064,064,064
                                    :rem 128
2010 DATA32,32,032,032,032,032,032,032
                                    :rem 89
2030 DATA16,16,016,016,016,016,016,016
                                    :rem 107
```

Figure 1

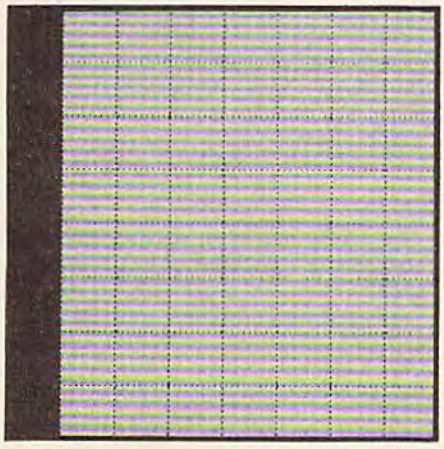


Figure 2

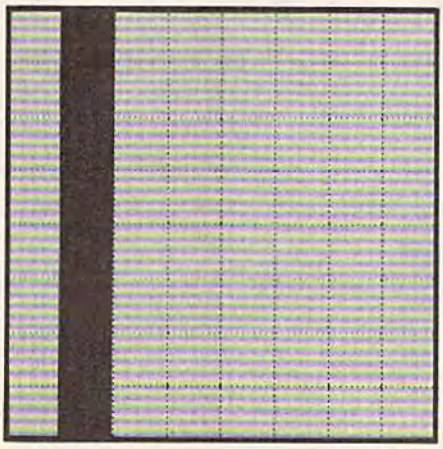
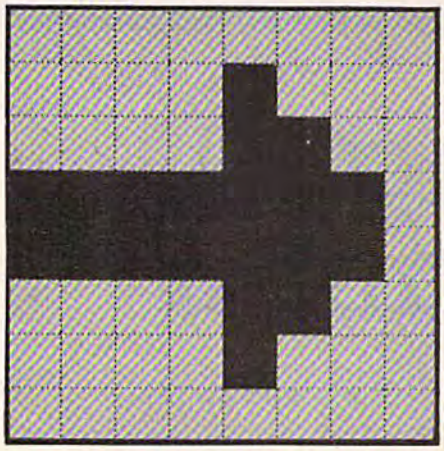


Figure 3

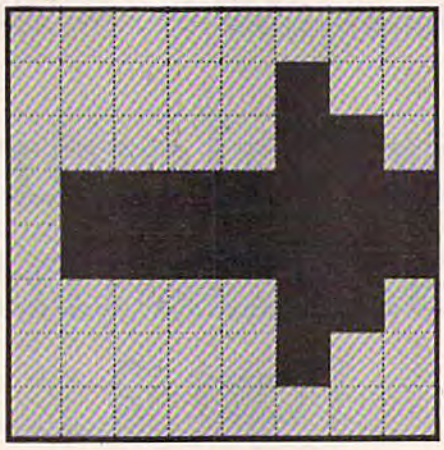
The vertical bar has been shifted to the right in each character. Shaded areas indicate pixels that are turned on.

Figure 4



In Figure 4, the arrow starts the animation cycle.

Figure 5



In Figure 5, the arrow has been shifted one row of pixels to the right.

```
2040 DATA8,008,008,008,008,008,008,008
                                    :rem 116
2050 DATA4,004,004,004,004,004,004,004
                                    :rem 85
2060 DATA2,002,002,002,002,002,002,002
                                    :rem 70
2070 DATA1,001,001,001,001,001,001,001
                                    :rem 63
2080 DATA 999                       :rem 143
```

Shifting Custom Characters

To simulate smooth animation, you need to design more than one custom character. Each one will have to be shifted one row of pixels within its 8 × 8 grid. As an example, take a look at Figures 1–3. These are three of the eight custom characters necessary to smoothly animate the vertical bar. Notice how the vertical bar has been shifted one row of pixels to the right in each successive character. This shifting continues through all eight characters, with the eighth character having the vertical bar all the way to the right.

In the case of the vertical bar, animation is done by POKEing the first eight characters to the same screen position in succession. As each character is POKEd to the screen, it appears that the vertical bar is shifting one row of pixels to the right. To move the vertical bar to the next screen position, a repeat of the first custom character is POKEd to the next successive space on the screen, and a blank space is POKEd to the previous one. This process is repeated, moving the vertical bar across the screen.

Animating Large Custom Characters

Animating the vertical bar is easy because of the size (width) of the character. One complete cycle (eight POKEs) of animation could be done within the confines of one screen position. Only eight custom characters had to be designed to perform the animation.

But what do you do if the characters are bigger and can't be moved a complete animation cycle within one screen position? Two screen positions must be used, and as many as 16 custom characters have to be designed for the animation

Figure 6

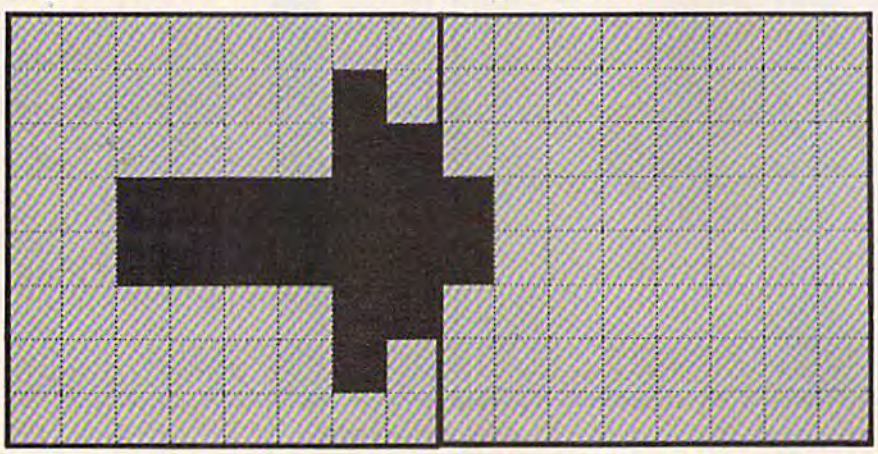
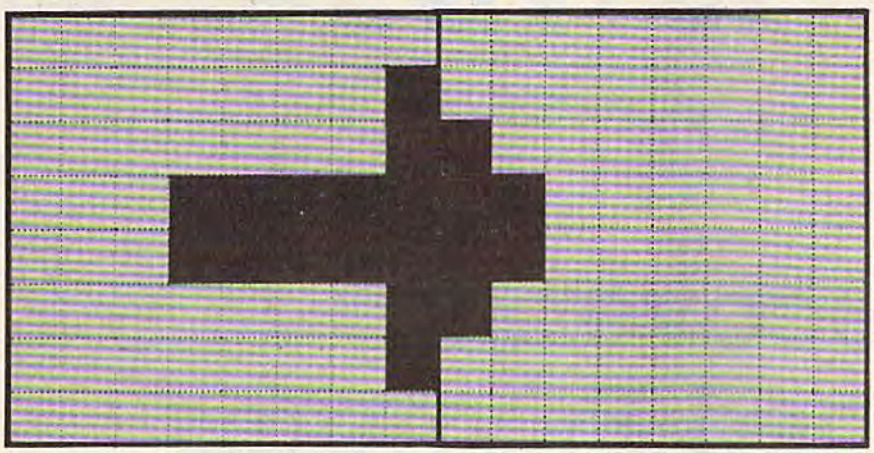


Figure 7

The arrow has been shifted one more row to the right and onto the next grid.

Figure 8 **Figure 9**



The animation cycle continues as the arrow moves even further into the adjacent grid.

effect. Enter and RUN this program, which will move an arrow smoothly across the screen.

```
10 PRINT"{CLR} PLEASE WAIT...":B=7168:S=7
   680:SE=8191:Z=999:M=7168:N=7679 :rem 6
12 C=38400:CE=38911                :rem 231
15 FORA=MTON:POKEA,0:NEXT:FORX=CTOCE:POKE
   X,6:NEXT                        :rem 72
20 READX:IFX=ZTHEN50               :rem 34
30 POKEB,X:B=B+1:GOTO20            :rem 132
50 POKE36869,255:PRINT"{HOME}{12 SPACES}"
                                   :rem 138
60 POKES,0:POKES,1                 :rem 93
70 FORA=2TO13STEP2:POKES,A:POKES+1,A+1:NE
   XT                              :rem 246
80 POKES,32:S=S+1:IFS=>8191THENPOKE36869,
   240:END                         :rem 159
90 GOTO60                          :rem 8
63000 DATA0,008,012,254,254,012,008,000
                                   :rem 147
63001 DATA0,004,006,127,127,006,004,000
                                   :rem 144
63002 DATA0,002,003,063,063,003,002,000
                                   :rem 133
63003 DATA0,000,000,128,128,000,000,000
                                   :rem 128
63004 DATA0,001,001,031,031,001,001,000
                                   :rem 119
63005 DATA0,000,128,192,192,128,000,000
                                   :rem 154
63006 DATA0,000,000,015,015,000,000,000
                                   :rem 121
63007 DATA0,128,192,224,224,192,128,000
                                   :rem 172
63008 DATA0,000,000,007,007,000,000,000
                                   :rem 125
63009 DATA0,064,096,240,240,096,064,000
                                   :rem 174
63010 DATA0,000,000,003,003,000,000,000
                                   :rem 110
63011 DATA0,032,048,248,248,048,032,000
                                   :rem 167
63012 DATA0,000,000,001,001,000,000,000
                                   :rem 108
63013 DATA0,016,024,252,252,024,016,000
                                   :rem 151
63020 DATA 999                     :rem 192
```

The arrow is seven pixels wide. Because of this width, animation is accomplished by moving the arrow from one custom character grid to the next. See Figures 4–9.

In Figure 4, the arrow is contained within one custom character grid. In Figure 5, it's still within one grid, but, as with the vertical bar, it's been shifted one row to the right. Figures 6 and 7 depict two different custom characters side-by-side. As you can see, the arrow is again shifted one pixel row to the right. However, to do this we've moved it onto the next custom character grid. Figures 8 and 9 continue the process, with the arrow being shifted one more row to the right, further onto the next grid. This process continues until the arrow is completely within the grid on the right, and the cycle begins again.

As you can see, moving the arrow one pixel at a time necessitates moving part of it into the next screen position. In some cases, as many as two custom characters may be needed to create one arrow. Your custom characters will have to be designed with this in mind, and in some cases one character will have to be designed using two grids simultaneously. And if you want to move a character smoothly up, down, left, and right, you may need to combine four different custom characters.

Speed and Fine Animation

One drawback to simulating hi-res animation is speed. Because so many custom characters are involved, animation is sometimes sluggish. There are ways, however, to speed up the movement. One thing you can do is PRINT the custom characters, rather than POKEing them to screen memory. Printing is usually faster (you'll have to remember which characters you redefined if you use this method). Or try moving the characters two rows of pixels at a time instead of one. This will not only increase the speed, but will also reduce the number of custom characters needed, and the amount of memory used.

Recovering Scratched Programs

Daryl Biberdorf

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

Perhaps you loaded a program you've been working on, made a few changes, and decided to save it. But first you got the disk directory (LOAD "$",8) and scratched the old version (OPEN 15,8,15, "S0:oldname": CLOSE15). Then it hits you. The newest version of the program has been destroyed, overwritten by the directory. And you don't have the older version, you just scratched it.

What do you do when you've scratched the only copy of a program you were working on?

There are utility programs which allow you to go in and restore or unscratch a scratched program. Even if you have such a program, you might want to try something faster and easier.

Loading A Scratched Program

When you scratch an important program, don't turn off your computer or disk drive. Try this technique for rescuing your program.

First, check the directory: LOAD "$",8 followed by LIST. If you used the wrong filename or incorrectly opened the command channel, your program may still be there. But chances are, you did not make a mistake. The file is gone.

Now type LOAD "*",8 and LIST. You should see the program you thought you just scratched. Before you do anything else, save the program.

This trick may not always work, depending on what you did before and after scratching the file. Your disk drive has a built-in microprocessor which performs the usual functions of loading, saving, and updating the directory. But it also remembers which program was the last to be accessed. When you enter LOAD "*",8 the computer looks for the previous program. It can even find a program which was just scratched, if it was the last program to be saved or loaded.

Scratching a file does not really erase it. It simply removes the program name from the directory and frees up some space on the disk for future SAVEs. If you scratch a file you just saved or loaded, it's still there and can be rescued with the asterisk. But if you've done something with another file, *it* becomes the previous program and this method will not work.

If you accidentally scratch a machine language file you were working on, you can recover it with LOAD "*",8,1 but to save it requires an ML monitor (and you have to know the starting and ending addresses).

NEWS & PRODUCTS

Dot Matrix Thermal Printer

The HUSH 80, an 80-column, 80 cps dot matrix thermal printer that retails for $159.99, has been introduced by Ergo Systems, Inc.

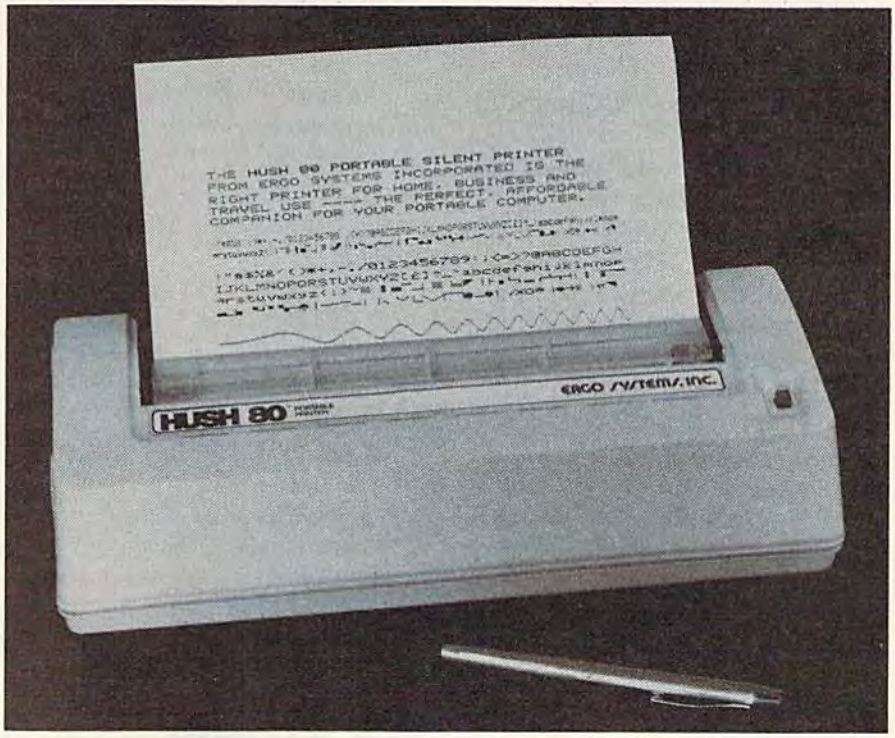
The HUSH 80 comes with interface and cable, and features bidirectional printing. Graphics are printed at 4,800 dots per square inch.

Three models are available, each of which can be equipped with a built-in rechargeable battery pack. The HUSH 80P version has a Centronics parallel interface, while the HUSH 80S provides a serial RS-232 interface.

The printer weighs 28 ounces, and measures 1.63 × 5.5 × 2.8 inches. The unit was manufactured to fit within a standard briefcase. The print line can be set for double size characters at 40 per line, or half-size characters at 160 per line. Line spacing can be programmed to 4.5, 6, or 9 lines to the inch.

All HUSH 80 models typically include the interface, interface cable, 100-foot roll of thermal paper, and a nine-volt AC wall transformer with power cable.

Ergo Systems, Inc.
1360 Willow Road
Menlo Park, CA 94025
(415) 322-3746



The HUSH 80, a bidirectional dot matrix thermal printer from Ergo Systems, Inc., retails for $159.99.

VIC, 64 Graphics Tablet

Personal Peripherals, Inc. has introduced Super Sketch, a graphics tablet for use with the VIC-20 and Commodore 64. Software that expands the capabilities of the tablets is also available.

Sketch-Master, the VIC version, allows the user to create simple line art with a number of computer automated graphics capabilities. Graphics-Master, for the 64, provides all Sketch-Master features, plus a number of advanced capabilities and more sophisticated menu selections.

Included with each version of Super Sketch is a software cartridge and a starter kit of drawings that can be traced from the pad.

Suggested retail price for each version is $59.95.

Personal Peripherals, Inc.
Merrick Park
930 North Beltline Road
Suite 120
Irving, TX 75061
(214) 790-1440

Terminal Software For 64

Madison Computer has introduced *McTerm 64*, a terminal program for the Commodore 64.

Features include an on-screen clock, word wrap, greater than 24K buffer, and auto linefeed options. Baud rate can be set at either 300 or 1,200. The program can be preset to send files at a later time.

Depending on the type of modem used, *McTerm 64* can store and dial up to ten phone numbers, and can automatically answer the phone.

McTerm 64, available on disk, retails for $49.95.

Madison Computer
1825 Monroe
Madison, WI 53711
(608) 255-5552

Games, Education, Home Applications Software For C-64, VIC-20

PandaSoft has announced a number of game, educational, and home applications software for the Commodore 64 and VIC-20.

Disk games for the 64 include: *Pro Golf Challenge* ($25.95), a golf graphics game; *Legend of Starship Terra* ($14.95), a futuristic adventure; and *Revenge of the Phoenix/Time Storm* ($19.95), a graphics arcade and logic program.

Under educational titles, PandaSoft offers *USA Math Star* ($19.95), which teaches addition, subtraction, multiplication, and division; *History 1 & 2* and *Authors* ($12.95 each), two quiz games; *Weights 2 in 1* ($9.98), which teaches both metric and English weights; and *Missing Letter* ($9.98), a word and letter quiz.

Kilowatt Miser ($9.98), an energy saving utility program, *The Budgeter* ($19.95), a budget program, and *M.D.B.* ($39.95), a master data base, are also offered.

Many of the programs are also available for the VIC-20, including *Kilowatt Miser* ($9.95 disk, $5.95 cassette). *General Catalog* ($9.95 disk, $5.95 cassette), a data base, is also available for the VIC.

PandaSoft Computer Software
2732 Rozzelle Ferry Road
P.O. Box 7647
Charlotte, NC 28217
(704) 394-8796

64 Printer Interface

Turboprint/GT, a graphic and text serial-to-parallel printer interface for the 64 has been introduced by Telesys.

The interface prints enhanced Commodore graphics, including reverse characters, and has a line buffer which doubles text printing speed on printers without on-board memory.

An optional Turboprint/B16 or B32 plug-in printer buffer is also available.

Retail price for the interface is $99.95. The 16K buffer retails for $99.95, the 32K buffer for $129.95.

Telesys
43334 Bryant Street
Fremont, CA 94539
(415) 651-2970
(800) 252-4733

VIC, 64 Weather Forecasting

Viasala Inc. has introduced *HAWS*, a software package that helps VIC-20 and Commodore 64 owners forecast the weather.

The *HAWS* (*Home Automatic Weather Station*) package includes weather sensors, plus a software program on either cassette or disk.

The weather sensor comes with a cable which plugs into the user port of the VIC or 64.

HAWS can collect data, graphically display weather trends, and output collected data to standard printers. The data gathered by the *HAWS* sensor can also be used within user-written programs.

HAWS retails for $199.95.

Viasala Inc.
2 Tower Office Park
Woburn, MA 01801
(617) 933-4500

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, [-], 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.

Since the Proofreader is a machine language program stored in the cassette buffer, it will be erased during a tape SAVE or LOAD. If you intend to type in a program in more than one sitting or wish to make a safety SAVE, follow this procedure:

1. LOAD and RUN the Proofreader.
2. Disable it by pressing RUN/STOP–RESTORE.
3. Type the following three lines in direct mode (without line numbers):

```
A$="PROOFREADER.T":B$="{10 SPACES}":FO
RX=1TO4:A$=A$+B$:NEXTX
FORX=886 TO 1018:A$=A$+CHR$(PEEK(X)):N
EXTX
OPEN1,1,1,A$:CLOSE1
```

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

You now have a new version of the Proofreader (PROOFREADER.T, as renamed in the above code). Turn your computer off and on, then LOAD the program you were working on. Put the cassette containing PROOFREADER.T into the tape unit and type:

OPEN1:CLOSE1

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

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

Automatic Proofreader For VIC And 64

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

Bug-Swatter:
Modifications And Corrections

• Program 2, "Change Disk ID," from "Disk Tricks" (September) changes the disk ID in the directory header, but does not actually change the ID on individual disk tracks and sectors. This program was intended to solve the problem of duplicate disk IDs, as explained in the article. It does *not* solve this problem.

When a disk is formatted with the *N0:* (new) command, the disk ID is written on every sector of every track. In addition, a directory header with the disk name and ID is created. When a file is accessed, the disk operating system (DOS) gets its information from the individual sectors, not from the header. The Change Disk ID program changes only the header. Writing to individual tracks and sectors (to actually change the ID) would require reprogramming the disk drive.

If you have disks with duplicate IDs, you can eliminate the problems by either 1) copying all important files to a separate disk and then reformatting the problem disk with a unique ID, or 2) initializing the disk, with the *I0:* command, every time you swap disks.

Since the program makes a cosmetic change to the directory, it does not destroy any data on the disk, and subsequent READ/WRITE operations should work without problems.

Readers who have used the Change Disk ID program can discover the original ID with this short program:

```
10 T=18:S=0:REM TRACK AND SECTOR
20 OPEN2,8,2,"#":OPEN15,8,15
30 PRINT#15,"I0"
40 PRINT#15,"U1";2;0;T;S
50 FORJ=22TO23:PRINT#15,"M-R";CHR$(J);CHR
   $(0):GET#15,Z$:AD$=AD$+Z$:NEXT
60 CLOSE15:CLOSE2
70 PRINT"ID=";AD$
```

This program reads the true ID from track 18, sector 0. After discovering the true ID, you can use the Change Disk ID program to change the ID back to what it should be.

• When the program listing for "Treasure Hunt/64" (September) was made, line 655 was garbled. In addition, line 836 was accidentally omitted. Here are the corrections:

```
655 POKETL,32:POKETL+Q,32:POKECL,46:POKEC
    L+Q,47:POKECL+CM,4:POKECL+CM+Q,4
                                  :rem 81
836 PRINT"APPROACH.{31 SPACES}";  :rem 41
```

• "Screen-80" (September) works as listed, but the information about using sprites contains an error. Screen memory (normally at 1024–2023) is followed by the eight sprite pointers (normally at 2040–2047, or beginning of screen memory plus 1016). Screen-80 uses a high-resolution screen and moves the equivalent of screen memory to 53248, which means the sprite pointers move to 53248+1016, not 53248+2040 as stated in the article. Line 60 of the sprite program on page 50 should be changed accordingly:

```
60 POKE 53248+1016,0              :rem 234
```

• Readers Joseph Maniscalco and Amir Findling have informed us that although the 64 version of "Balloon Blitz" (August) runs without errors, it has a slight glitch in the joystick reading routine. Pushing the joystick to the southeast (down and right) will cause a bomb to be dropped. Changing line 14 to

```
14 IF A>=117 THEN GO TO 3
```

will fix the problem.

• The VIC version of "3-D Tic-Tac-Toe" (June) contains some color codes accessible on the 64 but not on the VIC. This happened when the game was translated and does not adversely affect the program, although the Automatic Proofreader checksums will be incorrect. In lines 300, 320, and 590, rather than pressing the Commodore key and one of the numbered (color) keys, VIC users should press CTRL and the indicated color key.

• "Props" (May) runs as listed, but some readers have encountered an error message for lines 49000 to 49308. This is due to line 49151 where variable CJ is set to zero and then a different variable (CK) is used as a checksum. If line 49151 is entered as listed, the error message should not occur.

• Levitating in the VIC version of "Castle Dungeon" requires pressing the L key. Reader Michael Bank thinks it is more convenient to use the joystick button and suggests the following modification:

```
26 BN= PEEK(37137)AND32: IF BN=0 THEN LS=1:
   GOSUB82
```

SpeedScript Update:

• The VIC version of *SpeedScript* included on the May GAZETTE DISK scrambles the first five

characters of all files. Reader Jon Harmon has discovered that text memory and a few bytes at the end of program memory overlap. To fix this, follow these steps: First, load (but do not run) VICSPEEDSCRIPT from the May GAZETTE DISK. Next, POKE 4627,16: POKE 4989,21 and save the new version to disk. The problem should be solved.

• The July "Bug-Swatter" reported on hardware incompatibilities between the VIC-20, Datassette, and the Commodore 1526 printer. The solution (SYS 64490 after cassette operations) has been incorporated into the *original* (January) VIC version of *SpeedScript* by Reader Brian Mason. First, load (but do not run) the original VIC version. Next, POKE 8560,234: POKE 8561, 215: POKE 8562,96 and save the new version. This replaces the disk access command (Control/up arrow) with the appropriate SYS. VIC tape users should hold down the CTRL key and press the up arrow key, after VIC *SpeedScript* tape saves or loads. The serial bus will become available for use with a 1526. The equivalent POKEs for the May GAZETTE DISK version are 8572–8574, although the problem with 1526 printers does not occur with disk drives.

• Several readers who own a portable SX-64 have indicated that *SpeedScript* disk access—LOADs, SAVEs, and directories—can cause the computer to lock up. Because *SpeedScript* uses a raster interrupt to form the window at the top of the screen, the interrupt registers have to be reset before disk or printer operations.

Rodney L. Barnes disassembled the program and discovered that before disk operations, *SpeedScript* stores a 255 ($FF), the usual value on a Commodore 64, in the CIA interrupt register at 56333 ($DC0D). Bit 4 of this register enables tape operations. Because the SX-64 has no Datassette port and no provisions for tape use, storing this value in the CIA register can cause extraneous interrupts, interfering with the serial I/O. To fix this, load (but don't run) the January version of *SpeedScript* into your SX-64, POKE 4714, 239 (if you have the May GAZETTE DISK Version, POKE 4789,239), and then save the new version to disk.

Readers should note this modification applies only to the portable SX-64, not the Commodore 64.

• "SpeedScript Revisited" (May) included a modification to allow printing to an RS-232 printer. The modification applied only to the VIC version of *SpeedScript*. Readers Clifford Jensen, Lee Folgedalen, and Robert Latham have adapted the 64 version for use with such printers. First, load (but don't run) *SpeedScript*. Then, if you have the January version, POKE 5262,2 (for the May GAZETTE DISK version, POKE 5337,2) and save the new program to tape or disk.

Before running *SpeedScript*, POKE 660,0 and POKE 659,6 (Baud rate of 300) or POKE 659,3 (Baud rate of 110). Put an [a] (press CTRL-£ then "a" and you'll see a reverse "a") at the top of the file, and to print, press SHIFT-CTRL-P followed by a 2 (device 2) then another 2 (secondary address 2).

• Several readers have inquired about a Spelling Checker program for *SpeedScript*. Such a program is not feasible, because it would require users to type in hundreds, if not thousands, of words from a dictionary. However, Robert Murray has found that the commercial program *Spell-Right Plus* for the 64 from Professional Software, designed for use with WordPro3+/64, also works with *SpeedScript*. He suggests removing all formatting commands (reverse-video characters)—header, footer, spacing, and others—before running the program.

We appreciate receiving both corrections and suggested modifications from readers. Address them to:

Bug-Swatter
c/o COMPUTE!'s GAZETTE
P.O. Box 5406
Greensboro, NC 27403

Please indicate the type of error you have found, as well as the line number.

The Tomb

(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 Tomb (VIC Version)

```
0 PRINT"{CLR}":POKE36879,10:GOTO3000:PH=0
  :SC=0:S=7680                        :rem 236
1 GOSUB2000:S=7680:C=38400:DX=1:DY=1:EX(1
  )=10:EX(2)=13:EX(3)=2               :rem 16
2 Q=22:EY(1)=11:EY(2)=10:EY(3)=17:POKE816
  2,32:POKE8139,32:POKE7726,32        :rem 129
3 POKES+Q*DY+DX,33:POKEC+Q*DY+DX,4
                                      :rem 103
4 FORZ=1TO3:POKES+Q*EY(Z)+EX(Z),36:POKEC+
  Q*EY(Z)+EX(Z),5:NEXT                :rem 84
```

```
5 POKE8185,37:POKE38905,7          :rem 165
11 TR=500                          :rem 218
12 POKE36878,15                    :rem 52
18 POKE36875,140:FOR Z=1TO25:NEXT:rem 210
19 POKE36875,0                     :rem 2
20 POKE37154,127:P=PEEK(37152)AND128:J0=-
   (P=0):POKE37154,255             :rem 105
21 P=PEEK(37151):J1=-((PAND8)=0):J2=-((PA
   ND16)=0):J3=-((PAND4)=0)        :rem 48
22 CX=J0-J2:CY=J1-J3               :rem 151
23 IFCX=0ANDCY=0THEN80             :rem 111
24 IFPEEK(S+Q*(DY+CY)+(DX+CX))=35THEN80
                                   :rem 51
25 POKES+Q*DY+DX,32:DX=DX+CX:DY=DY+CY:IFP
   EEK(S+Q*DY+DX)=36THEN4000       :rem 166
28 POKEC+Q*DY+DX,4:POKES+Q*DY+DX,33
                                   :rem 158
29 IFS+Q*DY+DX=8185THEN5000        :rem 94
79 IFINT(RND(1)*3)=1THEN18         :rem 4
80 B=INT(RND(1)*3)+1:CX(B)=0:CY(B)=0:W=0
                                   :rem 235
81 POKES+Q*EY(B)+EX(B),32:CX(B)=((EX(B)>D
   X)-(EX(B)<DX))                  :rem 133
82 EX(B)=EX(B)+CX(B)               :rem 96
85 CY(B)=((EY(B)>DY)-(EY(B)<DY))   :rem 15
86 IFPEEK(S+Q*EY(B)+EX(B)+Q*CY(B))=35THEN
   CY(B)=0:EX(B)=EX(B)-CX(B)       :rem 111
87 EY(B)=EY(B)+CY(B)               :rem 104
88 IFPEEK(S+Q*EY(B)+EX(B))=33THEN4000
                                   :rem 144
89 POKES+Q*EY(B)+EX(B),36:POKEC+Q*EY(B)+E
   X(B),5:TR=TR-1:GOTO18           :rem 246
91 GOTO18                          :rem 12
299 CX(B)=0:CY(B)=0                :rem 21
1999 END                           :rem 179
2000 FORM=7702TO8163               :rem 120
2001 IFINT(RND(1)*3)=1 ANDPEEK(M-1)<>35TH
     EN2004                        :rem 138
2002 NEXT                          :rem 3
2003 RETURN                        :rem 165
2004 POKEM+30720,2:POKEM,35:NEXT   :rem 138
2999 RETURN                        :rem 189
3000 POKE56,28:POKE52,28:POKE51,PEEK(55):
     CLR:PRINT"{3 DOWN}{RVS}";:PRINTTAB(6
     )"REDEFINING"                 :rem 30
3001 CS=256*PEEK(52)+PEEK(51):FORI=CSTOCS
     +511:POKEI,PEEK(I+32768-CS):NEXT
                                   :rem 136
3002 FORI=7432TO7439:READJ:POKEI,J:NEXT
                                   :rem 189
3003 DATA60,60,24,255,60,60,102,102
                                   :rem 212
3004 FORI=7448TO7455:READJ:POKEI,J:NEXT
                                   :rem 196
3005 DATA255,255,255,255,255,255,255,255
                                   :rem 246
3006 FORI=7456TO7463:READJ:POKEI,J:NEXT
                                   :rem 196
3007 DATA129,153,102,60,255,60,66,66
                                   :rem 32
3008 FORI=7464TO7471:READJ:POKEI,J:NEXT:P
     RINT"{CLR}"                   :rem 98
3009 DATA0,0,126,126,126,126,0,0   :rem 62
3010 CLR:POKE36869,255             :rem 230
3011 FORC=7680TO7701:POKEC+30720,2:POKEC,
     35:POKEC+31204,2:POKEC+484,35:NEXT
                                   :rem 151
3012 FORC=7680TO8164STEP22:POKEC+30720,2:
     POKEC,35:POKEC+30741,2:POKEC+21,35:N
     EXT                           :rem 4
3111 GOTO 1                        :rem 48
4000 POKE 36878,15:POKE36876,200:FORL1=25
     0 TO 150STEP-17               :rem 70
4001 POKE36875,L1:FORL2=150TO190:POKE3687
     6,L2:NEXTL2,L1                :rem 26
4002 POKE36875,0:POKE36876,0       :rem 49
4003 PRINT"{HOME}{WHT}{DOWN}{2 SPACES}THE
      GUARDIAN HAS{9 SPACES}CAUGHT YOU."
                                   :rem 176
4004 PRINTTAB(8)"{2 DOWN}{YEL}GOLD={BLU}"
     SC                            :rem 209
4005 PRINT"{DOWN}{CYN}HIT THE TRIGGER IF
     {SPACE}YOU{2 SPACES}DARE TO TRY AGAI
     N."                           :rem 252
4006 PRINT"{DOWN}{CYN}TYPE (Q) IF YOU WIS
     H{9 SPACES}TO QUIT."          :rem 10
4007 GETA$:P=PEEK(37137):IF A$="Q"THENPOK
     E251,0:SYS251                 :rem 117
4008 IF-((PAND32)=0)<>1THEN4007    :rem 196
4009 IF SC>PH THEN PH=SC           :rem 98
4010 PRINT"{CLR}":SC=0:GOTO3011    :rem 158
5000 PRINT"{HOME}{WHT}YOU GRAB THE TREASU
     RE"                           :rem 19
5001 FORZ=1TO100:POKE36876,INT(RND(1)*128
     )+128                         :rem 121
5002 FORHG=1TO10:NEXT:NEXT:POKE36876,0
                                   :rem 116
5100 PRINT"{CLR}{5 DOWN}":PRINT"{5 RIGHT}
     {BLU}{2 SPACES}GOLD={YEL}"SC;:PRINT"
     "                             :rem 23
5101 FORL=1TOTR:SC=SC+1:PRINT"{UP}
     {12 RIGHT}"SC:NEXT            :rem 183
5120 PRINT"{2 DOWN}{RED} MOST TREASURE RE
     COV-{2 SPACES}ERED FROM TOMB YET":PR
     INTTAB(8)"{YEL}"PH            :rem 74
5121 PRINT"{DOWN}{PUR} PRESS Q TO QUIT NO
     W"                            :rem 255
5122 PRINT:PRINT{2 SPACES}"{3 SPACES}HIT
     {SPACE}THE TRIGGER{9 SPACES}TO CONTI
     NUE"                          :rem 68
5123 P=PEEK(37137):IF -((PAND32)=0)=1THEN
      PRINT"{CLR}":GOTO 3011       :rem 153
5124 IF PEEK(197)=48 THEN POKE251,0:SYS25
     1                             :rem 62
5125 GOTO 5123                     :rem 209
```

Program 2: The Tomb (64 Version)

```
0 POKE56,28:CLR:PRINT"{CLR}":POKE53280,2:
  POKE53281,0:GOTO3000:PH=0:SC=0   :rem 92
1 GOSUB2000:C=55296:DX=1:DY=1:EX(1)=5:EX(
  2)=20:EX(3)=35                   :rem 133
2 Q=40:EY(1)=12:EY(2)=6:EY(3)=22:POKE1982
  ,32                              :rem 183
3 POKE1024+Q*DY+DX,33:POKEC+Q*DY+DX,4
                                   :rem 219
4 FORZ=1TO3:POKE1024+Q*EY(Z)+EX(Z),36:POK
  EC+Q*EY(Z)+EX(Z),5:NEXT          :rem 200
5 POKE2023,37:POKE56295,7:S=1024   :rem 41
6 CL=54272:VL=CL+24:FORI=CLTOCL+24:POKEI,
  0:NEXT                           :rem 191
11 TR=500                          :rem 218
18 POKEVL,15:POKECL+5,15:POKECL+6,129:FOR
    Z=1TO25:NEXT:POKEVL,0          :rem 170
20 X1=PEEK(56320):CY=-(((X1AND1)=1)+((X1A
   ND2)=2)*-1)                     :rem 30
21 CX=((X1AND8)=8)+((X1AND4)=4)*-1:rem 60
23 IFCX=0ANDCY=0THEN80             :rem 111
24 IFPEEK(S+Q*(DY+CY)+(DX+CX))=35THEN80
                                   :rem 51
25 POKES+Q*DY+DX,32:DX=DX+CX:DY=DY+CY:IFP
```

```
   EEK(S+Q*DY+DX)=36THEN4000      :rem 166
28 POKEC+Q*DY+DX,4:POKES+Q*DY+DX,33
                                  :rem 158
29 IFS+Q*DY+DX=2023THEN5000       :rem 79
80 B=INT(RND(1)*3)+1:CX(B)=0:CY(B)=0:W=0
                                  :rem 235
81 POKES+Q*EY(B)+EX(B),32:CX(B)=((EX(B)>D
   X)-(EX(B)<DX))                 :rem 133
82 EX(B)=EX(B)+CX(B)              :rem 96
85 CY(B)=((EY(B)>DY)-(EY(B)<DY))  :rem 15
86 IFPEEK(S+Q*EY(B)+EX(B)+Q*CY(B))=35THEN
   CY(B)=0:EX(B)=EX(B)-CX(B)      :rem 111
87 EY(B)=EY(B)+CY(B)              :rem 104
88 IFPEEK(S+Q*EY(B)+EX(B))=33THEN4000
                                  :rem 144
89 POKES+Q*EY(B)+EX(B),36:POKEC+Q*EY(B)+E
   X(B),5:TR=TR-1:GOTO18          :rem 246
91 GOTO18                         :rem 12
299 CX(B)=0:CY(B)=0               :rem 21
1999 END                          :rem 179
2000 FORM=1064TO1983              :rem 118
2001 IF INT(RND(1)*3)=1ANDPEEK(M-1)<>35TH
     EN 2004                      :rem 138
2002 NEXT                          :rem 3
2003 RETURN                       :rem 165
2004 POKEM+54272,2:POKEM,35:NEXT  :rem 146
2999 RETURN                       :rem 189
3000 POKE56334,PEEK(56334)AND254:POKE1,PE
     EK(1)AND251:PRINT"{3 DOWN}{RVS}";
                                  :rem 111
3001 PRINTTAB(15)"REDEFINING{OFF}":FORI=0
     TO511:POKEI+12288,PEEK(I+53248):NEXT
     I                            :rem 97
3002 FORI=12552TO12559:READJ:POKEI,J:NEXT
                                  :rem 27
3003 DATA60,60,24,255,60,60,102,102
                                  :rem 212
3004 FORI=12568TO12575:READJ:POKEI,J:NEXT
                                  :rem 34
3005 DATA255,255,255,255,255,255,255,255
                                  :rem 246
3006 FORI=12576TO12583:READJ:POKEI,J:NEXT
                                  :rem 34
3007 DATA129,153,102,60,255,60,66,66
                                  :rem 32
3008 FORI=12584TO12591:READJ:POKEI,J:NEXT
     :PRINT"{CLR}"                :rem 192
3009 DATA0,0,126,126,126,126,0,0  :rem 62
3010 POKE1,PEEK(1)OR4:POKE56334,PEEK(5633
     4)OR1:POKE53272,PEEK(53272)AND240OR1
     2                            :rem 49
3011 FORC=1024TO1063:POKEC+54272,2:POKEC,
     35:POKEC+55232,2:POKEC+960,35:NEXT
                                  :rem 146
3012 FORC=1024TO1984STEP40:POKEC+54272,2:
     POKEC,35:POKEC+54311,2:POKEC+39,35
                                  :rem 144
3013 NEXT                          :rem 6
3111 GOTO 1                       :rem 48
4000 FORL2=1TO5:POKEVL,9:POKECL+5,72:POKE
     CL+6,241:FORL1=50TO25STEP-1  :rem 204
4001 POKE CL+1,L1:POKECL+4,33:NEXTL1:NEXT
     L2:POKECL+4,32               :rem 70
4003 PRINTTAB(6)"{WHT}{DOWN}THE GUARDIAN
     {SPACE}HAS CAUGHT YOU"       :rem 205
4004 PRINTTAB(17)"{2 DOWN}{YEL}GOLD={BLU}
     "SC                           :rem 1
4005 PRINT"{DOWN}{CYN}HIT THE TRIGGER IF
     {SPACE}YOU DARE TO TRY AGAIN"
                                  :rem 206
4006 PRINTTAB(6)"{DOWN}{CYN}TYPE (Q) IF Y
     OU WISH{2 SPACES}TO QUIT"    :rem 58
4007 GETA$:X1=PEEK(56320)AND16:IFA$="Q"TH
     ENPOKE53272,21:PRINT"{CLR}":END
                                  :rem 95
4009 IFX1<>0THEN 4007             :rem 137
4010 IF SC<>PH THEN PH=SC         :rem 150
4011 PRINT"{CLR}":SC=0:GOTO 3011  :rem 159
5000 PRINT:PRINTTAB(10)"{WHT}YOU GRAB THE
      TREASURE"                   :rem 80
5001 POKEVL,15:POKECL+5,71:POKECL+6,241:F
     ORZ=1TO100:POKECL+4,33       :rem 55
5002 POKECL+1,INT(RND(1)*128)+64:FORHG=1T
     O10:NEXT:NEXT:POKECL+4,32     :rem 5
5100 PRINT"{CLR}{5 DOWN}":PRINTTAB(14)"
     {BLU}{2 SPACES}GOLD={YEL}"SC;:PRINT"
      "                           :rem 19
5101 FORL=1TOTR:SC=SC+1:PRINTTAB(21)"{UP}
     "SC:NEXT                     :rem 230
5120 PRINT"{2 DOWN}{RED}{2 SPACES}MOST TR
     EASURE RECOVERED BEFORE=";:PRINTTAB(
     32)"{YEL}"PH                 :rem 29
5122 PRINTTAB(10)"{DOWN}{PUR} PRESS Q TO
     {SPACE}QUIT NOW"             :rem 137
5123 PRINT:PRINTTAB(7)"HIT THE TRIGGER TO
      CONTINUE"                   :rem 164
5124 GETA$:X1=PEEK(56320)AND16:IFX1=0 THE
     N PRINT"{CLR}":GOTO3011      :rem 75
5125 IF A$="Q"THEN POKE53272,21:PRINT"
     {CLR}":END                   :rem 50
5126 GOTO 5124                    :rem 211
```

Cabby

(Article on page 60.)

BEFORE TYPING...

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

Program 1: Cabby (VIC Version)

```
5 POKE56,28:POKE55,0:POKE52,28:POKE51,0
                                  :rem 37
10 PRINT"{CLR}":POKE36879,104      :rem 0
20 PRINT"{7 DOWN}{4 RIGHT}{CYN}SETTING UP
   ":PRINT"{DOWN}{4 RIGHT}PLEASE WAIT!!!!
   "                              :rem 131
50 FORI=7168TO7679:POKEI,PEEK(I+25600):NE
   XTI                            :rem 175
55 FORI=1TO13:READZ:FORJ=ZTOZ+7:READK:POK
   EJ,K:NEXTJ,I                   :rem 35
60 POKE37139,0:DD=37154:PA=37137:PB=37152
                                  :rem 19
65 DIM F$(26),T%(4),B%(4)         :rem 170
66 FORI=1TO26:READF$(I):NEXTI      :rem 49
150 H=7680:C=30720:CH=28:B%(1)=8055:B%(2)
    =7914:B%(3)=8051:B%(4)=7730   :rem 222
160 DEFFNA(X)=INT(RND(1)*X)+1:DEFFNL(Q)=H
    +22*Y+X                       :rem 70
170 Y=7:X=6:T%(1)=FNL(Q):Y=18:T%(2)=FNL(Q
    ):Y=8:X=16:T%(3)=FNL(Q):Y=15:T%(4)=FN
    L(Q)                          :rem 237
180 PRINT"{CLR}{11 DOWN}{WHT}LEVEL 1-EASY
     TO 4-HARD"                   :rem 232
```

```
190 GETZ$:LV=VAL(Z$):IFZ$=""THEN190
                                    :rem 84
193 IFLV<1ORLV>4THEN190             :rem 185
195 POKE36869,255:POKE36879,107     :rem 228
200 PRINT"{CLR}":GOSUB3000          :rem 115
210 P%=H+45:POKEP%,CH:POKEP%+C,7    :rem 101
220 XF%=0:CF%=0:GS=1400:RU=1        :rem 62
300 GOSUB1000                       :rem 212
310 IFXF%=0ANDRND(1)>.6THENGOSUB2000
                                    :rem 159
320 GOSUB500:IFSG=1THENSG=0:GOTO340
                                    :rem 187
330 IFRND(1)>.98THENGOSUB7000       :rem 222
340 GOSUB500:SG=0                   :rem 237
350 GOSUB5000                       :rem 221
360 GOSUB500:SG=0                   :rem 239
390 GOTO300                         :rem 104
500 POKEDD,127:S3=-((PEEK(PB)AND128)=0):P
    OKEDD,255                       :rem 122
510 P=PEEK(PA):S1=-((PAND8)=0):S2=((PAND1
    6)=0):S0=((PAND4)=0)            :rem 177
520 FR=-((PAND32)=0):DX=S2+S3:DY=S0+S1
                                    :rem 191
525 IFFRTHENGOSUB9600               :rem 65
530 CH=28                           :rem 202
535 IFDX=-1THENCH=29                :rem 197
540 IFDY<>0THENCH=27                :rem 207
550 Z=P%+22*DY+DX                   :rem 195
560 IFPEEK(Z)=32THEN610             :rem 98
570 IFPEEK(Z)=40ORPEEK(Z)=41THEN8000
                                    :rem 166
580 IFPEEK(Z)=31THEN500             :rem 97
590 IFPEEK(Z)=30THENIFPEEK(Z+C)<>13THEN50
    0                               :rem 59
600 IFPEEK(Z)>=0ANDPEEK(Z)<=26THENGOTO400
    0                               :rem 80
605 IFPEEK(Z)>=36ANDPEEK(Z)<=39THEN6000
                                    :rem 91
606 IFPEEK(Z)=42THEN9000            :rem 150
610 POKEP%,32                       :rem 204
620 POKEZ,CH:POKEZ+C,15:P%=Z        :rem 225
625 GS=GS-2:IFGS=<0THENSG=1:GOSUB9700
                                    :rem 58
630 RETURN                          :rem 121
1000 L1%=FNA(3)+1:L1%=L1%*2-1:IFL1%=3THEN
     L1%=2                          :rem 229
1010 L2%=FNA(4):POKET%(L2%),30:POKET%(L2%
     )+C,L1%+8                      :rem 214
1020 RETURN                         :rem 163
2000 RU=0:XF%=FNA(26):A$="{HOME}{20 DOWN}
     ":B$="{21 SPACES}"             :rem 70
2005 PRINTA$;B$:PRINTB$:PRINTA$;"{WHT}CAB
     BY, GO TO THE":PRINTF$(XF%);   :rem 94
2020 RETURN                         :rem 164
2500 ZT=PEEK(Z+C)                   :rem 240
2505 IFZT=20RZT=7THEN350              :rem 2
2510 RETURN                         :rem 168
3000 REM MAP                        :rem 133
3005 A$="{BLK}←←←←←←←←←←←←←←←←←←←←":PRIN
     T"{HOME}{BLK}←←←←←←←←{WHT}CABBY{BLK}
     ←←←←←←←←"                      :rem 157
3010 FORI=1TO20                     :rem 103
3015 PRINTA$:NEXTI                  :rem 125
3020 PRINT"{HOME}{DOWN}{11 RIGHT}{WHT}G
     {3 RIGHT}B":PRINT"{RIGHT}{3 SPACES}
     {2 RIGHT}{14 SPACES}Z"         :rem 154
3030 PRINT"{RIGHT} {4 RIGHT}D{RIGHT}
     {7 RIGHT} {RIGHT}S "           :rem 196
3040 PRINT"{RIGHT} {2 RIGHT}M{4 SPACES}
     {RIGHT}S{3 SPACES}Y{RIGHT} {2 RIGHT}
      "                             :rem 92
3050 PRINT"{RIGHT} {6 RIGHT} {RIGHT}L
     {RIGHT} {2 RIGHT} {2 RIGHT} V"
                                    :rem 180
3060 PRINT"{RIGHT} {RIGHT}I{4 RIGHT}
     {2 RIGHT} {RIGHT} {2 RIGHT} T{RIGHT}
      "                             :rem 147
3070 PRINT"{WHT}A{13 SPACES}{RIGHT}Q
     {4 SPACES}"                    :rem 79
3080 PRINT"{RIGHT} {3 RIGHT} N{2 RIGHT}S
     {RIGHT} {RIGHT} {2 RIGHT} {RIGHT}O "
                                    :rem 203
3090 PRINT"W {3 RIGHT} {5 RIGHT} {RIGHT}
     {4 SPACES}P{RIGHT} "           :rem 102
3100 PRINT"{RIGHT}{3 SPACES}G{7 SPACES}
     {RIGHT} {2 RIGHT} {2 RIGHT} U"
                                    :rem 223
3110 PRINT"{RIGHT} {3 RIGHT} {RIGHT}M
     {5 RIGHT} H{RIGHT}{4 SPACES}"
                                    :rem 106
3120 PRINT"{RIGHT} D{RIGHT}J {2 RIGHT}T
     {2 RIGHT}{3 SPACES}{RIGHT}B
     {2 RIGHT} "                    :rem 192
3130 PRINT"B {2 RIGHT}K {2 RIGHT}
     {2 RIGHT}R{RIGHT} {2 RIGHT} E{RIGHT}
      R"                            :rem 48
3140 PRINT"{RIGHT} {3 RIGHT}{12 SPACES}
     {2 RIGHT} "                    :rem 71
3150 PRINT"{RIGHT}{5 SPACES}{3 RIGHT}
     {6 RIGHT} {RIGHT}O "           :rem 40
3160 PRINT"G {RIGHT}X{RIGHT} {2 RIGHT}S
     {RIGHT}T{8 SPACES}"            :rem 114
3170 PRINT"{RIGHT} {3 RIGHT} {3 RIGHT}
     {2 RIGHT} G{2 RIGHT} {2 RIGHT} "
                                    :rem 92
3180 PRINT"{RIGHT}{12 SPACES}{2 RIGHT}
     {2 SPACES}M{RIGHT} F"          :rem 164
3190 PRINT"{RIGHT}C{2 RIGHT}N{2 RIGHT}R
     {4 RIGHT}B"                    :rem 200
3200 POKEH+46,40:POKEH+47,41:POKEH+46+C,5
     :POKEH+47+C,5                  :rem 100
3250 X=15:Y=19:POKEFNL(0),42:POKEFNL(0)+C
     ,7                             :rem 246
3300 RETURN                         :rem 166
4000 IFCF%<>0THEN4500               :rem 163
4005 IFPEEK(Z)=0THENXF%=99:GOTO4200
                                    :rem 111
4010 IFPEEK(Z)<>XF%THEN500          :rem 37
4200 CF%=FNA(26):SP=Z:POKEZ,CH:POKEZ+C,15
     :POKEP%,32                     :rem 95
4260 PRINTA$;B$:PRINTB$:PRINTA$;"{WHT}TAK
     E ME TO THE":PRINTF$(CF%):FORT=1TO99
     :NEXT                          :rem 24
4280 POKEP%,CH:POKEP%+C,15:POKEZ,XF%:POKE
     Z+C,1:IFXF%=99THEN4290         :rem 128
4285 GOTO 4300                      :rem 211
4290 POKEZ,31:POKEZ+C,0             :rem 117
4300 GOTO630                        :rem 153
4500 IFPEEK(Z)<>CF%THEN630          :rem 24
4510 POKEZ,CH:POKEZ+C,15:POKEP%,32:rem 60
4550 M1=ABS(SP-Z)/10+1              :rem 198
4560 M2=INT(FNA(100*M1)/10)/100     :rem 118
4570 FORI=1TO3:PRINTA$;B$:PRINTB$:PRINTA$
     ;"YOU COLLECT";M1:PRINT"PLUS TIP OF"
     ;M2                            :rem 48
4575 FORT=1TO500:NEXTT,I            :rem 248
4580 M3=INT(M1+M2+M3):M1=0:M2=0     :rem 232
4600 PRINTA$;B$:PRINTB$:PRINTA$;"ON YOU";
     M3:FORT=1TO650:NEXTT           :rem 120
4650 POKEP%,CH:POKEP%+C,15:POKEZ,CF%:POKE
     Z+C,1:CF%=0:XF%=0              :rem 242
4700 GOTO630                        :rem 157
```

```
5000 I=FNA(4):Q1=1                    :rem 207
5005 Y=INT((P%-H)/22):X=(P%-H)-22*Y
                                      :rem 136
5010 Y1=INT((B%(I)-H)/22):X1=(B%(I)-H)-22
     *Y1                              :rem 47
5015 BX=0:BY=0                        :rem 20
5020 BX=(X1>X)-(X>X1)                 :rem 171
5050 BY=(Y1>Y)-(Y>Y1)                 :rem 179
5100 NC=B%(I)+BX+BY*22:IFNC>8097THENRETUR
     N                                :rem 45
5110 IFPEEK(NC)=31THEN5200            :rem 244
5115 IFPEEK(NC)<=26THEN5200           :rem 57
5117 IFPEEK(NC)=30THEN5200            :rem 250
5120 IFNC=P%THEN6000                  :rem 143
5150 POKEB%(I),32                     :rem 140
5155 TY=37:IFBX<0THENTY=36            :rem 117
5156 IFBY<0THENTY=38                  :rem 235
5157 IFBY>0THENTY=39                  :rem 239
5160 POKENC,TY:POKENC+C,4:B%(I)=NC:rem 54
5170 I=I+1:IFI=>4THENI=1              :rem 159
5175 Q1=Q1+1:IFQ1<=LVTHEN5005         :rem 206
5180 RETURN                           :rem 174
5200 BX=FNA(2)-1:BY=FNA(2)-1          :rem 28
5240 GOTO5100                         :rem 202
6000 PRINT"{CLR}":POKE36879,8:POKE36869,2
     40                               :rem 67
6050 PRINT"{4 DOWN}{2 SPACES}YOUR CAB HAS
      BEEN"                           :rem 235
6060 PRINT"{DOWN}CRUNCHED BY A TROLLEY"
                                      :rem 1
6070 PRINT"{3 DOWN}{6 RIGHT}GAME OVER!"
                                      :rem 246
6075 PRINT"{2 DOWN}YOU MADE{RVS}";M4;"DOL
     LARS"                            :rem 55
6077 PRINT"{2 DOWN}{3 SPACES}PLAY AGAIN (
     Y/N)"                            :rem 132
6080 GETZ$:IFZ$=""OR(Z$<>"Y"ANDZ$<>"N")TH
     EN6080                           :rem 213
6090 IFZ$="N"THENEND                  :rem 177
6095 M1=0:M2=0:M3=0:M4=0:GOTO150      :rem 61
7000 X=FNA(20):Y=FNA(19)              :rem 68
7010 IFPEEK(FNL(0))<>31THEN7040       :rem 6
7030 POKEFNL(0),0:POKEFNL(0)+C,3      :rem 77
7040 RETURN                           :rem 171
8000 IFRU=1THENRETURN                 :rem 123
8005 M4=M4+M3                         :rem 183
8010 POKE214,19:PRINT:POKE211,1:PRINT"
     {LEFT}{16 SPACES}"               :rem 238
8015 POKE214,20:PRINT:POKE211,1:PRINT"
     {WHT}{LEFT}SAFE IN DEPOT";M4;:rem 25
8020 M3=0:POKEP%,CH:POKEP%+C,15:POKEZ,40:
     POKEZ+C,5                        :rem 15
8025 IFM4>200THEN8300                 :rem 169
8030 FORJ=1TO5:FORI=15TO0STEP-1:POKE36878
     ,I:POKE36876,230                 :rem 226
8035 FORT=1TO10:NEXTT,I:POKE36876,0:NEXTJ
                                      :rem 85
8040 IFCF%=0THENPRINTA$;B$:PRINTB$:PRINTA
     $;"{WHT}CABBY, GO TO THE":PRINTF$(XF
     %):GOTO630                       :rem 13
8050 IFCF%<>0THENPRINTA$;B$:PRINTB$:PRINT
     A$;"{WHT}TAKE ME TO THE":PRINTF$(CF%
     ):GOTO630                        :rem 202
8300 POKE36869,240:PRINT"{CLR}":PRINT"
     {5 DOWN}{WHT}YOU HAVE SAVED ENOUGH"
                                      :rem 41
8301 PRINT"{DOWN}{3 SPACES}TO BUY YOUR CA
     B"                               :rem 86
8400 PRINT"{3 DOWN}{6 RIGHT}{WHT}YOU WIN!
     !":FORI=1TO6:POKE36879,107:FORT=1TO5
     0:NEXTT:GOTO6077                 :rem 158
8450 POKE36879,15:FORT=1TO50:NEXTT,I
                                      :rem 208
9000 POKEZ,CH:POKEZ+C,15:POKEP%,32:rem 59
9005 GS=1400                          :rem 106
9020 FORI=1TO10:POKE36878,15:FORT=230TO28
     0STEP10:POKE36876,100+T/3        :rem 174
9025 NEXTT:POKE36876,0:POKE36878,0:FORT=1
     TO55:NEXTT,I                     :rem 61
9100 POKE P%,CH:POKEP%+C,1:POKEZ,42:POKEZ
     +C,7                             :rem 183
9150 GOTO630                          :rem 161
9600 IFRU=1THENRETURN                 :rem 130
9603 PRINTA$;B$:PRINTB$               :rem 146
9605 PRINTA$;"{WHT}{E}{5 SPACES}F{":PRINT"
     {7 SPACES}{";                    :rem 151
9610 A1=INT(GS/200)                   :rem 22
9620 PRINTA$;TAB(A1);"{RED}{":PRINTTAB(A1
     );"{RED}{"                       :rem 237
9630 FORT=1TO1000:NEXTT               :rem 172
9640 IFCF%<>0THENPRINTA$;B$:PRINTB$:PRINT
     A$;"{WHT}TAKE ME TO THE":PRINTF$(CF%
     ):GOTO9660                       :rem 12
9650 IFXF%<>0THENPRINTA$;B$:PRINTB$:PRINT
     A$;"{WHT}CABBY GO TO":PRINTF$(XF%)
                                      :rem 78
9660 RETURN                           :rem 181
9700 FORII=1TO9:PRINTA$;B$:PRINTB$
                                      :rem 141
9750 PRINTA$;"{CYN}{RVS}OUT OF GAS":GOSUB
     5000:NEXTII:GS=700               :rem 145
9760 PRINTA$;B$:PRINTB$:PRINTA$;"BACK WIT
     H 1/2 TANKFUL";:FORDL=1TO2500:NEXT
                                      :rem 114
9770 XF%=0:CF%=0:RETURN               :rem 118
10001 DATA 7384,40,105,125,60,60,125,105,
      40                              :rem 50
10002 DATA 7392,68,68,190,190,190,190,68,
      68                              :rem 98
10003 DATA7400,17,17,190,190,190,190,17,1
      7                               :rem 65
10004 DATA 7408,0,8,42,42,42,8,4,4
                                      :rem 124
10005 DATA 7416,255,255,255,255,255,255,2
      55,255                          :rem 34
10006 DATA 7168,60,60,25,127,124,124,24,6
      0                               :rem 14
10007 DATA 7456,64,32,15,255,131,253,253,
      108                             :rem 123
10008 DATA 7464,2,4,8,255,193,191,191,54
                                      :rem 190
10009 DATA 7472,30,24,23,23,64,87,151,30
                                      :rem 174
10010 DATA 7480,30,151,87,64,23,23,24,30
                                      :rem 165
10011 DATA 7488,0,255,170,255,136,136,136
      ,136                            :rem 176
10012 DATA 7496,0,255,171,255,143,143,141
      ,141                            :rem 165
10013 DATA 7504,15,25,41,47,47,47,31,15
                                      :rem 124
10015 DATAAIRPORT,BANK,"CURLING RINK","DO
      CTORS OFFICE","EMPLOYMENT OFFICE",F
      ACTORY                          :rem 214
10016 DATAGROCERY STORE,HOSPITAL,INN,JEWE
      LER,KENNEL,LIBRARY,MOTEL,"NIGHT CLU
      B",OFFICE                       :rem 84
10017 DATA "POST OFFICE",QUARRY,RESTAURAN
      T,SCHOOL,THEATER,UNIVERSITY,"VETS"
                                      :rem 187
10018 DATA"WEATHER OFFICE","XRAY OFFICE",
      YMCA,ZOO                        :rem 238
```

Program 2: Cabby (64 Version)

```
5 POKE53281,11:POKE53280,11:PRINTCHR$(142
  ):POKE52,56:POKE56,56:CLR        :rem 139
10 PRINT"{CLR}":DIM A%(26),F$(26),TL%(12)
   ,T1%(12),LC%(70),CM%(26)        :rem 168
15 POKE214,12:PRINT:POKE211,5:PRINT"{WHT}
   SETTING UP ........PLEASE WAIT":rem 18
20 FORI=1TO26:READF$(I):NEXTI       :rem 39
50 POKE56334,PEEK(56334)AND254:POKE1,PEEK
   (1)AND251                        :rem 134
51 FORI=0TO511:POKEI+14336,PEEK(I+53248):
   POKEI+15360,PEEK(I+54272):NEXT:rem 166
52 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)
   OR1                              :rem 86
55 FORI=1TO12:READZ:FORJ=ZTOZ+7:READK:POK
   EJ,K:NEXTJ,I                     :rem 34
100 FORI=1TO26:READA%(I):NEXT:FORI=1TO8:R
    EADV(I):NEXT                    :rem 123
110 FORI=1TO26:IFI=1THENCM%(I)=A%(I):GOTO
    150                             :rem 127
120 CM%(I)=CM%(I-1)+A%(I)           :rem 247
150 NEXT:H=1024:C=54272:S=C:CH=28:FORI=1T
    O5:READB%(I),E%(I):NEXT         :rem 66
160 FORI=1TO12:READT1%(I):NEXT      :rem 78
165 DEFFNA(X)=INT(RND(1)*X)+1:DEFFNL(Q)=H
    +40*Y+X                         :rem 75
167 DEFFND(ZZ)=ABS((ZZ=39)+(ZZ=-39)+(ZZ=4
    1)+(ZZ=-41))                    :rem 51
170 FORI=1TO5:A=FNA(2):D%(I)=(A=1)-(A=2):
    NEXT                            :rem 63
175 FORI=1TO5:TR(I)=(D%(I)=-1)*(-E%(I))-(
    D%(I)=1)*(B%(I))                :rem 208
176 VR(I)=(TR(I)=E%(I))-(TR(I)=B%(I)):IFI
    >2THENVR(I)=VR(I)*40            :rem 235
177 TY=(VR(I)=1)*36+(VR(I)=-1)*37+(VR(I)=
    40)*39+(VR(I)=-40)*38:TY(I)=(-1)*TY
                                    :rem 100
180 ED(I)=(TR(I)=B%(I))*(-E%(I))+(TR(I)=E
    %(I))*(-B%(I)):NEXT             :rem 147
185 PRINT"{CLR}":POKE214,8:PRINT:POKE211,
    13:PRINT"{WHT}SELECT A LEVEL"  :rem 67
187 POKE214,10:PRINT:POKE211,16:PRINT"1)
    {SPACE}EASY"                    :rem 225
189 POKE214,12:PRINT:POKE211,16:PRINT"2)
    {SPACE}HARD"                    :rem 211
190 GETZ$:LV=VAL(Z$):RD=RND(1):IFZ$=""THE
    N190                            :rem 199
191 IFLV<1ORLV>2THEN190             :rem 181
195 PRINT"{CLR}":POKE53272,(PEEK(53272)AN
    D240)OR14                       :rem 212
197 FORMC=STOS+24:POKEMC,0:NEXT:POKES+24,
    15:POKES+5,120:POKES+6,240:HF=S+1:LF=
    S                               :rem 166
200 PRINT"{CLR}":GOSUB3005:FORI=1TO12:A=T
    1%(I):GOSUB3200:NEXT            :rem 74
210 P%=H+81:POKEP%,CH:POKEP%+C,7    :rem 101
220 M1=0:M2=0:M3=0:M4=0:XF%=0:CF%=0:GS=14
    00                              :rem 137
300 GOSUB1000:IFEN=1THEN6000        :rem 147
310 IFXF%=0THENGOSUB2000            :rem 196
320 GOSUB400:IFEN=1THEN6000         :rem 104
330 GOSUB7000:GOSUB500:IFEN=1THEN6000
                                    :rem 235
340 GOSUB1000:IFEN=1THEN6000        :rem 151
350 GOSUB5000:IFEN=1THEN6000        :rem 156
360 GOSUB500:IFEN=1THEN6000         :rem 109
390 GOTO300                         :rem 104
400 FORT=1TO5:A=TR(T):TR(T)=TR(T)+VR(T)
                                    :rem 157
410 IFTR(T)=ED(T)THENVR(T)=(-1)*VR(T):GOS
    UB700:GOTO440                   :rem 21
420 IFPEEK(TR(T))>=36ANDPEEK(TR(T))<=39TH
    ENTR(T)=TR(T)-VR(T):GOTO450     :rem 203
430 IFPEEK(TR(T))=27ORPEEK(TR(T))=28THENE
    N=1:T=5:GOTO450                 :rem 210
440 POKEA,32:POKETR(T),TY(T):POKETR(T)+C,
    15:GOSUB9810:IFN4=1THEN450      :rem 140
445 GOSUB500:IFEN=1THENT=5          :rem 113
450 NEXT:RETURN                     :rem 242
500 JV=PEEK(56320):JV=15-(JVAND15):FR=PEE
    K(56320)AND16                   :rem 123
505 IFFR=0ANDSH=0THENGOSUB9600      :rem 135
510 DY=(JV=1)+(JV=5)+(JV=9)-(JV=6)-(JV=10
    )-(JV=2)                        :rem 201
520 DX=(JV=4)+(JV=5)+(JV=6)-(JV=9)-(JV=10
    )-(JV=8)                        :rem 210
525 IF(DX=0ANDDY=0)ORGS=0THENRETURN
                                    :rem 25
535 CH=28:IFDX<>0THENCH=28          :rem 63
540 IFDY<>0THENCH=27                :rem 207
550 ZZ=40*DY+DX:Z=P%+ZZ:PK=PEEK(Z)
                                    :rem 132
555 IFFND(ZZ)=1AND(PK=32ORPK=160)THEN625
                                    :rem 182
557 IFSH=1THEN800                    :rem 0
560 CP=(PK=30)+(PK=32)*2+(PK=40)*3+(PK=41
    )*4+(PK=42)*5+(PK=160)*6+(PK=0)*7
                                    :rem 129
570 ONABS(CP)GOTO600,620,8000,8000,9000,6
    25,3300                         :rem 30
580 IFPEEK(Z)>=129ANDPEEK(Z)<=154THEN4000
                                    :rem 188
590 IFPEEK(Z)>=36ANDPEEK(Z)<=39THENEN=1:R
    ETURN                           :rem 179
600 IF(PEEK(Z+C)AND15)=5THEN620     :rem 38
605 GOTO625                         :rem 113
620 GOSUB9820:POKEP%,32:POKEZ,CH:POKEZ+C,
    7:P%=Z                          :rem 174
625 GS=GS-2:IFGS=<0THENGS=0:GOSUB9700
                                    :rem 57
630 RETURN                          :rem 121
700 TY=(VR(T)=1)*36+(VR(T)=-1)*37+(VR(T)=
    40)*39+(VR(T)=-40)*38:TY(T)=(-1)*TY
                                    :rem 147
710 ED=(ED(T)=B%(T))*(-E%(T))+(ED(T)=E%(T
    ))*(-B%(T)):ED(T)=ED            :rem 181
720 RETURN                          :rem 121
800 IFPEEK(Z)=30AND(PEEK(Z+C)AND15)=5THEN
    620                             :rem 107
810 IFPEEK(Z)<>32THEN625            :rem 163
820 GOTO620                         :rem 107
1000 FORLL=1TO12:PK=PEEK(TL%(LL)+C)AND15
                                    :rem 145
1005 IFPEEK(TL%(LL))=32THENPOKETL%(LL),30
     :POKETL%(LL)+C,PK               :rem 2
1010 POKETL%(LL)+C,PK:TL=(PK=2)*1+(PK=5)*
     2+(PK=7)*3                     :rem 110
1012 GOSUB500:IFEN=1THENLL=12:GOTO1030
                                    :rem 65
1015 GOSUB400:IFEN=1THENLL=12:GOTO1030
                                    :rem 67
1020 GOSUB5000:IFEN=1THENLL=12:GOTO1030
                                    :rem 112
1025 ONABS(TL)GOSUB1040,1050,1060 :rem 86
1030 NEXT:RETURN                    :rem 29
1040 A=FNA(3):IFA=1THENPOKETL%(LL)+C,5:PO
     KETL%(LL),30                   :rem 149
1045 RETURN                         :rem 170
1050 A=FNA(3):IFA=2THENPOKETL%(LL)+C,7:PO
     KETL%(LL),30                   :rem 153
```

```
1055 RETURN                              :rem 171
1060 A=FNA(3):IFA=3THENPOKETL%(LL)+C,2:PO
     KETL%(LL),30                        :rem 150
1065 RETURN                              :rem 172
1500 FORI=22TO23:FORJ=1TO37              :rem 133
1510 POKE214,I:PRINT:POKE211,J:PRINTCHR$(
     32);:NEXTJ,I:RETURN                 :rem 71
2000 PX=FNA(26):XF%=PX+128               :rem 221
2005 GOSUB1500:POKE214,22:PRINT:POKE211,1
                                         :rem 190
2010 PRINT"{WHT}CABBY GO TO THE ";F$(XF%-
     128);:PT=PX:CL=0:X=1:GOSUB5155
                                         :rem 83
2020 GOSUB500:RETURN                     :rem 243
3005 I=49152:IF PEEK(49154)=216THENSYS491
     60:GOTO3030                         :rem 218
3010 READ A:IF A=256 THENSYS49160:GOTO303
     0                                   :rem 112
3020 POKE I,A:I=I+1:GOTO 3010            :rem 70
3030 POKE1064,40:POKE1065,41:POKE1066,40:
     POKE1067,41:FORI=1064TO1067:POKEI+C,
     1:                                  :rem 62
3040 NEXT:POKE1104+C,1:POKE1104,40:FORI=1
     105TO1107:POKEI,32:POKEI+40,32:NEXT
                                         :rem 39
3050 POKE1144,40:POKE1144+C,1:FORI=1265TO
     1301:POKEI,32:NEXT                  :rem 232
3055 FORI=1545TO1581:POKEI,32:NEXT:rem 55
3060 FORI=1114TO1754STEP40:POKEI,32:POKEI
     +10,32:POKEI+20,32:NEXT             :rem 108
3065 FORI=1TO5                           :rem 68
3067 X=FNA(38):Y=FNA(19):L=FNL(Q):rem 155
3070 IFPEEK(L)=32ORPEEK(L)<>160THEN3067
                                         :rem 48
3075 POKEL,42:POKEL+C,7:NEXT             :rem 219
3080 K=0:FORI=1TO26:FORJ=1TOA%(I):rem 226
3090 X=FNA(40):Y=FNA(19):L=FNL(Q):rem 144
3100 IF PEEK(L)=32ORPEEK(L)<>160THEN3090
                                         :rem 38
3110 IFPEEK(L+1)=160ANDPEEK(L-1)=160ANDPE
     EK(L+40)=160ANDPEEK(L-40)=160THEN309
     0                                   :rem 250
3120 K=K+1:LC%(K)=L:POKEL,I+128:NEXT:NEXT
     :RETURN                              :rem 4
3200 IFPEEK(A)<>32THENGOSUB3220          :rem 96
3210 TL%(I)=A:POKEA,30:POKEA+C,5:RETURN
                                         :rem 111
3220 FORJ=1TO8                           :rem 65
3230 IFPEEK(A+V(J))=32THENA=A+V(J):K=8
                                         :rem 208
3240 NEXT:RETURN                         :rem 34
3300 IFXF%<>0THENRETURN                  :rem 209
3310 POKEP%,32:POKEZ,CH:POKEZ+C,7:GOSUB20
     00                                  :rem 134
3320 POKEP%,CH:POKEP%+C,7:POKEZ,160:POKEZ
     +C,1:MN=1:RETURN                    :rem 67
3330 M1=ABS(SP-Z)/10+1:POKEP%,32:POKEZ,CH
     :POKEZ+C,7:GOSUB9820                :rem 203
3340 M2=INT(FNA(100*M1)/10)/100          :rem 113
3350 GOSUB1500:POKE214,22:PRINT:POKE211,1
                                         :rem 194
3360 PRINT"YOU COLLECT";M1;"PLUS TIP OF";
     M2;                                 :rem 147
3370 M3=M1+M2+M3:M1=0:M2=0:POKEP%,CH:POKE
     P%+C,7:POKEZ,XF%:POKEZ+C,1          :rem 44
3375 PT=PX:CL=1:X=1:GOSUB5155            :rem 28
3380 POKE214,23:PRINT:POKE211,1:PRINT"YOU
      HAVE";M3;:MN=0:XF%=0:CF%=0:RETURN
                                         :rem 131
4000 IFMN=1THEN3330                      :rem 84
4005 IFCF%<>0THEN4500                    :rem 168
4010 IFPEEK(Z)<>XF%THEN6300              :rem 41
4030 IFLV=2ANDZ<>DZTHEN6300              :rem 116
4050 PC=FNA(26):CF%=PC+128               :rem 165
4200 SP=Z:POKEZ,CH:POKEZ+C,7:POKEP%,32:GO
     SUB9820                             :rem 10
4210 IFLV=1THENPT=PC:CL=1:GOSUB5155
                                         :rem 202
4260 GOSUB1500:POKE214,22:PRINT:POKE211,1
                                         :rem 195
4265 PRINT"{WHT}TAKE ME TO THE ";F$(CF%-1
     28);:PT=PC:CL=0:X=2:GOSUB5155
                                         :rem 248
4280 POKEP%,CH:POKEP%+C,7:GOSUB9820:POKEZ
     ,XF%:POKEZ+C,1                      :rem 165
4285 IFCF%=XF%THENPOKEZ+C,0              :rem 146
4290 IFLV=1ANDXF%<>CF%THENPT=PX:CL=1:GOSU
     B5155                               :rem 165
4300 GOTO6300                            :rem 153
4500 IFPEEK(Z)<>CF%THEN6300               :rem 24
4505 IFLV=2ANDZ<>DCTHEN6300              :rem 100
4510 POKEZ,CH:POKEZ+C,7:POKEP%,32:GOSUB98
     20                                  :rem 154
4550 M1=ABS(SP-Z)/10+1                   :rem 198
4560 M2=INT(FNA(100*M1)/10)/100          :rem 118
4570 GOSUB1500:POKE214,22:PRINT:POKE211,1
                                         :rem 199
4573 PRINT"YOU COLLECT";M1;"PLUS TIP OF";
     M2;                                 :rem 154
4580 M3=M1+M2+M3:M1=0:M2=0               :rem 172
4600 POKE214,23:PRINT:POKE211,1:PRINT"YOU
      HAVE";M3;                          :rem 100
4650 POKEP%,CH:POKEP%+C,7:GOSUB9820:POKEZ
     ,CF%:POKEZ+C,1                      :rem 145
4655 IFLV=1THENPT=PC:CL=1:GOSUB5155
                                         :rem 215
4660 CF%=0:XF%=0:GOTO6300                 :rem 97
5000 MX=0:MY=0:GOTO5010                   :rem 93
5005 MY=INT((P%-H)/40):MX=(P%-H)-40*MY:RE
     TURN                                :rem 137
5010 GOSUB5005:CT=ABS((MY=6)*1+(MY=13)*2)
     :T=CT:IFCT>0THENGOSUB5050           :rem 210
5015 GOSUB5005:CT=ABS((MX=10)*3+(MX=20)*4
     +(MX=30)*5):T=CT:IFCT>0THENGOSUB5050
                                         :rem 34
5020 RETURN                              :rem 167
5050 IFP%>TR(CT)ANDVR(CT)>0THEN5080
                                         :rem 103
5060 IFP%<TR(CT)ANDVR(CT)<0THEN5080
                                         :rem 100
5070 VR(CT)=(-1)*VR(CT):GOSUB700         :rem 83
5080 B=FNA(2):DN=(B=1)*(-P%)-(B=2)*(ED(CT
     ))                                  :rem 230
5100 POKETR(CT),32:FORMV=TR(CT)TODNSTEPVR
     (CT):PK=PEEK(MV)                    :rem 207
5110 IFPK>=36ANDPK<=39THENTR(CT)=MV-VR(CT
     ):MV=DN:NEXT:GOSUB5150:RETURN:rem 70
5115 IFPK=27ORPK=28THENMV=DN:NEXT:EN=1:RE
     TURN                                :rem 238
5120 POKEMV,TY(CT):POKEMV+C,15:SH=1
                                         :rem 170
5125 GOSUB9810:GOSUB500:SH=0:POKEMV,32
                                         :rem 77
5126 IFEN=1THENMV=DN                     :rem 255
5130 NEXT:IFEN=1THENRETURN               :rem 225
5135 IFDN=ED(CT)THENTR(CT)=ED(CT)-VR(CT):
     POKEED(CT),TY(CT):RETURN            :rem 122
5140 TR(CT)=DN:POKETR(CT),TY(CT):RETURN
                                         :rem 249
5150 POKETR(CT),TY(CT):GOSUB500:RETURN
                                         :rem 178
5155 IFLV=2THENONXGOTO5160,5170           :rem 146
```

```
5156 FORLC=CM%(PT-1)+1TOCM%(PT)    :rem 53
5157 POKELC%(LC)+C,CL:NEXT:RETURN  :rem 81
5160 R=FNA(A%(PT)):DZ=LC%(CM%(PT-1)+R):PO
     KEDZ+C,0:RETURN               :rem 102
5170 R=FNA(A%(PT)):DC=LC%(CM%(PT-1)+R):PO
     KEDC+C,0:RETURN               :rem 57
6000 PRINT"{CLR}":EN=0:GOSUB9830:POKE5327
     2,21                          :rem 237
6050 POKE214,8:PRINT:POKE211,6:PRINT"
     {WHT}YOUR CAB LOST TO THE TROLLEY"
                                   :rem 49
6070 POKE214,10:PRINT:POKE211,15:PRINT"GA
     ME OVER!"                     :rem 248
6075 POKE214,12:PRINT:POKE211,8:PRINT"YOU
      MADE {RVS}";M4;"{OFF}DOLLARS"
                                   :rem 94
6080 PRINT"{2 DOWN}{12 RIGHT}PLAY AGAIN (
     Y/N)"                         :rem 218
6085 GETZ$:RD=RND(1):IFZ$=""OR(Z$<>"Y"AND
     Z$<>"N")THEN6085              :rem 82
6090 IFZ$="N"THENEND               :rem 177
6095 PRINT"{CLR}":RESTORE:GOTO165:rem 165
7000 X=FNA(40):Y=FNA(19)           :rem 70
7010 IFPEEK(FNL(Q))<>160THEN7040   :rem 90
7030 POKEFNL(Q),0:POKEFNL(Q)+C,3   :rem 143
7040 RETURN                        :rem 171
8000 M4=M4+M3:GOSUB9820            :rem 63
8010 GOSUB1500:POKE214,22:PRINT:POKE211,1
     :PRINT"{WHT}SAFE IN DEPOT";M4
                                   :rem 190
8020 M3=0:POKEP%,CH:POKEP%+C,7:POKEZ,40:P
     OKEZ+C,1:FORDL=1TO500:NEXT    :rem 172
8025 IFM4>200THEN8300              :rem 169
8040 IFCF%<>0THEN8050              :rem 175
8041 IFXF%=0THEN630                :rem 84
8043 GOSUB1500:POKE214,22:PRINT:POKE211,1
                                   :rem 198
8045 PRINT"{WHT}CABBY GO TO THE ";F$(XF%-
     128);:N4=1:GOSUB400:N4=0      :rem 127
8046 RETURN                        :rem 178
8050 GOSUB1500:POKE214,22:PRINT:POKE211,1
                                   :rem 196
8060 PRINT"{WHT}TAKE ME TO THE ";F$(CF%-1
     28);:N4=1:GOSUB400:N4=0       :rem 39
8070 RETURN                        :rem 175
8300 PRINT"{CLR}":POKE214,8:PRINT:POKE211
     ,9:PRINT"{WHT}YOU HAVE SAVED ENOUGH"
                                   :rem 38
8310 POKE214,10:PRINT:POKE211,11:PRINT"TH
     E CAB IS YOURS"               :rem 97
8400 PRINT"{2 DOWN}{16 RIGHT}YOU WIN":GOS
     UB9840                        :rem 9
8450 POKE53272,21:GOTO6080         :rem 211
9000 POKEZ,CH:POKEZ+C,15:POKEP%,32:GS=140
     0                             :rem 17
9100 POKE P%,CH:POKEP%+C,1:POKEZ,42:POKEZ
     +C,7                          :rem 183
9150 RETURN                        :rem 175
9600 GOSUB1500:POKE214,22:PRINT:POKE211,1
                                   :rem 198
9605 PRINT"{WHT}{RVS} {OFF}E{5 SPACES}F
     {RVS} {OFF}"                  :rem 125
9610 A1=INT(GS/200):POKE214,23:PRINT:POKE
     211,A1+1:PRINT"{CYN}"CHR$(95);
                                   :rem 10
9630 FORDL=1TO500:NEXT:IFCF%=0THEN9645
                                   :rem 83
9635 GOSUB1500:POKE214,22:PRINT:POKE211,1
                                   :rem 206
9640 PRINT"{WHT}TAKE ME TO THE ";F$(CF%-1
     28);:N4=1:GOSUB400:N4=0:RETURN
                                   :rem 70
9645 IFXF%=0THENRETURN             :rem 166
9650 GOSUB1500:POKE214,22:PRINT:POKE211,1
                                   :rem 203
9655 PRINT"{WHT}CABBY GO TO THE ";F$(XF%-
     128);:N4=1:GOSUB400:N4=0:RETURN
                                   :rem 161
9700 GOSUB1500:POKE214,22:PRINT:POKE211,1
                                   :rem 199
9750 PRINT"{WHT}OUT OF GAS":N4=1:GOSUB400
     :N4=0:IFEN=1THENRETURN        :rem 141
9755 GOSUB1500:POKE214,22:PRINT:POKE211,1
                                   :rem 209
9760 PRINT"BACK WITH 1/2 TANKFUL{WHT}";:N
     4=1:GOSUB400:N4=0:IFEN=1THENRETURN
                                   :rem 85
9765 GOSUB5000:IFEN=1THENRETURN    :rem 249
9770 GS=700:GOSUB1500:POKE214,22:PRINT:PO
     KE211,1                       :rem 118
9780 IFXF%<>0THENPRINT"{WHT}CABBY GO TO T
     HE ";F$(XF%-128);:RETURN      :rem 42
9790 IFCF%<>0THENPRINT"{WHT}TAKE ME TO TH
     E ";F$(CF%-128);:RETURN       :rem 193
9800 RETURN                        :rem 177
9810 POKES+4,17:POKEHF,40:POKELF,250:POKE
     S+4,16:RETURN                 :rem 84
9820 POKES+4,33:POKEHF,50:POKELF,100:POKE
     S+4,32:RETURN                 :rem 76
9830 POKES+4,33:FORI=200TO70STEP-5:POKEHF
     ,I:POKELF,INT(I/2):NEXT:POKES+4,32
                                   :rem 198
9835 POKES+24,0:RETURN             :rem 98
9840 POKES+4,17:POKEHF,40:POKELF,200:FORI
     =1TO10:FORJ=70TO200STEP5:POKEHF,J
                                   :rem 136
9850 POKELF,90:NEXT:G=G+1:IFG<5THEN9840
                                   :rem 139
9860 POKES+4,16:POKES+24,0:RETURN :rem 14
9900 DATAAIRPORT,BANK,"CURLING RINK","DOC
     TORS OFFICE","EMPLOYMENT OFFICE"
                                   :rem 109
9910 DATAFACTORY,"GROCERY STORE",HOSPITAL
     ,INN                          :rem 190
9915 DATA"JEWELRY STORE",KENNEL,LIBRARY,M
     OTEL                          :rem 188
9920 DATA"NIGHT CLUB",OFFICE,"POST OFFICE
     ",QUARRY,RESTAURANT,SCHOOL,THEATER
                                   :rem 126
9930 DATA UNIVERSITY,"VETS","WEATHER OFFI
     CE","XRAY OFFICE",YMCA,ZOO    :rem 201
10001 DATA 14552,24,255,255,24,24,255,255
      ,24                          :rem 117
10002 DATA 14560,0,102,126,255,255,126,10
      2,0                          :rem 93
10004 DATA 14576,0,8,42,42,42,8,4,4
                                   :rem 176
10005 DATA 14584,8,28,62,28,28,28,28,28
                                   :rem 146
10006 DATA 14336,60,60,25,127,124,124,24,
      60                           :rem 57
10007 DATA 14624,64,32,15,255,131,253,253
      ,108                         :rem 166
10008 DATA 14632,2,4,8,255,193,191,191,54
                                   :rem 233
10009 DATA 14640,30,24,23,23,64,87,151,30
                                   :rem 217
10010 DATA 14648,30,151,87,64,23,23,24,30
                                   :rem 217
10011 DATA 14656,0,255,170,255,136,136,13
      6,136                        :rem 219
10012 DATA 14664,0,255,171,255,143,143,14
      1,141                        :rem 208
```

```
10013 DATA 14672,15,25,41,47,47,47,31,15
                                    :rem 176
10014 DATA 1,5,2,3,1,2,5,1,2,4,1,1,6,7,5,
      2,1,6,3,5,1,2,1,1,1,1       :rem 130
10015 DATA1,-1,40,-40,-39,39,-41,41
                                    :rem 197
10016 DATA1265,1301,1545,1581,1114,1754,1
      124,1764,1134,1774          :rem 157
10260 DATA1151,1159,1169,1179,1429,1439,1
      449,1459,1669,1679,1689,1699
                                    :rem 214
10270 DATA 1,0,216,255,255,255,40 :rem 78
10280 DATA 0,169,81,133,251,169,40
                                    :rem 137
10290 DATA 133,253,169,4,133,252,133
                                    :rem 234
10300 DATA 254,169,147,32,210,255,162
                                     :rem 26
10310 DATA 0,160,0,169,160,145,253
                                    :rem 121
10320 DATA 200,192,39,208,249,24,165
                                    :rem 237
10330 DATA 253,105,40,133,253,144,2
                                    :rem 169
10340 DATA 230,254,232,224,19,208,229
                                     :rem 26
10350 DATA 160,0,169,4,145,251,169
                                    :rem 136
10360 DATA 255,141,15,212,169,128,141
                                     :rem 26
10370 DATA 18,212,173,27,212,41,3 :rem 76
10380 DATA 133,173,170,10,168,24,185
                                    :rem 236
10390 DATA 0,192,101,251,133,170,185
                                    :rem 226
10400 DATA 1,192,101,252,133,171,24
                                    :rem 165
10410 DATA 185,0,192,101,170,133,253
                                    :rem 221
10420 DATA 185,1,192,101,171,133,254
                                    :rem 225
10430 DATA 160,0,177,253,201,160,208
                                    :rem 222
10440 DATA 18,138,145,253,169,32,145
                                    :rem 243
10450 DATA 170,165,253,133,251,165,254
                                     :rem 80
10460 DATA 133,252,76,62,192,232,138
                                    :rem 241
10470 DATA 41,3,197,173,208,189,177
                                    :rem 205
10480 DATA 251,170,169,32,145,251,224
                                     :rem 30
10490 DATA 4,240,26,138,10,168,162
                                    :rem 135
10500 DATA 2,56,165,251,249,0,192 :rem 84
10510 DATA 133,251,165,252,249,1,192
                                    :rem 233
10520 DATA 133,252,202,208,238,76,62
                                    :rem 234
10530 DATA 192,169,1,160,0,153,0  :rem 24
10540 DATA 216,153,0,217,153,0,218
                                    :rem 124
10550 DATA 153,0,219,200,208,241,96,256
                                    :rem 125
```

First Aid

(Article on page 95.)

```
10 REM ***FIRST AID***              :rem 151
20 PRINT"{CLR}":POKE53280,0:POKE53281,0:P
   OKE646,3                         :rem 189
25 PRINT"{5 DOWN}{10 SPACES}PLEASE STAND
   {SPACE}BY"                       :rem 92
30 DIMM$(50),A$(50,100)             :rem 104
40 A=1                               :rem 19
50 READM$(A):IFM$(A)=""THEN70       :rem 103
60 A=A+1:GOTO50                      :rem 89
70 A=1:B=1                           :rem 0
80 READA$(A,B)                      :rem 233
90 IFA$(A,B)="9"THENGOTO120          :rem 18
100 IFA$(A,B)=""THENA=A+1:B=1:GOTO80
                                    :rem 21
110 B=B+1:GOTO80                    :rem 138
120 PRINT"{CLR}":PRINT"{UP}{10 RIGHT}
    {RVS}FIRST AID SELECTION{DOWN}"
                                    :rem 212
130 Q=1:S=0:RR=1                    :rem 153
140 FORA=1TO34:G=A                   :rem 52
150 IFA>20THENIFQTHENQ=0:PRINT"{HOME}
    {DOWN}":S=24                     :rem 97
160 IFA>26THENG=-16+A-26             :rem 85
170 PRINT"{YEL}"TAB(S)CHR$(G+64)"{CYN}"SP
    C(1)M$(A)                       :rem 254
180 NEXT                            :rem 216
190 POKE198,0                       :rem 199
200 PRINT"{HOME}{23 DOWN}{RVS}{YEL}ENTER
    {SPACE}LETTER OR NUMBER:{CYN}":rem 62
210 GETA$:IFA$=""THEN210             :rem 73
220 A=ASC(A$)-64                     :rem 54
225 IFASC(A$)<49ORASC(A$)>90THEN210
                                    :rem 245
230 IFA<1THENA=91+(A+15)-64         :rem 206
240 IFA>34ORA<0THEN210               :rem 27
250 PRINT"{CLR}"                    :rem 251
260 FORB=1TO100                      :rem 99
270 IFB>10ANDA=14ANDRR=1THENPRINT"
    {2 DOWN} {YEL}PRESS ANY KEY TO CONTIN
    UE{CYN}":GOSUB370               :rem 189
280 PRINT"{DOWN}"A$(A,B)            :rem 225
290 IFA$(A,B)=""THEN310             :rem 211
300 NEXT                            :rem 210
310 PRINT"{RVS}HIT ANY KEY TO RETURN TO M
    AIN MENU"                       :rem 173
320 POKE198,0                       :rem 194
330 GETA$:IFA$=""THEN330             :rem 79
340 GOTO120                          :rem 99
350 GETA$:IFA$=""THEN350             :rem 83
360 C=VAL(A$)                       :rem 178
370 POKE198,0                       :rem 199
380 GETA$:IFA$=""THEN380             :rem 89
390 PRINT"{CLR}":RR=0:RETURN        :rem 101
400 DATA"{RVS}{YEL}EMERGENCY NUMBERS{CYN}
    "                               :rem 252
410 DATA"APPENDICITIS"              :rem 112
420 DATA"ARTIF. RESPIRATION"        :rem 232
430 DATA"BITES (ANIMAL)","BITES (INSECT)"
     "BITES (SNAKE)"                 :rem 23
440 DATA "BLEEDING (SEVERE)"         :rem 75
450 DATA"BONE & JOINT INJURIES","BRUISES"
                                    :rem 187
460 DATA"BURNS (MINOR)"              :rem 88
470 DATA"BURNS (SERIOUS)","BURNS (V. SERI
    OUS)"                           :rem 247
480 DATA"BURNS (CHEMICAL)","CHOKING","DRU
    GS"                             :rem 115
490 DATA"EARACHE","ELECTRIC SHOCK","EPILE
    PSY"                            :rem 242
500 DATA"EXPOSURE TO COLD","EXPOSURE TO H
    EAT"                            :rem 227
```

```
510 DATA"FAINTING","FEVER","HEART ATTACK"
    ,"HERNIAS","NOSE BLEED"          :rem 67
520 DATA"POISON (GAS)","POISON (ORAL)","S
    HOCK","SPRAINS","STRAINS"        :rem 204
530 DATA "STROKE","SUNBURN","SUNSTROKE","
    TOOTHACHE",""                    :rem 40
540 DATA"{RVS}{YEL}{8 RIGHT}***EMERGENCY
    {SPACE}NUMBERS***{CYN}"          :rem 229
550 DATA"{DOWN}POISON CONTROL CENTER:
    {2 SPACES}798-6200":DATA"DOCTOR:
    {2 SPACES}823-4796"              :rem 219
560 DATA"AMBULANCE:{2 SPACES}429-5111"
                                     :rem 79
570 DATA"DOCTOR:{2 SPACES}823-4796"
                                     :rem 163
580 DATA"HOSPITAL:{2 SPACES}823-1000"
                                     :rem 36
590 DATA"POLICE:{2 SPACES}429-8231"
                                     :rem 140
600 DATA"FIRE:{2 SPACES}823-2233",""
                                     :rem 88
610 DATA"{RVS}{12 RIGHT}APPENDICITIS","
    {DOWN}{RVS}1){OFF} CALL YOUR DOCTOR."
                                     :rem 195
620 DATA"{RVS}2){OFF} NEVER GIVE ANYTHING
     BY MOUTH."                      :rem 88
630 DATA"{RVS}3){OFF} ICE BAG MAY REDUCE
    {SPACE}DISCOMFORT.",""           :rem 201
640 DATA"{RVS}{6 RIGHT}ARTIFICIAL RESPIRA
    TION"                            :rem 224
650 DATA"{DOWN}{RVS}1){OFF} TILT HEAD BAC
    K."                              :rem 150
660 DATA"{RVS}2){OFF} PINCH NOSE SHUT. BL
    OW AIR INTO{10 SPACES}VICTIM'S MOUTH.
    "                                :rem 93
670 DATA"{RVS}3){OFF} REMOVE MOUTH;LOOK F
    OR EXHALATION."                  :rem 200
680 DATA"{RVS}4){OFF} REPEAT BLOWING CYCL
    E."                              :rem 110
690 DATA"{DOWN}{RVS}ADULT{OFF} - BREATHE
    {SPACE}DEEPLY EVERY 5 SECONDS."
                                     :rem 20
700 DATA"{RVS}CHILD{OFF} - BREATHE GENTLY
     EVERY 3 SECONDS.",""            :rem 99
710 DATA"{RVS}{13 RIGHT}ANIMAL BITES","
    {DOWN}{RVS}1){OFF} STOP BLEEDING."
                                     :rem 215
720 DATA"{DOWN}{RVS}2){OFF} WASH WOUND WI
    TH SOAP & WATER."                :rem 13
730 DATA"{DOWN}{RVS}3){OFF} RESTRICT MOVE
    MENT OF AFFECTED PART."          :rem 16
740 DATA"{DOWN}{RVS}4){OFF} ISOLATE ANIMA
    L FOR RABIES TEST.",""           :rem 73
750 DATA"{RVS}{13 RIGHT}INSECT BITES","
    {RVS}MINOR:","{RVS}1){OFF}"      :rem 113
760 DATA"{2 UP}{3 RIGHT}APPLY COLD SOOTHI
    NG LOTIONS(CALAMINE)"            :rem 58
770 DATA"{RVS}SEVERE REACTIONS:","{RVS}1)
    {OFF} GIVE ARTIFICIAL RESPIRATION"
                                     :rem 123
780 DATA"{UP}{3 RIGHT}IF NECESSARY."
                                     :rem 79
790 DATA"{RVS}2){OFF} APPLY CONSTRICTING
    {SPACE}BAND"                     :rem 47
800 DATA"{UP}{3 RIGHT}2-4 INCHES ABOVE ST
    ING."                            :rem 75
810 DATA"{RVS}3){OFF} KEEP AFFECTED PART
    {SPACE}DOWN"                     :rem 189
820 DATA"{UP}{3 RIGHT}AND APPLY ICE PACK.
    "                                :rem 87
830 DATA"{RVS}4){OFF} IF ALLERGIC REACTIO
    N -"                             :rem 78
840 DATA"{UP}{3 RIGHT}GET MEDICAL ATTENTI
    ON.",""                          :rem 5
850 DATA"{RVS}{13 RIGHT}SNAKE BITES"
                                     :rem 111
860 DATA"{DOWN}{RVS}1){OFF} GET VICTIM TO
     HOSPITAL."                      :rem 236
870 DATA"{RVS}2){OFF} RESTRICT MOVEMENT O
    F VICTIM."                       :rem 102
880 DATA"{RVS}3){OFF} IMMOBOLIZE AFFECTED
     PART BELOW"                     :rem 215
890 DATA"{UP}{3 RIGHT}LEVEL OF HEART."
                                     :rem 150
900 DATA"{RVS}4){OFF} APPLY CONSTRICTING
    {SPACE}BAND 2-4 INCHES"          :rem 119
910 DATA"{UP}{3 RIGHT}ABOVE BITE & BETWEE
    N BITE & HEART."                 :rem 141
920 DATA"{RVS}5){OFF} MAKE INCISION LENGT
    H-WISE(NOT DEEP)"                :rem 252
930 DATA"{UP}{3 RIGHT}AT FANG MARKS."
                                     :rem 63
940 DATA"{RVS}6){OFF} SUCK VENOM FROM WOU
    ND & SPIT OUT."                  :rem 6
950 DATA"{UP}{3 RIGHT}REPEAT FOR AT LEAST
     40 MINUTES."                    :rem 81
960 DATA"{RVS}7){OFF} TREAT FOR SHOCK.","
    "                                :rem 126
970 DATA"{10 RIGHT}{RVS}SEVERE BLEEDING",
    "{RVS}1){OFF} CALL FOR MEDICAL ASSIST
    ANCE."                           :rem 178
980 DATA"{RVS}2){OFF} LAY VICTIM DOWN & E
    LEVATE LEGS IN A{5 SPACES}SEMI-FLEXED
     POSITION."                      :rem 205
990 DATA"{RVS}3){OFF} CONTROL BLEEDING BY
     APPLYING DIRECT{5 SPACES}PRESSURE TO
     WOUND."                         :rem 236
1000 DATA"{RVS}4){OFF} IF BLEEDING CONTIN
     UES APPLY DIGITAL"              :rem 37
1010 DATA"{UP}{3 RIGHT}PRESSURE AT PRESSU
     RE POINT."                      :rem 71
1020 DATA"{RVS}5){OFF} ELEVATE BLEEDING P
     ART OF BODY{11 SPACES}ABOVE LEVEL OF
      HEART."                        :rem 121
1030 DATA"{RVS}6){OFF} MAINTAIN OPEN AIRW
     AY & GIVE VICTIM{6 SPACES}PLENTY OF
     {SPACE}FRESH AIR."              :rem 133
1040 DATA"{RVS}7){OFF} PREVENT LOSS OF HE
     AT WITH BLANKETS{6 SPACES}OVER & UND
     ER VICTIM.",""                  :rem 29
1050 DATA"{9 RIGHT}{RVS}BONE & JOINT INJU
     RIES"                           :rem 114
1060 DATA"{DOWN}{RVS}1){OFF} KEEP BONE EN
     DS & ADJACENT JOINTS{8 SPACES}STILL.
     "                               :rem 148
1070 DATA"{RVS}2){OFF} DO NOT MOVE VICTIM
      UNLESS ABSOLUTELY{4 SPACES}NECESSAR
     Y."                             :rem 101
1080 DATA"{RVS}3){OFF} APPLY WELL PADDED
     {SPACE}SPLINT TO INJURED{5 SPACES}PA
     RT."                            :rem 118
1090 DATA"{RVS}4){OFF} IF BROKEN BONE IS
     {SPACE}PROTRUDING  CONTROL{3 SPACES}
     BLEEDING."                      :rem 252
1100 DATA"{2 UP}{13 RIGHT}COVER WITH CLEA
     N DRESSING{5 SPACES}BEFORE SPLINTING
     ."                              :rem 209
1110 DATA"{RVS}5){OFF} TREAT FOR SHOCK.",
     ""                              :rem 160
1120 DATA"{15 RIGHT}{RVS}BRUISES"  :rem 4
```

```
1130 DATA"{DOWN}{RVS}1){OFF} APPLY COLD C
     LOTH OR ICE PACK TO{9 SPACES}RELIEVE
      PAIN AND"                      :rem 143
1140 DATA"{2 UP}{19 RIGHT} REDUCE SWELLIN
     G.",""                         :rem 40
1150 DATA"{7 RIGHT}{RVS}MINOR BURNS(1ST D
     EGREE)"                        :rem 230
1160 DATA"{DOWN}{RVS}1){OFF} SUBMERGE RED
     DENED SKIN IN COLD WATER."     :rem 105
1170 DATA"{RVS}2){OFF} APPLY DRY DRESSING
      IF NECESSARY.",""             :rem 212
1180 DATA"{6 RIGHT}{RVS}SERIOUS BURNS(2ND
      DEGREE)"                      :rem 93
1190 DATA"{DOWN}{RVS}1){OFF} SUBMERGE BLI
     STERED SKIN IN COLD{9 SPACES}(NOT IC
     E) WATER."                     :rem 226
1200 DATA"{RVS}2){OFF} APPLY CLEAN CLOTHS
      SOAKED IN ICE{8 SPACES}WATER."
                                    :rem 166
1210 DATA"{RVS}3){OFF} APPLY PROTECTIVE B
     ANDAGE."                       :rem 189
1220 DATA"{RVS}4){OFF} ELEVATE AFFECTED P
     ARTS.",""                      :rem 132
1230 DATA"{5 RIGHT}{RVS}VERY SERIOUS BURN
     S(3RD DEGREE)"                 :rem 135
1240 DATA"{DOWN}{RVS}1){OFF} DO NOT REMOV
     E ADHERED PARTICLES OF{6 SPACES}CLOT
     HING FROM CHARRED"             :rem 52
1250 DATA"{2 UP}{25 RIGHT}SKIN." :rem 128
1260 DATA"{RVS}2){OFF} ELEVATE AFFECTED P
     ARTS ABOVE VICTIM'S{3 SPACES}HEART."
                                    :rem 61
1270 DATA"{RVS}3){OFF} COVER BURNS WITH C
     LEAN THICK{12 SPACES}DRESSINGS-NO OI
     NTMENTS."                      :rem 174
1280 DATA"{RVS}4){OFF} TREAT FOR SHOCK, T
     HEN GET VICTIM TO{5 SPACES}HOSPITAL.
     ",""                           :rem 181
1290 DATA"{3 RIGHT}{9 SPACES}{RVS}CHEMICA
     L BURNS"                       :rem 83
1300 DATA"{DOWN}{RVS}1){OFF} WASH FOR 5 O
     R MORE MINUTES WITH LARGE{3 SPACES}A
     MOUNTS OF WATER."              :rem 141
1310 DATA"{RVS}2){OFF} APPLY CLEAN DRESSI
     NG; GET TO HOSPITAL",""        :rem 252
1320 DATA"{8 RIGHT}{7 SPACES}{RVS}CHOKING
     ","{RVS}1){OFF} IF VICTIM CAN COUGH,
      BREATHE,OR TALK,"             :rem 126
1330 DATA"{2 UP}{3 RIGHT}DO NOTHING."
                                    :rem 118
1340 DATA"{RVS}2){OFF} IF VICTIM IS CLUTC
     HING THROAT,{10 SPACES}ENCOURAGE VIG
     OROUS"                         :rem 210
1350 DATA"{2 UP}{22 RIGHT}COUGHING."
                                    :rem 73
1360 DATA"{RVS}3){OFF} IF VICTIM STILL CA
     NNOT COUGH, BREATHE{3 SPACES}OR TALK
     , GIVE 4 SHARP"                :rem 65
1370 DATA"{UP}{3 SPACES}BLOWS BETWEEN SHO
     ULDER BLADES,"                 :rem 136
1380 DATA"{UP}{3 SPACES}FOLLOWED BY 4 ABD
     OMINAL THRUSTS."               :rem 216
1390 DATA"{UP}{3 SPACES}REPEAT UNTIL OBJE
     CT IS DISLODGED."              :rem 25
1400 DATA"{RVS}4){OFF} IF VICTIM IS A CHI
     LD,HOLD UPSIDE DOWN{3 SPACES}& SLAP
     {SPACE}SHARPLY ON BACK."       :rem 106
1410 DATA"{RVS}5){OFF} IF BREATHING HAS S
     TOPPED & FOREIGN{6 SPACES}MATERIAL C
     ANNOT BE"                      :rem 13
1420 DATA"{2 UP}{22 RIGHT}DISLODGED"
                                    :rem 84
1430 DATA"{UP}{3 SPACES}ADMINISTER ARTIFI
     CIAL RESPIRATION."             :rem 253
1440 DATA"{UP}{3 SPACES}GET PROMPT MEDICA
     L ATTENTION.",""               :rem 189
1450 DATA"{6 SPACES}{11 RIGHT}{RVS}DRUGS"
                                    :rem 254
1460 DATA"{DOWN}{RVS}1){OFF} KEEP AIRWAY
     {SPACE}OPEN; GIVE ARTIFICIAL"
                                    :rem 154
1470 DATA"{UP}{3 RIGHT}VENTILATION OR CPR
      IF NEEDED."                   :rem 71
1480 DATA"{RVS}2){OFF} TREAT FOR SHOCK."
                                    :rem 55
1490 DATA"{RVS}3){OFF} PLACE UNCONSCIOUS
     {SPACE}VICTIM IN A 3"          :rem 193
1491 DATA"{UP}{3 SPACES}QUARTERS PRONE PO
     SITION."                       :rem 92
1500 DATA"{RVS}4){OFF} PROTECT VICTIM FRO
     M INJURY."                     :rem 85
1510 DATA"{RVS}5){OFF} HALLUCINOGENIC VIC
     TIMS MAY NEED{9 SPACES}CAREFUL ATTEN
     TION."                         :rem 52
1520 DATA"{RVS}6){OFF} GET VICTIM TO HOSP
     ITAL.",""                      :rem 122
1530 DATA"{16 RIGHT}{RVS}EARACHE":rem 242
1540 DATA"{DOWN}{RVS}1){OFF} CALL YOUR DO
     CTOR."                         :rem 155
1550 DATA"{RVS}2){OFF} APPLY ICE BAG OR H
     OT WATER BOTTLE;{6 SPACES}WHICHEVER
     {SPACE}GIVES RELIEF."          :rem 101
1560 DATA"","{9 SPACES}{RVS}{2 RIGHT}ELEC
     TRIC SHOCK"                    :rem 169
1570 DATA"{DOWN}{RVS}1){OFF} TURN OFF ELE
     CTRICITY IF POSSIBLE."         :rem 189
1580 DATA"{RVS}2){OFF} REMOVE ELECTRIC CO
     NTACT FROM VICTIM{5 SPACES}WITH DRY
     {SPACE}WOOD OR DRY"            :rem 68
1590 DATA"{2 UP}{24 RIGHT}CLOTH.","{RVS}3
     ){OFF} GIVE ARTIFICIAL RESPIRATION."
                                    :rem 160
1600 DATA"{RVS}4){OFF} KEEP VICTIM WARM."
      "{RVS}5){OFF} CALL YOUR DOCTOR.",""
                                    :rem 194
1610 DATA"{14 RIGHT}{RVS}EPILEPSY":rem 57
1620 DATA"{DOWN}{RVS}1){OFF} PREVENT VICT
     IM FROM INJURING HIMSELF{4 SPACES}BU
     T DO NOT RESTRAIN."            :rem 205
1630 DATA"{RVS}2){OFF} INSERT CLOTH BETWE
     EN TEETH TO PROTECT{3 SPACES}TONGUE.
     "                              :rem 190
1640 DATA"{RVS}3){OFF} LOOSEN TIGHT COLLA
     R; KEEP VICTIM WARM",""        :rem 9
1650 DATA"{11 RIGHT}{RVS}EXPOSURE (TO COL
     D)"                            :rem 12
1660 DATA"{DOWN}{RVS}1){OFF} COVER FROZEN
      PART.","{RVS}2){OFF} PROVIDE EXTRA
     {SPACE}CLOTHING BLANKETS."     :rem 216
1670 DATA"{RVS}3){OFF} BRING INDOORS."
                                    :rem 234
1680 DATA"{RVS}4){OFF} IMMERSE FROZEN PAR
     T IN WARM WATER"               :rem 156
1681 DATA"{UP}{3 SPACES}(102-105F)OR WRAP
      GENTLY"                       :rem 90
1690 DATA"{UP}{3 RIGHT}IN WARM BLANKETS.
     {SPACE}DO NOT RUB."            :rem 1
1700 DATA"{RVS}5){OFF} DO NOT APPLY HEAT,
      HEAT LAMPS, OR HOT{3 SPACES}WATER B
     OTTLES."                       :rem 165
```

```
1710 DATA"{RVS}6){OFF} ELEVATE AFFECTED P
     ART.","                          :rem 55
1720 DATA"{11 RIGHT}{RVS}EXPOSURE(TO HEAT
     )"                               :rem 10
1730 DATA"{DOWN}{RVS}1){OFF} APPLY COLD W
     ATER OR RUBBING ALCOHOL{5 SPACES}TO
     {SPACE}BARE SKIN."               :rem 47
1740 DATA"{RVS}2){OFF} USE FANS OR AIR CO
     NDITIONING TO{9 SPACES}PROMOTE COOLI
     NG."                             :rem 66
1750 DATA"{RVS}3){OFF} CONTINUE UNTIL TEM
     PERATURE IS REDUCED"             :rem 2
1760 DATA"{RVS}4){OFF} GUARD AGAINST OVER
      CHILLING.",""                   :rem 203
1770 DATA"{15 RIGHT}{RVS}FAINTING":rem 66
1780 DATA"{DOWN}{RVS}1){OFF} PLACE IN LYI
     NG POSITION WITH HEAD{7 SPACES}LOWER
      THAN BODY."                    :rem 143
1790 DATA"{RVS}2){OFF} LOOSEN CLOTHING &
     {SHIFT-SPACE}APPLY COLD CLOTHS."
                                      :rem 191
1800 DATA"{RVS}3){OFF} USE SMELLING SALTS
      & UPON REVIVAL:{6 SPACES}COFFEE, FR
     ESH AIR.",""                     :rem 119
1810 DATA"{16 RIGHT}{RVS}FEVER"  :rem 130
1820 DATA"{DOWN}{RVS}1){OFF} INCREASE FLU
     ID INTAKE."                      :rem 224
1830 DATA"{RVS}2){OFF} DO NOT COVER EXCES
     SIVELY."                         :rem 174
1840 DATA"{RVS}3){OFF} GIVE ASPIRIN OR AC
     ETAMINOPHEN."                    :rem 247
1850 DATA"{RVS}4){OFF} COOLING SPONGES WI
     TH WATER ONLY."                  :rem 134
1860 DATA"{RVS}5){OFF} CALL YOUR DOCTOR."
     ,""                              :rem 3
1870 DATA"{15 RIGHT}{RVS}HEART ATTACK"
                                      :rem 31
1880 DATA"{RVS}1){OFF} CALL HOSPITAL (SPE
     CIFY NEED{13 SPACES}FOR OXYGEN)."
                                      :rem 28
1890 DATA"{RVS}2){OFF} KEEP VICTIM LYING
     {SPACE}DOWN IN A{13 SPACES}COMFORTAB
     LE POSITION."                    :rem 132
1900 DATA"{RVS}3){OFF} ADMINISTER ARTIFIC
     IAL RESPIRATION IF{4 SPACES}BREATHIN
     G HAS STOPPED."                  :rem 140
1910 DATA"{RVS}4){OFF} GIVE CPR IF THERE
     {SPACE}IS NO PULSE{11 SPACES}(IF TRA
     INED)."                          :rem 24
1920 DATA"{RVS}5){OFF} CHECK VICTIM FOR E
     MERGENCY MEDICAL{6 SPACES}INFORMATIO
     N &"                             :rem 55
1930 DATA"{2 UP}{17 RIGHT}PRESCRIPTION FO
     R"                               :rem 195
1940 DATA"{UP}{3 SPACES}MEDICINE. ADMINIS
     TER IF PATIENT IS{6 SPACES}CONSCIOUS
     .","                             :rem 173
1950 DATA"{8 RIGHT}{8 SPACES}{RVS}HERNIAS
     "                                :rem 49
1960 DATA"{DOWN}{RVS}1){OFF} LAY VICTIM O
     N BACK WITH KNEES{11 SPACES}WELL DRA
     WN UP."                          :rem 210
1970 DATA"{RVS}2){OFF} PLACE A CRAVAT BAN
     DAGE AROUND{11 SPACES}THIGHS AND TIE
     ."                               :rem 234
1980 DATA"{RVS}3){OFF} PUT BLANKET UNDER
     {SPACE}KNEES."                   :rem 76
1990 DATA"{RVS}4){OFF} DO NOT TRY TO FORC
     E PROTRUSION{10 SPACES}BACK INTO CAV
     ITY."                            :rem 53
2000 DATA"{RVS}5){OFF} COVER WITH BLANKET
      & TAKE TO HOSPITAL{3 SPACES}IN THIS
      POSITION.",""                   :rem 18
2010 DATA"{16 RIGHT}{RVS}NOSE BLEED"
                                      :rem 148
2020 DATA"{DOWN}{RVS}1){OFF} TIP HEAD FOR
     WARD, BLOW NOSE TO REMOVE{3 SPACES}A
     LL CLOTS."                       :rem 215
2030 DATA"{RVS}2){OFF} SQUEEZE NOSTRILS F
     IRMLY TOGETHER FOR{4 SPACES}10 MINUT
     ES.",""                          :rem 2
2040 DATA"{11 RIGHT}{RVS}POISONING BY GAS
     "                                :rem 161
2050 DATA"{DOWN}{RVS}1){OFF} CALL YOUR DO
     CTOR.","{RVS}2){OFF} OPEN OR BREAK W
     INDOWS."                         :rem 152
2060 DATA"{RVS}3){OFF} SHUT OFF GAS OR ST
     OP MOTOR."                       :rem 198
2070 DATA"{RVS}4){OFF} REMOVE VICTIM TO F
     RESH AIR."                       :rem 231
2080 DATA"{RVS}5){OFF} GIVE ARTIFICIAL RE
     SPIRATION."                      :rem 171
2090 DATA"{RVS}6){OFF} KEEP PATIENT WARM.
     ",""                             :rem 59
2100 DATA"{11 RIGHT}{RVS}POISONING BY MOU
     TH"                              :rem 80
2110 DATA"{DOWN}{RVS}1){OFF} DILUTE POISO
     N WITH WATER OR MILK.{7 SPACES}DISCO
     NTINUE IF"                       :rem 95
2120 DATA"{2 UP}{18 RIGHT}NAUSEATED."
                                      :rem 23
2130 DATA"{RVS}2){OFF} GET MEDICAL HELP &
      CALL POISON{10 SPACES}CONTROL CENTE
     R"                               :rem 23
2140 DATA"{2 UP}{18 RIGHT}IMMEDIATLY."
                                      :rem 110
2150 DATA"{RVS}3){OFF} DO NOT INDUCE VOMI
     TING UNLESS SO{8 SPACES}ADVISED BY P
     OISON"                           :rem 190
2160 DATA"{2 UP}{21 RIGHT}CONTROL CENTER.
     "                                :rem 186
2170 DATA"{RVS}4){OFF} DO NOT NEUTRALIZE.
      DO NOT GIVE OLIVE{4 SPACES}OIL OR O
     THER OILS."                      :rem 120
2180 DATA"{RVS}5){OFF} KEEP AIRWAY OPEN.
     {SPACE}ADMINISTER{12 SPACES}ARTIFICI
     AL RESPIRATION"                  :rem 149
2190 DATA"{2 UP}{26 RIGHT}IF NEEDED."
                                      :rem 160
2200 DATA"{RVS}6){OFF} TREAT FOR SHOCK."
                                      :rem 50
2210 DATA"{RVS}7){OFF} SAVE LABEL, CONTAI
     NER OR VOMITUS.",""              :rem 251
2220 DATA"{17 RIGHT}{RVS}SHOCK{DOWN}"
                                      :rem 172
2230 DATA"{RVS}1){OFF} KEEP VICTIM LYING
     {SPACE}DOWN."                    :rem 253
2240 DATA"{RVS}2){OFF} IF HEAD AND/OR CHE
     ST INJURY, RAISE{6 SPACES}HEAD & SHO
     ULDERS."                         :rem 128
2250 DATA"{RVS}3){OFF} IF IT WILL NOT AGG
     RAVATE ANY INJURY,{4 SPACES}RAISE FE
     ET."                             :rem 201
2260 DATA"{RVS}4){OFF} KEEP PATIENT WARM,
      PREVENT LOSS OF{6 SPACES}BODY HEAT.
     "                                :rem 62
2270 DATA"{RVS}5){OFF} TREAT FOR BREATH S
     TOPPAGE, BLEEDING,{4 SPACES}PAIN."
                                      :rem 147
2280 DATA"{RVS}6){OFF} GET PROMPT MEDICAL
```

```
     ASSISTANCE.","                    :rem 106
2290 DATA"{8 RIGHT}{7 SPACES}{RVS}SPRAINS
     "                                 :rem 69
2300 DATA"{DOWN}{RVS}1){OFF} ELEVATE INJU
     RED AREA; PLACE AT{10 SPACES}COMPLET
     E REST."                          :rem 92
2310 DATA"{RVS}2){OFF} APPLY ICE BAG, COL
     D CLOTH, OR{11 SPACES}CHEMICAL COLD
     {SPACE}PACK."                     :rem 126
2320 DATA"{RVS}3){OFF} IF SWELLING & PAIN
      CONTINUE SEE{9 SPACES}DOCTOR.","
                                       :rem 18
2330 DATA"{7 RIGHT}{7 SPACES}{RVS}STRAINS
     "                                 :rem 39
2340 DATA"{DOWN}{RVS}1){OFF} GET VICTIM C
     OMFORTABLE."                      :rem 62
2350 DATA"{RVS}2){OFF} APPLY HOT, WET TOW
     EL."                              :rem 109
2360 DATA"{RVS}3){OFF} KEEP INJURED AREA
     {SPACE}AT REST."                  :rem 121
2370 DATA"{RVS}4){OFF} GET MEDICAL ATTENT
     ION."                             :rem 222
2380 DATA""                            :rem 43
2382 DATA"{16 RIGHT}{RVS}STROKE"       :rem 231
2385 DATA"{RVS}1){OFF} COLD CLOTH ON FORE
     HEAD."                            :rem 211
2390 DATA"{RVS}2){OFF} TURN HEAD OF VOMIT
     ING PATIENT TO{8 SPACES}SIDE."
                                       :rem 147
2400 DATA"{RVS}3){OFF} GIVE NO STIMULANTS
      AND NOTHING BY{7 SPACES}MOUTH."
                                       :rem 64
2410 DATA"{RVS}4){OFF} CALL YOUR DOCTOR."
                                       :rem 138
2420 DATA""                            :rem 38
2430 DATA"{15 RIGHT}{RVS}SUNBURN","{RVS}
     {DOWN}1){OFF} APPLY BURN CREAM OR PE
     TROLEUM JELLY."                   :rem 201
2440 DATA"{RVS}2){OFF} CALL DOCTOR IF FEV
     ER, CHILLS, AND{7 SPACES}SICKNESS OC
     CUR."                             :rem 12
2450 DATA"{RVS}3){OFF} PROTECT REDNESS FR
     OM FURTHER SUN.","                :rem 70
2460 DATA"{15 RIGHT}{RVS}SUNSTROKE","
     {RVS}{DOWN}1){OFF} CALL YOUR DOCTOR.
     "                                 :rem 160
2470 DATA"{RVS}2){OFF} MOVE PATIENT TO A
     {SPACE}COOL, SHADY SPOT."         :rem 160
2480 DATA"{RVS}3){OFF} SPONGE ENTIRE BODY
      WITH COLD WATER."                :rem 252
2490 DATA"{RVS}4){OFF} DO NOT GIVE STIMUL
     ANTS.","                          :rem 143
2500 DATA"{13 RIGHT}{RVS}TOOTHACHE","
     {DOWN}{RVS}1){OFF} NO TREATMENT."
                                       :rem 77
2510 DATA"{RVS}2){OFF} APPLY COLD COMPRES
     SES UNTIL YOU GET{5 SPACES}TO THE DE
     NTIST.","                         :rem 119
2520 DATA"9"                           :rem 96
```

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.

Quiz Master

(Article on page 80.)

Program 1: Quiz Generator

```
10 REM QUIZ MASTER GENERATOR
20 DIMQ$(100),A$(100),B$(100),C$(100),D$(
   100),E$(100),M$(15)               :rem 48
30 PRINT"{WHT}":POKE53280,13:POKE53281,5:
   GOTO50                            :rem 217
40 POKE198,6:POKE631,30:POKE632,34:POKE63
   3,34:POKE634,20:POKE635,5:RETURN
                                     :rem 110
50 POKE53272,23:GOTO970              :rem 59
60 PRINT"{CLR}":CLR                  :rem 229
70 DIMQ$(100),A$(100),B$(100),C$(100),D$(
   100),E$(100),M$(15),SN$(400),G(400)
                                     :rem 99
80 GOSUB1830:GOSUB1360:GOSUB2070     :rem 187
90 PRINT"ENTER NUMBER OF TEST TO LOAD:":I
   NPUTN                             :rem 235
100 IFN<0ORN>XTHENPRINT"INVALID RANGE":GO
    TO90                             :rem 175
110 N$=M$(N):PRINTSPC(12)"{CLR}{RVS}
    {6 DOWN}{9 RIGHT}{3 SPACES}LOADING
    {SHIFT-SPACE}DATA{3 SHIFT-SPACE}"
                                     :rem 30
120 GOSUB1830:GOSUB2150:OPEN2,8,2,+N$+" F
    ILE,S,R":X=0                     :rem 21
130 X=X+1                            :rem 221
140 INPUT#2,Q$(X):INPUT#2,A$(X):INPUT#2,B
    $(X):INPUT#2,C$(X):INPUT#2,D$(X):INPU
    T#2,E$(X)                        :rem 139
150 IFST AND64THEN170                :rem 208
160 GOTO130                          :rem 100
170 CLOSE2:POKE198,0:L=X:N=X:T=0     :rem 218
180 IFR=1THEN240                     :rem 173
190 GOSUB2150:GOSUB2070:GOSUB1670:rem 237
200 IFH=0THEN970                     :rem 165
210 IFH=1THEN240                     :rem 157
220 REM INPUT QUESTIONS              :rem 211
230 PRINT"{CLR}{DOWN}ENTER NAME FOR QUIZ"
    :INPUTN$:GOSUB1460:N=0           :rem 15
240 N=N+1:PRINT"{CLR}":PRINTSPC(13)"{RVS}
     QUIZ MASTER {OFF}"              :rem 200
250 PRINT:PRINT"{RVS} WARNING!{2 SPACES}D
    O NOT EXCEED 80 CHARACTERS{2 SPACES}"
                                     :rem 114
260 PRINT:PRINT"{RVS}{7 SPACES}ENTER £ T
    O EXIT ROUTINE{10 SPACES}"       :rem 159
270 IFN>=100THENPRINT"{CLR}{5 DOWN}
    {14 SPACES}FILE{SHIFT-SPACE}FULL":FOR
    T=1TO2000:NEXT:GOTO980           :rem 50
280 GOSUB40                          :rem 126
290 H=0                              :rem 80
300 PRINT"ENTER QUESTION #";N:PRINT
                                     :rem 205
310 INPUTQ$(N)                       :rem 56
320 IFQ$(N)=""THEN310                :rem 124
330 IFQ$(N)=CHR$(92)THENN=N-1:GOTO770
                                     :rem 170
340 IFLEN(Q$(N))>80THENGOSUB1290     :rem 139
350 IFH=1THEN240                     :rem 162
360 PRINT"ENTER FIRST ANSWER:":PRINT:GOSU
    B40:H=0                          :rem 206
370 INPUT"A. ";A$(N):IFA$(N)=""THEN370
                                     :rem 51
380 IFASC(A$(N))=92THENN=N-1:GOTO770
                                     :rem 117
```

```
390 A$(N)="A. "+A$(N)                    :rem 191
400 IFLEN(A$(N))>80THENGOSUB1290         :rem 120
410 IFH=1THENGOTO360                     :rem 219
420 PRINT"ENTER SECOND ANSWER:":PRINT:GOS
    UB40:H=0                             :rem 255
430 INPUT"B. ";B$(N):IFB$(N)=""THEN430
                                         :rem 48
440 IFASC(B$(N))=92THENN=N-1:GOTO770
                                         :rem 115
450 B$(N)="B. "+B$(N)                    :rem 191
460 IFLEN(B$(N))>80THENGOSUB1290         :rem 127
470 IFH=1THENGOTO420                     :rem 222
480 PRINT"ENTER THIRD ANSWER:":PRINT:GOSU
    B40:H=0                              :rem 196
490 INPUT"C. ";C$(N):IFC$(N)=""THEN490
                                         :rem 63
500 IFASC(C$(N))=92THENN=N-1:GOTO770
                                         :rem 113
510 C$(N)="C. "+C$(N)                    :rem 191
520 IFLEN(C$(N))>80THENGOSUB1290         :rem 125
530 IFH=1THENGOTO480                     :rem 225
540 PRINT"ENTER FOURTH ANSWER:":PRINT:GOS
    UB40:H=0                             :rem 30
550 INPUT"D. ";D$(N):IFD$(N)=""THEN550
                                         :rem 60
560 IFASC(D$(N))=92THENN=N-1:GOTO770
                                         :rem 120
570 D$(N)="D. "+D$(N)                    :rem 200
580 IFLEN(D$(N))>80THENGOSUB1290         :rem 132
590 IFH=1THENGOTO540                     :rem 228
600 PRINT"ENTER LETTER OF CORRECT ANSWER:
    ":PRINT:H=0                          :rem 28
610 INPUTE$(N):IFE$(N)=""THEN610         :rem 71
620 IFASC(E$(N))=92THENN=N-1:GOSUB770:GOT
    O990                                 :rem 210
630 IFLEN(E$(N))<>1THENGOSUB1290         :rem 134
640 IFE$(N)="A"THEN690                   :rem 193
650 IFE$(N)="B"THEN690                   :rem 195
660 IFE$(N)="C"THEN690                   :rem 197
670 IFE$(N)="D"THEN690                   :rem 199
680 PRINT"{RVS} ERROR: RE-ENTER":GOTO610
                                         :rem 17
690 IFH=1THEN600                         :rem 169
700 IFP=1THENRETURN                      :rem 243
710 L=N:GOSUB2070:IFN=100THEN730         :rem 160
720 GOTO240                              :rem 104
730 PRINT:PRINTSPC(7)"FILE CONTAINS 100 E
    NTRIES."                             :rem 248
740 PRINT:PRINT"DATA WILL BE STORED. OPEN
     NEW TEXT FILE"                      :rem 140
750 GOSUB1720:GOSUB770:GOTO990           :rem 83
760 REM STORE DATA                       :rem 40
770 GOSUB2070:PRINTSPC(10)"{RVS} WAIT, ST
    ORING DATA ":GOSUB2150               :rem 63
780 GOSUB1830:OPEN2,8,2,"@0:"+N$+" FILE,S
    ,W"                                  :rem 147
790 FORX=1TOL:PRINT#2,Q$(X):PRINT#2,A$(X)
    :PRINT#2,B$(X):PRINT#2,C$(X):PRINT#2,
    D$(X):PRINT#2,E$(X):NEXT             :rem 211
800 CLOSE2:POKE198,0:GOSUB2150:GOSUB2070:
    GOSUB1670:RETURN                     :rem 78
810 REM CHANGE ANSWER                    :rem 243
820 GOSUB2070:P=1:PRINT"{CLR}{3 DOWN}
    {RVS} ENTER NUMBER OF QUESTION":INPUT
    W                                    :rem 179
830 PRINT"{CLR}{2 DOWN}":S$=Q$(W):GOSUB11
    90:S$=A$(W):GOSUB1190:S$=B$(W):GOSUB1
    190                                  :rem 178
840 S$=C$(W):GOSUB1190:S$=D$(W):GOSUB1190
                                         :rem 103
850 PRINT"CORRECT ANSWER IS:":PRINTE$(W)
                                         :rem 126
860 GOSUB1670:N=W-1:GOSUB240:GOSUB770:RET
    URN                                  :rem 40
870 REM REVIEW ROUTINE                   :rem 123
880 GOSUB2070:Y=1:PRINT"{CLR}{DOWN}"
                                         :rem 153
890 PRINT:PRINTTAB(20-LEN(N$)/2);N$:GOSUB
    1670:PRINT"{2 DOWN}"                 :rem 17
900 FORN=1TOL:PRINT"{CLR}"               :rem 201
910 IFQ$(N)=""THENGOTO960                :rem 197
920 S$=STR$(N)+". "+Q$(N):PRINT:GOSUB1190
                                         :rem 51
930 REM ANSWER CHOICES                   :rem 78
940 S$=A$(N):GOSUB1190:S$=B$(N):GOSUB1190
    :S$=C$(N):GOSUB1190:S$=D$(N):GOSUB119
    0                                    :rem 69
950 PRINT:PRINT"CORRECT ANSWER IS: ":PRIN
    TTAB(7)E$(N):GOSUB2070               :rem 31
960 GOSUB1670:NEXT:RETURN                :rem 128
970 REM PROGRAM MENU                     :rem 209
980 H=0                                  :rem 86
990 PRINT"{CLR}":POKE53280,13:POKE53281,5
                                         :rem 206
1000 GOSUB2070                           :rem 10
1010 P=0                                 :rem 127
1020 PRINTSPC(13)"{DOWN}{RVS} QUIZ
    {SHIFT-SPACE}MASTER "                :rem 7
1030 PRINT:PRINTSPC(5)"ENTER NUMBER OF FU
    NCTION:"                             :rem 196
1040 PRINT:PRINTSPC(8)"1. ENTER NEW QUEST
    IONS"                                :rem 222
1050 PRINT:PRINTSPC(8)"2. REVIEW QUESTION
    S"                                   :rem 202
1060 PRINT:PRINTSPC(8)"3. CHANGE A QUESTI
    ON"                                  :rem 142
1070 PRINT:PRINTSPC(8)"4. LOAD PREVIOUS D
    ATA"                                 :rem 104
1080 PRINT:PRINTSPC(8)"5. ADD TO TEST IN
    {SPACE}FILE"                         :rem 22
1090 PRINT:PRINTSPC(8)"6. INITIALIZE DISK
    "                                    :rem 210
1100 PRINT:PRINTSPC(8)"7. END"           :rem 133
1110 PRINT:PRINTSPC(5)"NUMBER?"          :rem 79
1120 GETG$:IFG$=""THEN1120               :rem 183
1130 G=ASC(G$)-48:IFG<1ORG>8THEN1120
                                         :rem 67
1140 ONGGOSUB230,880,820,60,1640,1730,117
    0                                    :rem 198
1150 GOTO990                             :rem 162
1160 GOSUB2070                           :rem 17
1170 POKE198,0:SYS198                    :rem 209
1180 REM PRINT JUSTIFY                   :rem 105
1190 PRINT                               :rem 88
1200 IFLEN(S$)<40THENPRINTS$:GOTO1280
                                         :rem 10
1210 X=40:Y=1                            :rem 190
1220 X=X-1                               :rem 16
1230 IFASC(MID$(S$,X,Y)+CHR$(0))<>32THEN1
    220                                  :rem 204
1240 PRINTLEFT$(S$,X)                    :rem 239
1250 Z=LEN(S$)                           :rem 6
1260 Z=Z-X                               :rem 63
1270 PRINTRIGHT$(S$,Z)                   :rem 71
1280 RETURN                              :rem 171
1290 PRINT"ENTRY TOO LONG: RE-PHRASE"
                                         :rem 18
1300 H=1:FORT=1TO2000:NEXT:RETURN        :rem 85
1310 REM TEST TITLE FILE                 :rem 139
1320 PRINT:PRINT"HAS TEST TITLE FILE BEEN
     INITIATED?(Y/N)":GOSUB2070          :rem 186
1330 GETG$:IFG$=""THEN1330               :rem 189
1340 IF G$="N"THEN1460                   :rem 139
1350 IFG$<>"Y"THEN1330                   :rem 208
```

```
1360 PRINT:PRINTSPC(17)"{RVS} WAIT "
                                    :rem 72
1370 GOSUB2150:GOSUB1830:OPEN3,8,3,"TEST
     {SPACE}TITLES,S,R"             :rem 125
1380 X=0                            :rem 145
1390 X=X+1:INPUT#3,M$(X)            :rem 124
1400 IFST AND64THEN1420             :rem 46
1410 GOTO1390                       :rem 204
1420 CLOSE3:POKE198,0:GOSUB2150     :rem 89
1430 IFS1<>0THEN2110                :rem 120
1440 PRINT"{CLR}":PRINTSPC(14)"TEST TITLE
     S":PRINT                       :rem 176
1450 FORA=1TOX:PRINTA;". ";M$(A):NEXT:RET
     URN                            :rem 225
1460 REM INITIATE TEST FILE         :rem 102
1470 IFX=15THENGOSUB1710            :rem 206
1480 IF X=15THENX=1                 :rem 76
1490 IFX=1THEN1600                  :rem 25
1500 PRINT"{CLR}HAS FILE OF TEST NAMES BE
     EN STARTED?"                   :rem 95
1510 GETG$:IFG$=""THEN1510          :rem 189
1520 IF G$="N"THEN1600              :rem 135
1530 IFG$="Y"THENGOSUB1370          :rem 23
1540 PRINTX+1". {RVS}"N$            :rem 1
1550 PRINT"IS YOUR TITLE ORIGINAL?"
                                    :rem 29
1560 GETG$:IFG$=""THEN1560          :rem 199
1570 IF G$="Y"THEN1600              :rem 151
1580 PRINT"ENTER NEW TITLE FOR TEST:"
                                    :rem 106
1590 INPUTN$                        :rem 209
1600 M$(X+1)=N$:GOSUB2070           :rem 111
1610 PRINT:PRINTSPC(13)"{RVS} SAVING TITL
     E "                            :rem 87
1620 GOSUB1830:OPEN3,8,3,"@0:TEST TITLES,
     S,W"                           :rem 168
1630 FORA=1TOX+1:PRINT#3,M$(A):NEXT:CLOSE
     3:POKE198,0:GOSUB2150:RETURN   :rem 98
1640 REM TEST ADDITION ROUTINE      :rem 97
1650 CLR                            :rem 173
1660 R=1:GOSUB70:R=0:GOTO990        :rem 188
1670 PRINT:PRINTTAB(5)"{RVS} PRESS
     {SHIFT-SPACE}SPACE{SHIFT-SPACE}BAR
     {SHIFT-SPACE}TO{SHIFT-SPACE}CONTINUE
      {OFF}"                        :rem 61
1680 GETG$:IFASC(G$+CHR$(0))<>32THEN1680
                                    :rem 0
1690 RETURN                         :rem 176
1700 PRINT"{CLR}"                   :rem 44
1710 PRINTSPC(10)"{CLR}{10 DOWN}FILE FULL
     "                              :rem 194
1720 FORT=1TO2000:NEXT:RETURN       :rem 107
1730 PRINT"{CLR}{2 DOWN}{RVS}{2 SPACES}DO
      YOU WANT TO INITIALIZE A NEW DISK?
     {SPACE}":PRINTTAB(17)"{RVS} (Y/N) "
                                    :rem 171
1740 GETG$:IFG$=""THEN1740          :rem 199
1750 IFG$="Y"THENPRINT"TYPE GOTO 1790 AND
      PRESS RETURN":END             :rem 62
1760 IFG$<>"N"THEN1740              :rem 207
1770 IFG$="N"THENRETURN             :rem 167
1780 END                            :rem 167
1790 PRINT"{CLR}{5 DOWN}{6 SPACES}INSERT
     {SPACE}NEW DISK INTO DRIVE     :rem 6
1800 PRINT"{4 DOWN}{2 SPACES}PRESS ANY KE
     Y WHEN READY TO PROCEED"       :rem 8
1810 GOSUB1820:GOTO1850             :rem 86
1820 GETG$:IFG$=""THEN1820          :rem 197
1830 OPEN15,8,15:PRINT#15,"I0:":CLOSE15:R
     ETURN                          :rem 253
1840 END                            :rem 164
1850 REM{2 SPACES}PRINT"{CLR}{5 DOWN}
     {17 RIGHT}{RVS} WAIT1730       :rem 72
1860 PRINT"{2 DOWN}{15 SPACES}{RVS} WARNI
     NG!!!"                         :rem 205
1870 PRINT"{2 SPACES}{RVS} DISK
     {SHIFT-SPACE}IN{SHIFT-SPACE}DRIVE
     {SHIFT-SPACE}IS{SHIFT-SPACE}ABOUT
     {SHIFT-SPACE}TO{SHIFT-SPACE}BE
     {SHIFT-SPACE}ERASED!"          :rem 101
1880 PRINT"{2 SPACES}{RVS}{SHIFT-SPACE}
     {7 SPACES}{SHIFT-SPACE}ARE
     {SHIFT-SPACE}YOU{SHIFT-SPACE}SURE? (
     Y/N){9 SPACES}"                :rem 174
1890 GETG$:IFG$=""THEN1890          :rem 211
1900 IFG$="Y"THEN1930               :rem 154
1910 IFG$="N"THEN990                :rem 101
1920 GOTO1890                       :rem 215
1930 PRINT"{CLR}{2 DOWN}ENTER DISKNAME";:
     INPUTDN$                       :rem 88
1940 IFLEN(DN$)>15THENPRINT"{2 DOWN}NAME
     {SHIFT-SPACE}TOO{SHIFT-SPACE}LONG, T
     RY{SHIFT-SPACE}AGAIN":FORT=1TO1000:N
     EXT:GOTO1930                   :rem 55
1950 PRINT"{2 DOWN}ENTER 2 CHARACTER DISK
      I.D.":INPUTID$                :rem 142
1960 POKE53281,2:POKE53280,2:PRINT"{CLR}
     {5 DOWN}{10 SPACES}LAST{SHIFT-SPACE}
     CHANCE{SHIFT-SPACE}TO{SHIFT-SPACE}ST
     OP!!!"                         :rem 30
1970 PRINT:PRINT"{9 SPACES}PRESS
     {SHIFT-SPACE}ANY{SHIFT-SPACE}KEY
     {SHIFT-SPACE}TO{SHIFT-SPACE}STOP!!!"
     :FORT=1TO1000                  :rem 90
1980 GETG$:IFG$<>""THEN990          :rem 224
1990 NEXT                           :rem 18
2000 PRINT"{CLR}{4 DOWN}DISK{SHIFT-SPACE}
     IS{SHIFT-SPACE}BEING{SHIFT-SPACE}FOR
     MATTED--WAIT"                  :rem 171
2010 OPEN15,8,15:PRINT#15,"N0:"+DN$+","+I
     D$                             :rem 33
2020 INPUT#15,S1,S$,S2,S3:CLOSE15:IFS1<>0
     THEN2110:GOSUB2050             :rem 20
2030 PRINT"{CLR}{10 DOWN}{9 SPACES}DISK
     {SHIFT-SPACE}FORMATTED{SHIFT-SPACE}"
     :FORT=1TO2000:NEXT             :rem 37
2040 POKE53280,13:POKE53281,5:GOTO990
                                    :rem 105
2050 FORT=1TO1000:NEXT:POKE53280,13:POKE5
     3281,5:RETURN                  :rem 47
2060 REM SOUND ROUTINE              :rem 91
2070 S=54272                        :rem 93
2080 POKES,100:POKES+1,125:POKES+5,0:POKE
     S+6,240:POKES+24,15:POKES+4,17
                                    :rem 202
2090 FORT=0TO100:NEXT               :rem 32
2100 POKES+4,0:RETURN               :rem 26
2110 PRINT"DISK{SHIFT-SPACE}ERROR ";S1,S$
     ,S2,S3                         :rem 47
2120 PRINT:PRINT"CORRECT ERROR CONDITION
     {SPACE}AND TRY AGAIN"          :rem 210
2130 GOSUB1670                      :rem 20
2140 GOTO980                        :rem 161
2150 OPEN15,8,15:INPUT#15,S1,S$,S2,S3:CLO
     SE15:IFS1<>0THEN2110           :rem 95
2160 PRINT"DISK STATUS: "S$         :rem 90
2170 RETURN                         :rem 170
```

Program 2: Student Quiz

```
10 REM STUDENT QUIZ
20 PRINT"{CLR}{WHT}":CLR:POKE53280,16:POK
   E53281,16:POKE808,225:POKE649,0:S=5472
   7                                :rem 236
```

```
30 DIMQ$(100),A$(100),B$(100),C$(100),D$(
   100),E$(100),M$(15),A(100)     :rem 128
40 GOSUB840:PRINT"{CLR}{N}":PRINTSPC(12)"
   {RVS}{2 SPACES}LOADING{SHIFT-SPACE}DAT
   A{3 SHIFT-SPACE}":PRINT"{BLK}":GOSUB39
   0:PRINT"{WHT}"                 :rem 252
50 FORX=1TOA:PRINTX". "M$(X):NEXT :rem 26
60 PRINT"{DOWN}ENTER NUMBER OF TEST":POKE
   649,10:INPUTX                  :rem 159
70 IFX<1ORX>ATHENPRINT"INVALID RANGE":GOT
   O60                            :rem 128
80 N$=M$(X):POKE649,0:OPEN15,8,15:PRINT"
   {CLR}":OPEN2,8,2,+N$+" FILE,S,R"
                                  :rem 180
90 PRINT"{9 DOWN}{5 SPACES}LOADING
   {SHIFT-SPACE}";N$;" QUIZ":PRINT"{BLK}"
                                  :rem 1
100 X=0                           :rem 86
110 X=X+1                         :rem 219
120 INPUT#2,Q$(X):INPUT#2,A$(X):INPUT#2,B
    $(X):INPUT#2,C$(X):INPUT#2,D$(X):INPU
    T#2,E$(X)                     :rem 137
130 IFST AND64THEN150             :rem 204
140 GOTO110                       :rem 96
150 CLOSE2:POKE198,0:L=X:CLOSE15:GOSUB920
    :PRINT"{CLR}{WHT}"            :rem 206
160 REM TEST ROUTINE              :rem 225
170 Y=1:POKE649,10:GOSUB540       :rem 75
180 FORN=1TOL-1:PRINT"{CLR}{DOWN}":PRINTT
    AB(20-LEN(N$)/2);N$           :rem 102
190 S$=STR$(N)+". "+Q$(A(N)):PRINT:GOSUB4
    50                            :rem 146
200 REM ANSWER CHOICES            :rem 68
210 S$=A$(A(N)):GOSUB450:S$=B$(A(N)):GOSU
    B450:S$=C$(A(N)):GOSUB450     :rem 225
220 S$=D$(A(N)):GOSUB450:S$=E$(A(N))
                                  :rem 188
230 PRINT"{DOWN}ENTER LETTER OF MOST CORR
    ECT ANSWER:":POKE198,0        :rem 160
240 INPUTF$                       :rem 144
250 IFLEN(F$)<>1THENPRINT"ENTER ONE LETTE
    R ONLY":GOTO240               :rem 102
260 IFASC(F$)<65ORASC(F$)>68THENPRINT"ANS
    WER MUST BE A,B,C, OR D":GOTO240
                                  :rem 151
270 IFASC(F$)=ASC(S$)THENP=P+1    :rem 254
280 IFASC(F$)=ASC(S$)THENPRINTSPC(9)"
    {RVS}{2 SPACES}ANSWER IS CORRECT!! ":
    GOSUB1030                     :rem 215
290 IFASC(F$)<>ASC(S$)THEN:GOSUB1060:GOSU
    B820                          :rem 217
300 FORT=1TO4000:NEXT:NEXT        :rem 149
310 N=N-1                         :rem 203
320 S=INT(P/N*100+.5):PRINT"{CLR}{DOWN}YO
    U SCORED ";S;" %"             :rem 149
330 IFS>80ANDS<90THENPRINT"STUDY THIS SEC
    TION AGAIN"                   :rem 175
340 IFS>=90ANDS<100THENPRINT"VERY GOOD, B
    UT MORE STUDY WOULD HELP"     :rem 214
350 IFS=100THENPRINT"EXCELLENT!!
    {2 SPACES}PERFECT SCORE!!"    :rem 245
360 FORT=1TO3000:NEXT             :rem 33
370 PRINT"{4 DOWN}ENTER RUN TO RE-START P
    ROGRAM":POKE808,237:END       :rem 17
380 REM PRINT JUSTIFY             :rem 58
390 OPEN15,8,15:OPEN3,8,3,"TEST TITLES,S,
    R":PRINT"{BLK}"               :rem 169
400 X=X+1                         :rem 221
410 INPUT#3,M$(X)                 :rem 193
420 IFSTATUSAND64THEN440          :rem 13
430 GOTO400                       :rem 100
440 CLOSE3:POKE198,0:A=X:CLOSE15:PRINT"
    {CLR}{WHT}":RETURN            :rem 139
450 IFLEN(S$)<40THENPRINTS$:GOTO510
                                  :rem 171
460 X=40:Y=1                      :rem 148
470 X=X-1                         :rem 230
480 IFASC(MID$(S$,X,Y)+CHR$(0))<>32THEN47
    0                             :rem 120
490 PRINTLEFT$(S$,X)              :rem 197
500 Z=LEN(S$):Z=Z-X:PRINTRIGHT$(S$,Z)
                                  :rem 58
510 RETURN                        :rem 118
520 PRINT:PRINTSPC(14)"TEST TITLES":PRINT
    :FORA=1TOX:PRINTA;". ";M$(A):NEXT:RET
    URN                           :rem 248
530 REM DISABLE CURSOR CONTROLS   :rem 194
540 IFPEEK(830)=133THEN560        :rem 215
550 FORI=828TO977:READA:POKEI,A:NEXT
                                  :rem 34
560 SYS828:RETURN                 :rem 86
570 DATA169,000,133,252,169,080   :rem 42
580 DATA133,251,169,164,133,002   :rem 38
590 DATA169,083,141,036,003,169   :rem 49
600 DATA003,141,037,003,096,152   :rem 25
610 DATA072,138,072,165,252,208   :rem 42
620 DATA007,032,116,003,169,000   :rem 21
630 DATA133,253,166,253,189,000   :rem 41
640 DATA002,133,254,198,252,230   :rem 36
650 DATA253,104,170,104,168,165   :rem 40
660 DATA254,096,160,000,132,252   :rem 34
670 DATA165,002,032,210,255,169   :rem 37
680 DATA157,032,210,255,032,228   :rem 38
690 DATA255,240,251,164,252,133   :rem 42
700 DATA254,169,032,032,210,255   :rem 33
710 DATA169,157,032,210,255,165   :rem 43
720 DATA254,201,013,240,043,201   :rem 17
730 DATA020,208,013,192,000,240   :rem 18
740 DATA211,136,169,157,032,210   :rem 36
750 DATA255,076,118,003,041,127   :rem 39
760 DATA201,032,144,196,196,251   :rem 44
770 DATA240,192,165,254,153,000   :rem 38
780 DATA002,032,210,255,169,000   :rem 27
790 DATA133,212,200,076,118,003   :rem 30
800 DATA230,252,153,000,002,169   :rem 23
810 DATA032,032,210,255,096,013   :rem 27
820 PRINTSPC(10)"{RVS} SORRY ANSWER IS WR
    ONG "                         :rem 45
830 PRINT"{DOWN}CORRECT CHOICE IS: ";S$:R
    ETURN                         :rem 92
840 PRINT"{CLR}{5 DOWN}":PRINTSPC(13)"
    {RVS} QUIZ MASTER ":POKE53272,23
                                  :rem 31
850 PRINT"{DOWN}{4 SPACES}THESE TESTS ARE
     MULTIPLE CHOICE."            :rem 40
860 PRINT"ENTER THE BEST ANSWER FROM THE
    {SPACE}CHOICES"               :rem 95
870 PRINT"GIVEN."                 :rem 23
880 PRINT"{DOWN}{4 SPACES}ENTER THE NUMBE
    R OF THE TEST YOU "           :rem 221
890 PRINT"HAVE BEEN ASSIGNED WHEN THE PRO
    GRAM "                        :rem 41
900 PRINT"CALLS FOR IT."          :rem 139
910 FORT=1TO6000:NEXT:RETURN      :rem 63
920 REM RANDOM GEN.               :rem 72
930 PRINT"{CLR}{DOWN}WAIT-- PREPARING QUI
    Z":PRINT"{BLK}"               :rem 44
940 FORX=1TOL                     :rem 57
950 A(X)=INT(RND(.)*L)+1          :rem 54
960 IFX=1THEN1000                 :rem 228
970 FORY=1TOX-1                   :rem 167
980 IFA(Y)=A(X)THEN950            :rem 15
990 NEXTY                         :rem 58
1000 NEXTX                        :rem 88
```

```
1010 PRINT"{WHT}":RETURN          :rem 178
1020 REM CORRECT ANSWER SOUND     :rem 18
1030 S=54272:POKES,150:POKES+1,100:POKES+
     5,0:POKES+6,240:POKES+24,15:POKES+4,
     17                           :rem 144
1040 FORT=0TO200:NEXT:POKES+4,0:RETURN
                                  :rem 172
1050 REM WRONG ANSWER SOUND       :rem 144
1060 S=54272:POKES,150:POKES+1,5:POKES+5,
     0:POKES+6,240:POKES+24,15:POKES+4,17
                                  :rem 55
1070 FORT=0TO200:NEXT:POKES+4,0:RETURN
                                  :rem 175
```

Turtle Graphics Interpreter

(Article on page 90.)

BEFORE TYPING...

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

Program 1: The Interpreter

```
10 REM TURTLE GRAPHICS INTERPRETER
                                  :rem 202
30 IF PEEK(49152)<>173 THEN PRINT CHR$(15
   0) "TURTLE DATA DID NOT LOAD": END
                                  :rem 87
40 X=0: Y=0: IX=0: IY=0: D=0: NU=0: BY=0:
    BI=0: XH=160: XL=-159: C=↑/180
                                  :rem 121
50 CR=.74: YH=INT(79/CR): YL=-YH: BA=2: B
   B=8: BL=320: SC=8192: PE=0: DR=-1
                                  :rem 195
60 MA=7: H=0: PX=53248: BB=8: BL=320: SC=
   8192: PE=0: DR=-1: MA=7: H=0: PX=53248
                                  :rem 33
70 PY=53249: BG=256: RO=0: CO=0: XS=0: YS
   =0: SP=0: PT=2040: SE=53269: HA=.5
                                  :rem 189
80 C1=12: C2=40: C3=50: C4=28: C5=24: C6=
   3: C7=5: CI=360: MX=53264: PC=0:rem 10
90 FF=255: SS=45: SB=56: YM=79    :rem 88
100 DIM ST$(255),ST(255),RP(255),PR$(255)
    ,PN$(255)                     :rem 88
110 DEF FNR(X)=INT((X+.005)*100)/100
                                  :rem 123
120 REM INITIALIZE SCREEN AND TURTLE
                                  :rem 220
130 GOSUB 3000: POKE 2, 110: POKE 53277,
    {SPACE}0: POKE 53271, 0: POKE 53287,0
                                  :rem 146
140 SYS 49295: SYS 49235: SYS 49322: POKE
     SE, 1: POKE 53280,2: POKE53281,11
                                  :rem 63
150 PRINT CHR$(129) "TURTLE GRAPHICS INTE
    RPRETER"                      :rem 218
170 PRINT CHR$(30)                :rem 218
200 REM MAIN LOOP - GET A LINE OF COMMAND
    S AND PROCESS IT              :rem 193
210 ST$(0)="": INPUT ST$(0)       :rem 118
220 NE=0: ST(0)=0: RP(0)=0: ER=0  :rem 107
230 IF ST$(0)="" THEN 210         :rem 179
240 REM COPY UNEXECUTED PART OF CURRENT C
    OMMAND STRING (NESTING LEVEL = NE)
                                  :rem 37
250 REM INTO IN$ TO BE PROCESSED  :rem 66
260 IN$=RIGHT$(ST$(NE), LEN(ST$(NE))-ST(N
    E)): IN=0                     :rem 51
270 GOSUB 5000{2 SPACES}FILL WD$ WITH NEX
    T WORD FROM IN$               :rem 106
280 IF WD$<>"" THEN 350           :rem 109
290 REM IN$ IS EMPTY; WE ARE DONE WITH AL
    L COMMANDS IF NESTING LEVEL IS 0
                                  :rem 140
300 IF NE=0 THEN 200              :rem 227
310 REM WE HAVE COMPLETED A REPETITION OF
     THE CURRENT COMMAND STRING ST$(NE)
                                  :rem 55
320 REM IF NEEDED, REPEAT.{2 SPACES}ELSE,
     POP NESTING LEVEL            :rem 156
330 RP(NE)=RP(NE)-1: IF RP(NE)>0 THEN ST(
    NE)=0: GOTO 240               :rem 42
340 NE=NE-1: GOTO 240             :rem 97
350 IF (WD$="REPEAT")OR(WD$="RP") THEN 44
    0                             :rem 20
360 REM CHECK IF COMMAND IS A PROCEDURE N
    AME                           :rem 16
370 GOSUB 6000: IF PN=0 THEN 410 :rem 120
380 REM STUFF IN$ WITH PROC STRING AS IF
    {SPACE}IT WERE A REPEAT LOOP  :rem 56
390 IN$= "[" + PR$(PN) + "]" + RIGHT$(IN$
    , LEN(IN$)-IN): IN=0: NU=1    :rem 28
400 ST(NE)=ST(NE)-LEN(PR$(PN))-2: GOTO 48
    0                             :rem 103
410 REM IDENTIFY AND EXECUTE WD$ AS A COM
    MAND                          :rem 78
420 GOSUB 1000: IF ER THEN 200    :rem 248
430 GOTO 270: REM WE ARE DONE CURRENT COM
    MAND                          :rem 67
440 REM GET REPETITION FACTOR FOR REPEAT
    {SPACE}LOOP                   :rem 0
450 GOSUB 4000: IN$=RIGHT$(IN$, LEN(IN$)-
    IN): IN=0                     :rem 214
460 IF (NOT ER)AND(NU>0)AND(INT(NU)=NU) T
    HEN 480                       :rem 229
470 PRINT "I CAN'T REPEAT SOMETHING " WD$
     " TIMES" :IN$="": GOTO 200   :rem 113
480 REM PUSH THE COMMAND STRING STACK (IN
    CREMENT NESTING LEVEL)        :rem 115
490 NE=NE+1: IF NE=256 THEN PRINT "NESTIN
    G TOO DEEP": GOTO 200         :rem 191
495 RP(NE)=NU: ST(NE)=1: K=0      :rem 45
500 REM FILL ST$(NE) WITH CONTENTS OF REP
    EAT BRACKETS                  :rem 158
510 ST$(NE)="": QQ=0: K=0         :rem 1
520 T$=MID$(IN$, ST(NE), 1)       :rem 106
530 IF T$="]" THEN K=K-1          :rem 221
540 IF K>0 THEN ST$(NE)=ST$(NE)+T$:rem 78
550 IF T$="[" THEN K=K+1: QQ=-1   :rem 82
560 IF K<=0 THEN 600              :rem 227
570 ST(NE)=ST(NE)+1               :rem 75
580 IF ST(NE)<=LEN(IN$) THEN 520 :rem 225
590 PRINT "MISMATCHED BRACKETS IN REPEAT"
    : IN$="": GOTO 200            :rem 112
600 IF (K<0) OR ((K=0)AND(NOTQQ)) THEN 59
    0                             :rem 172
610 ST(NE-1)=ST(NE)+ST(NE-1): ST(NE)=0
                                  :rem 142
620 GOTO 240: REM EXECUTE THE NEW COMMAND
     STRING                       :rem 57
1000 REM IDENTIFY AND EXECUTE COMMAND
                                  :rem 230
```

```
1005 ER=0                              :rem 202
1010 IF (WD$="FORWARD")OR(WD$="FD") THEN
     {SPACE}GOSUB 9000: RETURN         :rem 69
1020 IF (WD$="RIGHT")OR(WD$="RT") THEN GO
     SUB 10000: RETURN                 :rem 243
1030 IF (WD$="LEFT")OR(WD$="LT") THEN GOS
     UB 11000: RETURN                  :rem 156
1040 IF (WD$="PENUP")OR(WD$="PU") THEN PE
     =-1: RETURN                       :rem 189
1050 IF (WD$="PENDOWN")OR(WD$="PD") THEN
     {SPACE}PE=0: RETURN               :rem 18
1060 IF WD$="HOME" THEN GOSUB 12000: RETU
     RN                                :rem 123
1070 IF WD$="CLEAN" THEN SYS 49295: RETUR
     N                                 :rem 79
1080 IF (WD$="CLEARSCREEN")OR(WD$="CS") T
     HEN GOSUB 12000: SYS 49295: RETURN
                                       :rem 218
1090 IF (WD$="SETHEADING")OR(WD$="SETH")
     {SPACE}THEN GOSUB 13000: RETURN
                                       :rem 233
1100 IF (WD$="SETPOSITION")OR(WD$="SETP")
      THEN GOSUB 14000: RETURN         :rem 111
1110 IF (WD$="PENERASE")OR(WD$="PE") THEN
      DR=0: RETURN                     :rem 73
1120 IF (WD$="PENDRAW")OR(WD$="PW") THEN
     {SPACE}DR=-1: RETURN              :rem 72
1130 IF (WD$="ST")OR(WD$="SHOWTURTLE") TH
     EN POKE SE, 1: RETURN             :rem 76
1140 IF (WD$="HIDETURTLE")OR(WD$="HT") TH
     EN POKE SE, 0: RETURN             :rem 26
1150 IF (WD$="PENCOLOR")OR(WD$="PC") THEN
      GOSUB 15000: RETURN              :rem 205
1160 IF (WD$="BACKGROUNDCOLOR")OR(WD$="BC
     ") THEN GOSUB 16000: RETURN       :rem 190
1170 IF (WD$="TURTLECOLOR")OR(WD$="TC") T
     HEN GOSUB 17000: RETURN           :rem 210
1180 IF WD$="PRINTHEADING" THEN PRINT FNR
     (H): RETURN                       :rem 107
1190 IF WD$="PRINTPOSITION" THEN PRINT "(
     " FNR(X) "," FNR(Y) ")": RETURN
                                       :rem 218
1200 IF WD$="DEFINE" THEN GOSUB 18000: RE
     TURN                              :rem 255
1210 IF WD$="NAMES" THEN GOSUB 19000: RET
     URN                               :rem 202
1220 IF (WD$="PRINTPROCEDURE")OR(WD$="PPR
     OC") THEN GOSUB 20000: RETURN
                                       :rem 140
1230 IF WD$="ERASE" THEN GOSUB 21000: RET
     URN                               :rem 193
1240 IF WD$="ERASEALL" THEN PC=0: PRINT "
     ALL PROCEDURES ERASED": RETURN
                                       :rem 188
1250 IF WD$="RENAME" THEN GOSUB 22000: RE
     TURN                              :rem 12
1260 IF WD$="LOAD" THEN GOSUB 23000: RETU
     RN                                :rem 118
1270 IF WD$="SAVE" THEN GOSUB 24000: RETU
     RN                                :rem 135
1280 IF WD$="SCRATCH" THEN GOSUB 25000: R
     ETURN                             :rem 98
1290 IF WD$="QUIT" THEN PRINT "BYE": END
                                       :rem 207
1300 ER=-1: PRINT "I DON'T UNDERSTAND " W
     D$: RETURN                        :rem 119
2000 REM MOVE TURTLE                   :rem 189
2010 RO=YM-(Y*CR): CO=X-XL             :rem 15
2020 IF (SP/BA)=INT(SP/BA) THEN XS=CO+C1:
      YS=RO+C2: GOTO 2200              :rem 170
2030 XS=CO: IF SP>C6 THEN XS=XS+C5
                                       :rem 199
2050 IF (SP=C6)OR(SP=C7) THEN YS=RO+C4: G
     OTO 2200                          :rem 222
2060 YS=RO+C3                          :rem 243
2200 IF XS<BG THEN POKE PX, XS: POKE MX,
     {SPACE}0: GOTO 2220               :rem 67
2210 POKE PX, XS-BG: POKE MX, 1        :rem 148
2220 POKE PY, YS                       :rem 118
2230 RETURN                            :rem 167
3000 REM CHANGE HEADING                :rem 61
3010 H=H+DH                            :rem 72
3020 IF H>=CI THEN H=H-CI: GOTO 3020
                                       :rem 144
3030 IF H<0 THEN H=H+CI: GOTO 3030
                                       :rem 245
3040 SP=(INT(H/SS+HA)) AND MA:         :rem 160
3050 QQ=PEEK(SE): POKE SE, 0: POKE PT, SB
     +SP: GOSUB 2000                   :rem 42
3065 POKE SE, QQ                       :rem 99
3070 RETURN                            :rem 170
4000 REM NUMERIC INPUT                 :rem 75
4010 REM GETS NEXT WORD FROM IN$ AS A NUM
     BER (NU).{2 SPACES}CHECKS FOR ERROR
                                       :rem 40
4020 GOSUB 5000: ER=0: NU=0: IF WD$="" TH
     EN ER=-1: RETURN                  :rem 23
4030 FOR K= 1 TO LEN(WD$): T$=MID$(WD$, K
     , 1)                              :rem 202
4040 IF ((T$<"0")OR(T$>"9")) AND (T$<>"-"
     )AND(T$<>"+")AND(T$<>".") THEN ER=-1
                                       :rem 59
4050 NEXT: NU=VAL(WD$): RETURN         :rem 47
5000 REM FILL WD$ WITH NEXT WORD FROM IN$
                                       :rem 53
5010 WD$="": IF IN$="" THEN 5070       :rem 6
5020 IN$=RIGHT$(IN$, LEN(IN$)-IN): IN=0
                                       :rem 134
5030 ST(NE)=ST(NE)+1: IN=IN+1          :rem 120
5040 IF IN>LEN(IN$) THEN IN=IN-1: ST(NE)=
     ST(NE)-1: GOTO 5070               :rem 58
5050 IF MID$(IN$, IN, 1)<>" " THEN WD$=WD
     $ + MID$(IN$, IN, 1): GOTO 5030
                                       :rem 187
5060 IF (WD$="")AND(IN$<>"") THEN 5020
                                       :rem 126
5070 RETURN                            :rem 172
6000 REM IDENTIFY PROCEDURE            :rem 175
6010 REM RETURNS INDEX (PN) OF PROCNAME I
     N WD$; 0 IF NOT A PROCNAME        :rem 6
6020 K=0: PN=0                         :rem 197
6030 K=K+1: IF K>PC THEN RETURN        :rem 236
6040 IF WD$<>PN$(K) THEN 6030          :rem 232
6050 PN=K: RETURN                      :rem 11
7000 REM OPEN DISK FILE                :rem 40
7010 ER=0: GOSUB 5000: IF WD$<>"" THEN 70
     30                                :rem 138
7020 ER=-1: PRINT "YOU MUST SUPPLY A FILE
     NAME": RETURN                     :rem 213
7030 OPEN 15,8,15                      :rem 88
7040 OPEN 2,8,2, "0:" + WD$ + ".TURTLE,S,
     " + MD$: INPUT#15, QQ,T$,K,ZZ
                                       :rem 217
7050 IF (QQ=26)AND(MD$="W") THEN PRINT "W
     RITE-PROTECTED DISK": ER=-1: RETURN
                                       :rem 183
7060 IF (QQ=67)AND(MD$="W")AND(K=36) THEN
      PRINT "DISK IS FULL.": ER=-1: RETUR
     N                                 :rem 109
7070 IF (QQ=63)AND(MD$="W") THEN PRINT "F
```

```
     ILENAME IS USED": ER=-1: RETURN
                                        :rem 59
7080 IF (QQ=62)AND(MD$="R") THEN PRINT "N
     O SUCH FILE ON DISK": ER=-1: RETURN
                                        :rem 224
7090 IF QQ>19 THEN PRINT "I'M HAVING TROU
     BLE WITH THE DISK": ER=-1    :rem 244
7100 RETURN                       :rem 168
8000 REM GET VALID COLOR NUMBER   :rem 68
8010 GOSUB 4000 NUMERIC INPUT     :rem 176
8020 IF ER OR (NU>15)OR(NU<0) THEN ER=-1
                                  :rem 139
8030 RETURN                       :rem 171
9000 REM FORWARD COMMAND          :rem 193
9010 GOSUB 4000: IF ER OR (NU<=0) THEN PR
     INT "I CAN'T GO FORWARD " WD$: RETUR
     N                            :rem 198
9020 IX=X: IY=Y: FOR D= 0 TO NU: X=FNR(D*
     SIN(H*C)+IX): Y=FNR(D*COS(H*C)+IY)
                                  :rem 232
9030 IF X>XH THEN X=XH            :rem 245
9040 IF X<XL THEN X=XL            :rem 252
9050 IF Y>YH THEN Y=YH            :rem 251
9060 IF Y<YL THEN Y=YL             :rem 2
9070 IF PE THEN 9120              :rem 239
9080 BY=SC + BL*INT((YM-(Y*CR))/BB) +BB*I
     NT((X-XL)/BB) + ((YM-(Y*CR)) AND MA)
                                   :rem 74
9090 BI=MA - ((X-XL) AND MA)      :rem 129
9100 IF DR THEN POKE BY, PEEK(BY) OR BA↑B
     I: GOTO 9120                 :rem 113
9110 POKE BY, PEEK(BY) AND (FF-BA↑BI)
                                   :rem 27
9120 GOSUB 2000: NEXT: RETURN     :rem 161
10000 REM RIGHT COMMAND            :rem 82
10010 GOSUB 4000: IF ER OR (NU<0) THEN PR
      INT "I CAN'T TURN RIGHT " WD$: RETU
      RN                           :rem 205
10020 DH=NU: GOSUB 3000: RETURN    :rem 246
11000 REM LEFT COMMAND              :rem 0
11010 GOSUB 4000: IF ER OR (NU<0) THEN PR
      INT "I CAN'T GO LEFT " WD$: RETURN
                                   :rem 200
11020 DH=-NU: GOSUB 3000: RETURN    :rem 36
12000 REM HOME COMMAND             :rem 255
12010 X=0: Y=0: H=0: DH=0: GOSUB 3000: RE
      TURN                         :rem 114
13000 REM SETHEADING COMMAND       :rem 179
13010 GOSUB 4000: IF (NOT ER)AND(H<=360)
      {SPACE}THEN 13030            :rem 127
13020 ER=-1: PRINT "I CAN'T SET A HEADING
       OF " WD$: RETURN            :rem 84
13030 H=NU: DH=0: GOSUB 3000: RETURN
                                   :rem 233
14000 REM SETPOSITION COMMAND       :rem 57
14010 GOSUB 4000: IF (NOT ER)AND(NU>=XL)A
      ND(NU<=XH) THEN 14030        :rem 201
14020 ER=-1: PRINT "I CAN'T SET AN X-VALU
      E OF "WD$: RETURN            :rem 181
14030 QQ=NU: GOSUB 4000            :rem 248
14040 IF (NOT ER)AND(NU>=YL)AND(NU<=YH) T
      HEN X=QQ: Y=NU: GOSUB 2000: RETURN
                                   :rem 152
14050 ER=-1: PRINT "I CAN'T SET A Y-VALUE
       OF "WD$: RETURN             :rem 107
15000 REM PENCOLOR COMMAND         :rem 59
15010 GOSUB 8000: IF ER THEN PRINT WD$ "
      {SPACE}IS NOT A PENCOLOR": RETURN
                                   :rem 168
15020 POKE 2, (PEEK(2)AND15)+16*NU: SYS 4
      9235: RETURN                 :rem 112
16000 REM BACKGROUNDCOLOR COMMAND :rem 57
16010 GOSUB 8000: IF ER THEN PRINT WD$ "
      {SPACE}IS NOT A BACKGROUNDCOLOR": R
      ETURN                        :rem 166
16020 POKE 2, (PEEK(2)AND240)+NU: SYS 492
      35: RETURN                   :rem 16
17000 REM TURTLECOLOR COMMAND      :rem 58
17020 GOSUB 8000: IF ER THEN PRINT WD$ "
      {SPACE}IS NOT A TURLTECOLOR": RETUR
      N                            :rem 168
17030 POKE 53287, NU: RETURN       :rem 28
18000 REM DEFINE NEW PROCEDURE     :rem 27
18010 GOSUB 5000:IF WD$<>"" THEN 18030
                                   :rem 176
18020 PRINT "I NEED A PROCEDURE NAME": ER
      =-1: RETURN                  :rem 194
18030 IF PC=FF THEN PRINT"I CAN'T REMEMBE
      R ANY MORE PROCEDURES": ER=-1: RETU
      RN                           :rem 105
18040 GOSUB 6000: IF PN<>0 THEN PRINT WD$
       " ALREADY EXISTS": ER=-1: RETURN
                                   :rem 123
18050 PC=PC+1: PN$(PC)=WD$: PRINT WD$;: I
      NPUT PR$(PC)                 :rem 206
18060 PRINT WD$ " IS NOW DEFINED": RETURN
                                   :rem 40
19000 REM PRINTNAMES COMMAND       :rem 222
19010 PRINT "NUMBER OF PROCEDURES:" PC
                                   :rem 243
19020 IF PC=0 THEN RETURN          :rem 154
19030 FOR K= 1 TO PC: PRINT PN$(K): NEXT:
       RETURN                      :rem 139
20000 REM PRINTPROCEDURE COMMAND   :rem 11
20010 GOSUB 5000: IF WD$<>"" THEN 20030
                                   :rem 162
20020 ER=-1: PRINT "I NEED A PROCEDURE NA
      ME": RETURN                  :rem 187
20030 GOSUB 6000: IF PN<>0 THEN PRINT PR$
      (PN): RETURN                 :rem 215
20040 ER=-1: PRINT "THERE IS NO PROCEDURE
       " WD$: RETURN               :rem 102
21000 REM ERASE COMMAND            :rem 70
21010 GOSUB 5000: IF WD$<>"" THEN 21030
                                   :rem 164
21020 ER=-1: PRINT "I NEED A PROCEDURE NA
      ME": RETURN                  :rem 188
21030 GOSUB6000: IF PN<>0 THEN 21050
                                   :rem 116
21040 ER=-1: PRINT "THERE IS NO PROCEDURE
       " WD$: RETURN               :rem 103
21050 PR$(PN)=PR$(PC): PN$(PN)=PN$(PC): P
      C=PC-1:PRINT WD$ " IS ERASED": RETU
      RN                           :rem 145
22000 REM RENAME COMMAND           :rem 143
22010 GOSUB 5000: IF WD$<>"" THEN 22030
                                   :rem 166
22020 ER=-1: PRINT "I NEED TO KNOW THE OL
      D NAME": RETURN              :rem 117
22030 GOSUB 6000                    :rem 61
22040 IF PN=0 THEN PRINT "PROCEDURE " WD$
       " DOESN'T EXIST": ER=-1: RETURN
                                   :rem 69
22050 QQ=PN                        :rem 118
22060 GOSUB 5000: IF WD$<>"" THEN22080
                                   :rem 176
22070 PRINT "I NEED TO KNOW THE NEW NAME"
      : ER=-1: RETURN              :rem 133
22080 GOSUB 6000                    :rem 66
22090 IF PN<>0 THEN PRINT "YOU HAVE ALREA
      DY USED THAT NAME": ER=-1: RETURN
                                    :rem 0
```

```
22100 PN$(QQ)=WD$: PRINT "RENAMING OK": R
      ETURN                          :rem 182
23000 REM LOAD COMMAND               :rem 248
23010 MD$="R": GOSUB 7000: IF ER THEN 230
      60                             :rem 137
23020 INPUT#2, NP                    :rem 166
23030 IF (NP+PC)>FF THEN PRINT "TOO MANY
      {SPACE}PROCEDURES": ER=-1: GOTO 230
      60                             :rem 251
23040 FOR K= 1 TO NP: INPUT#2, PN$(PC+K),
       PR$(PC+K): NEXT: PC=PC+NP     :rem 108
23050 PRINT NP "PROCEDURES LOADED":rem 14
23060 CLOSE 2: CLOSE 15: RETURN      :rem 211
24000 REM SAVE COMMAND                :rem 8
24010 MD$="W": GOSUB 7000: IF ER THEN 240
      40                             :rem 142
24020 PRINT#2, PC: FOR K= 1 TO PC: PRINT#
      2, PN$(K): PRINT#2, PR$(K): NEXT
                                     :rem 114
24030 PRINT PC "PROCEDURES SAVED":rem 204
24040 CLOSE 2: CLOSE 15: RETURN      :rem 210
25000 REM SCRATCHFILE COMMAND         :rem 2
25010 ER=0: GOSUB 5000: IF WD$<>"" THEN 2
      5030                           :rem 234
25020 PRINT "YOU MUST SUPPLY A FILENAME":
       ER=-1: RETURN                 :rem 5
25030 OPEN 15,8,15                   :rem 136
25040 PRINT#15, "S0:" + WD$ + ".TURTLE":
      {SPACE}INPUT#15, QQ,T$,ZZ,ZZ:rem 42
25050 IF (QQ>19)AND(QQ<>62) THEN PRINT "I
      'M HAVING TROUBLE WITH THE DISK":ER
      =-1                            :rem 25
25060 CLOSE 15: RETURN               :rem 243
```

Program 2: Turtle Data

```
100 REM TURTLE DATA, FOR USE WITH TURTLE
    {SPACE}GRAPHICS INTERPRETER. USE TURT
    LE BOOT                          :rem 95
110 REM TO LOAD.                     :rem 103
120 FOR K= 3584 TO 4095: READ J: POKE K,
    {SPACE}J: NEXT                   :rem 142
200 REM HEADING 0                    :rem 150
210 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,24,0,0
                                     :rem 239
220 DATA 60,0,0,126,0,0,255,0,1,255,128,3
    ,255,192,7,255,224,15,255,240,31,255
                                     :rem 171
230 DATA 248,0,0,0,0,0,0,0           :rem 209
240 REM HEADING 45                   :rem 211
250 DATA 0,63,255,0,31,255,0,15,255,0,7,2
    55,0,3,255,0,1,255,0,0,127,0,0,63,0
                                     :rem 90
260 DATA 0,31,0,0,7,0,0,3,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                     :rem 252
270 DATA 0,0,0,0,0                   :rem 83
280 REM HEADING 90                   :rem 215
290 DATA 0,0,0,0,0,0,0,224,0,0,240,0,0,248,
    0,0,254,0,0,255,0,0,255,128,0,255,192
                                     :rem 163
300 DATA 0,255,224,0,255,224,0,255,192,0,
    255,128,0,255,0,0,254,0,0,248,0,0,240
                                     :rem 199
310 DATA 0,0,224,0,0,0,0,0,0,0,0,0,0,0,0
                                     :rem 78
320 REM HEADING 135                   :rem 2
330 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0
                                     :rem 27
340 DATA 7,0,0,31,0,0,63,0,0,127,0,1,255,
    0,3,255,0,7,255,0,15,255,0,31,255,0
                                     :rem 76
350 DATA 63,255,0                    :rem 63
360 REM HEADING 180                   :rem 6
370 DATA 0,0,0,31,255,248,15,255,240,7,25
    5,224,3,255,192,1,255,128,0,255,0,0
                                     :rem 128
380 DATA 126,0,0,60,0,0,24,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                     :rem 58
390 DATA 0,0,0,0,0,0,0,0,0           :rem 198
400 REM HEADING 235                   :rem 2
410 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,192,0,0
                                     :rem 203
420 DATA 224,0,0,248,0,0,252,0,0,254,0,0,
    255,128,0,255,192,0,255,224,0,255,240
                                     :rem 199
430 DATA 0,255,248,0,255,252,0,0     :rem 20
440 REM HEADING 270                   :rem 5
450 DATA 0,0,0,0,0,0,0,0,14,0,0,30,0,0,62
    ,0,0,254,0,1,254,0,3,254,0,7,254,0,15
                                     :rem 128
460 DATA 254,0,15,254,0,7,254,0,3,254,0,1
    ,254,0,0,254,0,0,62,0,0,30,0,0,14,0,0
                                     :rem 159
470 DATA 0,0,0,0,0,0,0,0,0           :rem 105
480 REM HEADING 315                   :rem 9
490 DATA 255,252,0,255,248,0,255,240,0,25
    5,224,0,255,192,0,255,128,0,254,0,0
                                     :rem 134
500 DATA 252,0,0,248,0,0,224,0,0,192,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                     :rem 82
510 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0 :rem 212
600 REM SPLITSCREEN ROUTINE          :rem 236
610 FOR K= 49152 TO 49349: READ J: POKE K
    , J: NEXT                        :rem 254
620 DATA 173,25,208,141,25,208,41,1,208,3
    ,76,188,254,173,18,208,16,18,169,21
                                     :rem 166
630 DATA 141,24,208,169,27,141,17,208,169
    ,1,141,18,208,76,188,254,169,25,141
                                     :rem 178
640 DATA 24,208,169,59,141,17,208,169,209
    ,141,18,208,24,165,214,105,236,16
                                     :rem 75
650 DATA 3,32,83,192,165,197,201,4,208,3,
    238,32,208,201,5,208,3,238,33,208,32
                                     :rem 200
660 DATA 132,192,76,49,234,165,2,162,0,15
    7,0,4,232,208,250,157,0,5,232,208,250
                                     :rem 245
670 DATA 157,0,6,232,208,250,162,31,157,0
    ,7,202,16,250,169,32,162,201,157,31,7
                                     :rem 238
680 DATA 202,208,250,24,160,0,162,20,32,2
    40,255,96,162,39,165,2,157,248,6,202
                                     :rem 199
690 DATA 16,250,96,24,169,32,133,252,169,
    0,133,251,168,145,251,200,208,251,230
                                      :rem 2
700 DATA 252,165,252,201,64,208,1,96,152,
    240,239,120,169,127,141,13,220,169,1
                                     :rem 198
710 DATA 141,26,208,169,192,141,21,3,169,
    0,141,20,3,169,1,141,18,208,88,96
                                     :rem 58
```

Program 3: Turtle Boot

```
10 REM TURTLE BOOT                          :rem 89
20 POKE 53281, 6                            :rem 246
30 PRINT CHR$(147); CHR$(154) TAB(10) "TU
   RTLE GRAPHICS BOOT": PRINT: PRINT
                                            :rem 197
40 PRINT "THIS PROGRAM WILL LOAD AND RUN
   {SPACE}THE"                              :rem 134
50 PRINT "TURTLE DATA AND INTERPRETER PRO
   GRAMS.": PRINT                           :rem 183
60 PRINT "WHILE THEY ARE LOADING THE SCRE
   EN WILL"                                 :rem 153
70 PRINT "BLANK.": PRINT                    :rem 149
80 PRINT "DO NOT REMOVE THE DISK UNTIL TH
   E"                                       :rem 4
90 PRINT "INTERPRETER PROMPTS YOU FOR YOU
   R FIRST"                                 :rem 126
100 PRINT "COMMAND.": PRINT: PRINT: POKE
    {SPACE}198, 0                           :rem 132
110 PRINT "PRESS " CHR$(18) "SPACE" CHR$(
    146) " WHEN READY"                      :rem 51
120 GETA$: IF A$="" THEN 120                :rem 73
130 Q$=CHR$(34): D$=CHR$(17)                :rem 152
140 PRINT CHR$(147); CHR$(31); D$; D$; D$
     "POKE 16384, 0: POKE 44, 64: NEW"
                                            :rem 74
150 PRINT D$; D$ "LOAD" Q$ "TURTLE GRAPHI
    C 2" Q$ ",8"                            :rem 120
160 PRINT D$; D$; D$; D$; D$ "RUN":rem 81
170 PRINT D$; D$ "LOAD" Q$ "TURTLE GRAPHI
    C 1" Q$ ",8"                            :rem 121
180 PRINT D$; D$; D$; D$; D$ "RUN" CHR$(1
    9)                                      :rem 15
190 FOR K= 1 TO 7: POKE 630+K, 13: NEXT:
    {SPACE}POKE 198, 7                      :rem 3
```

Teaching Your Computer English

(Article on page 126.)

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.

```
1 SYS65517:AA=PEEK(781):GOTO62      :rem 34
2 DIML(35),M(6,6),N$(35),R$(6):FORX=1TO6:
  FORY=1TO6:READM(X,Y):NEXTY,X      :rem 155
3 FORX=1TO29:READN$(X):NEXT:FORX=1TO29:RE
  ADL(X):NEXT:FORX=1TO6:READR$(X):NEXT
                                    :rem 63
4 L=2                               :rem 239
5 L5=36879:IFAA=40THENL5=53281      :rem 224
6 IFL=1THENPOKEL5,104:PRINT"{WHT}":rem 47
7 IFL=2THENPOKEL5,125:PRINT"{BLK}"
                                    :rem 191
8 IFL=3THENPOKEL5,62:PRINT"{BLK}":rem 145
9 IFL=4THENPOKEL5,47:PRINT"{BLK}":rem 150
10 IFL=5THENPOKEL5,11:PRINT"{WHT}":rem 43
11 IFL=6THENPOKEL5,94:PRINT"{BLK}"
                                    :rem 195
12 PRINT"{CLR}EEEEEEEEEEEEEEEEEEEEE":PRI
   NT"YOU ARE IN THE":PRINTR$(L)    :rem 231
13 PRINT"YOU CAN SEE"               :rem 225
14 FORX=1TO29:IFL(X)=LTHENPRINTN$(X)
                                    :rem 62
15 NEXT                             :rem 165
16 PRINT"EEEEEEEEEEEEEEEEEEEEE":GOTO66
                                    :rem 5
17 PRINT"WHAT IS YOUR":INPUT"COMMAND";A$
                                    :rem 5
18 FORX=1TOLEN(A$):IFMID$(A$,X,1)=" "ANDL
   EN(A$)>X+1THENB$=MID$(A$,X+1,2):rem 75
19 NEXT                             :rem 169
20 IFLEN(A$)>1THENA$=LEFT$(A$,2)    :rem 41
21 V$="GOGEPUTAHIINLO"              :rem 125
22 N$="NOSOEAWEUPDOBECHTAPITHSEGRMUBUARSW
   SHMASPOIDRTRLEBONECARISP"        :rem 138
23 FORY=1TOLEN(V$):IFMID$(V$,Y,2)=A$THENV
   =Y+1:V=V/2                       :rem 114
24 NEXT:FORX=1TOLEN(N$):IFMID$(N$,X,2)=B$
   THENN=X+1:N=N/2                  :rem 194
25 NEXT:ONVGOTO26,29,39,29,46,53,55,39
                                    :rem 146
26 IFN>6THENPRINT"YOU CAN'T":GOTO17
                                    :rem 222
27 FORX=1TO6:IFN=XANDM(L,N)>0THENL=M(L,N)
                                    :rem 92
28 NEXT:GOTO5                       :rem 81
29 IFN=7ORN=9ORN=11THENPRINT"YOU CAN'T":P
   RINT"EEEEEEEEEEEEEEEEEEEEE":GOTO17
                                    :rem 205
30 IFN=14ORN=15ORN=22ORN=23THENPRINT"YOU
   {SPACE}CAN'T":PRINT"EEEEEEEEEEEEEEEEEE
   EEE"                             :rem 216
31 IFN=14ORN=15ORN=22ORN=23THEN17    :rem 45
32 IFB$="AR"THENL(16)=0              :rem 14
33 IFN=24THENPRINT"YOU CAN'T":PRINT"EEEEE
   EEEEEEEEEEEEEEEE":GOTO17         :rem 4
34 IFN=26THENPRINT"CAN'T, IT RAN AWAY":PR
   INT"EEEEEEEEEEEEEEEEEEEEE":L(26)=10
                                    :rem 232
35 IFN=26THEN17                     :rem 129
36 IFN=27THENPRINT"TOO HOT":PRINT"EEEEEEE
   EEEEEEEEEEEEEE":GOTO17           :rem 157
37 FORX=7TO29:IFN=XANDL(X)=LTHENL(X)=0
                                    :rem 185
38 NEXT:GOTO5                       :rem 82
39 IFN=24ANDL=5ANDL(23)=10THENPRINT"{CLR}
   YOU MADE IT!!":END               :rem 135
40 IFL=5ANDL(23)=LANDN=24THENPRINT"{CLR}
   {3 DOWN}TROLL PUTS YOU":PRINT"IN THE D
   UNGEON."                         :rem 48
41 IFL=5ANDL(23)=LANDN=24THENPRINT"
   {2 DOWN}{3 SPACES}GAME OVER":END
                                    :rem 93
42 FORX=1TO29:IFN=XANDL(N)<>0THENNEXT
                                    :rem 163
43 IFL(N)<>0THENPRINT"DON'T HAVE IT":PRIN
   T"EEEEEEEEEEEEEEEEEEEEE":GOTO17
                                    :rem 124
44 FORX=1TO29:IFN=XANDL(X)=0THENL(X)=L
                                    :rem 177
45 NEXT:PRINT"EEEEEEEEEEEEEEEEEEEEE":GOT
   O17                              :rem 124
46 IFL<>5ORN<>23THENPRINT"{DOWN}HURT YOUR
    HAND{DOWN}":GOTO17              :rem 108
47 IFL=5ANDN=23ANDL(17)<>0THENPRINT"
   {DOWN}HURT YOUR HAND{DOWN}"      :rem 204
48 IFL=5ANDN=23ANDL(17)=0ANDL(28)<>0THENP
   RINT"{DOWN}YOU CAN'T{DOWN}"      :rem 177
49 IFL=5ANDN=23ANDL(17)=0ANDL(28)=0ANDL(2
   5)<>0THENPRINT"{DOWN}YOU CAN'T"
                                    :rem 229
```

```
50 IFL=5ANDN=23ANDL(17)=0ANDL(28)=0ANDL(2
   5)=0THENPRINT"{DOWN}THE TROLL FLEES
   {DOWN}"                              :rem 68
51 IFL=5ANDN=23ANDL(17)=0ANDL(28)=0ANDL(2
   5)=0THENL(23)=10                     :rem 21
52 GOTO17                               :rem 8
53 PRINT"{CLR}{DOWN}YOU ARE CARRYING
   {DOWN}":FORX=1TO29:IFL(X)=0THENPRINTN$
   (X)"{DOWN}"                          :rem 110
54 NEXT:FORI=1TO1500:NEXT:GOTO6         :rem 11
55 IFN=9ANDL=1THENL(28)=L:GOTO12        :rem 223
56 IFN<>14THEN59                        :rem 196
57 PRINT"{DOWN}THE KEY TO WHAT YOU":PRINT
   "SEEK IS HIDDEN WHERE"               :rem 226
58 PRINT"THE KING SLEEPS{DOWN}":GOTO17
                                        :rem 0
59 PRINT"{DOWN}YOU CAN SEE{DOWN}":FORX=1T
   O29:IFN=XANDL(X)=L THEN61            :rem 149
60 FORX=1TO29:IFN=XANDL(X)=L THEN61
                                        :rem 180
61 NEXT:FORI=1TO1000:NEXT:GOTO12        :rem 49
62 PRINT"{CLR}{BLK}YOU ARE IN A CASTLE":P
   RINT"GUARDED BY AN EVIL"             :rem 38
63 PRINT"TROLL.":PRINT"{DOWN}CAN YOU GET
   {SPACE}OUT?":PRINT"{2 DOWN}HIT ANY KEY
   "                                    :rem 218
64 GETZ$:IFZ$=""THEN64                  :rem 41
65 GOTO2                                :rem 214
66 PRINT"EXITS ARE:":FORY=1TO6:IFM(L,Y)>0
   THENPRINTN$(Y)                       :rem 27
67 NEXT:PRINT"EEEEEEEEEEEEEEEEEEEEE":GOT
   O17                                  :rem 128
68 REM M(X)                             :rem 72
69 DATA0,,,3,,2,4,5,6,,1,,,,1,,,6,,2,,,,,
   2,0,,,,,,,,2,3,0                     :rem 130
70 REM N$(X)                            :rem 102
71 DATA"NORTH","SOUTH","EAST","WEST","UP"
   ,"DOWN"                              :rem 97
72 DATA"BED","CHAIR","TABLE","PILLOW","TH
   RONE","SCEPTER","GRAIL","MURAL":rem 30
73 DATA"BUNK","ARMOR","SWORD","SHIELD","M
   ACE","SPEARS","OIL","DRAWBRIDGE"
                                        :rem 116
74 DATA"TROLL","DRAWBRIDGE LEVER","SPELL
   {SPACE}BOOK","EYE OF NEWT","CAULDRON"
                                        :rem 40
75 DATA"RING","SPELL"                   :rem 234
76 REM L(X)                             :rem 70
77 DATA10,10,10,10,10,10                :rem 170
78 DATA1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,5,5,
   5,6,6,6,10,10                        :rem 72
79 REM R$(X)                            :rem 115
80 DATA"KINGS BEDROOM","THRONE ROOM","KNI
   GHTS QUARTERS","ARMORY"              :rem 16
81 DATA"DRAWBRIDGE ROOM","SORCERER'S ROOM
   "                                    :rem 107
```

Trek

(Article on page 54.)

BEFORE TYPING...

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

```
10 POKE53280,6:POKE53281,0:PRINT"{GRN}"
                                        :rem 229
11 PRINT"{CLR}{17 SPACES}{10 DOWN}TREK"
                                        :rem 166
12 PRINT"{9 DOWN}{14 SPACES}PLEASE WAIT"
                                        :rem 188
13 GOSUB 5000                           :rem 169
14 PRINT"{11 SPACES}{2 DOWN}SKILL
   {2 SPACES}LEVEL (1-4) "              :rem 50
18 GETFQ$:IF FQ$<"1"OR FQ$>"4" THEN 18
                                        :rem 227
19 FQ=VAL(FQ$):PRINT"{CLR}"             :rem 203
20 GOSUB 500                            :rem 119
30 GOSUB 800                            :rem 123
35 GOSUB 880                            :rem 136
40 V=53248:POKE723,10:V1=2040:SS=0:EX=80:
   TI$="140000":POKE53275,48:BSC=0:rem 67
45 XX$={2 SPACES}"{HOME}{23 DOWN}":rem 88
50 POKE V+21,63                         :rem 13
60 POKEV1,215:POKEV1+1,215:POKEV1+2,215:P
   OKEV1+3,211:POKEV1+4,212:POKEV1+5,213
                                        :rem 49
70 POKEV+39,6:POKEV+40,10:POKEV+41,5:POKE
   V+42,15:POKEV+43,9:POKEV+44,9        :rem 191
80 POKE V+9,202:POKEV+11,202            :rem 29
90 GOSUB 900                            :rem 130
95 QI=.01:POKEV+6,80:E=100:POKEV+5,145:PO
   KEV+29,8:POKE53280,0:IFFQ>2THENQI=.06
                                        :rem 164
96 POKE 2046,214:POKE53293,10:POKEV+30,0
                                        :rem 210
97 POKE V+1,150:POKEV+3,155:POKEV+5,160:P
   OKEV+7,180:EP=180:POKE650,128:HIT=0
                                        :rem 187
99 SS=.1:EX=80:S=54272:POKE54296,9:POKES+
   5,9:POKES+6,0:POKES,240:POKES+1,33
                                        :rem 248
100 SYS 989:POKE53277,PEEK(53277)OR(2↑4):
    POKE53277,PEEK(53277)OR(2↑5)        :rem 221
110 SYS THRST:POKE53280,0:POKEV+7,PEEK(V+
    7)+1:IF PEEK(V+7)>185THEN POKEV+7,185
                                        :rem 119
112 SYS THRST:POKE49402,PEEK(V+30):SYS PH
    AS:SYS THRST                        :rem 77
114 SYS THRST:SYS BAM:SYS THRST:IF(PEEK(4
    9402)AND15)>8THENPOKE53280,2:HIT=HIT+
    1                                   :rem 158
116 SYS THRST:SYS MOVE:POKE54273,0:SYS PH
    AS:IF PEEK(197)=60 THEN HIT=HIT+1
                                        :rem 224
117 SYSTHRST:POKE54277,15:POKE54276,17:SY
    S THRST:SYSTHRST:POKE54278,129:rem 13
118 SYS890:POKE54296,4:PRINT"{HOME}
    {2 DOWN}{GRN}"SPC(31)"SCORE{HOME}":PR
    INT"{DOWN}"SPC(25),(TI-T1)-HIT
                                        :rem 113
119 SYS THRST:POKE53280,0:SYS PHAS:IF PEE
    K(V)<20 THEN POKEV+1,(PEEK(V+7))
                                        :rem 73
120 SYS THRST:SYS PHAS:POKE54273,20:IF PE
    EK(V+2)<20 THEN POKEV+3,(PEEK(V+7))
                                        :rem 213
121 SYS THRST:SYS PHAS:IF PEEK(V+4)<20 TH
    EN POKEV+5,(PEEK(V+7))              :rem 222
123 SYS THRST:SYS BAM:SYS PHAS:SYS THRST
                                        :rem 102
124 SYS890:SYS PHAS:POKE54276,16:POKE5427
    6,17:SYS THRST                      :rem 115
125 SYS THRST:ON FQ GOTO 130,129,128,127,
    126                                 :rem 129
```

```
126 SYS THRST:SYS MOVE:SYS MOVE:SYS MOVE:
    SYS MOVE:SYS MOVE:SYS PHAS    :rem 194
127 SYS THRST:SYS MOVE:SYS MOVE:SYS MOVE:
    SYS PHAS                      :rem 227
128 SYS THRST:SYS MOVE:SYS MOVE:SYS MOVE:
    SYS PHAS                      :rem 228
129 SYS THRST:SYS MOVE:SYS MOVE:SYS PHAS
                                  :rem 117
130 SYS THRST:SYS MOVE:SYS MOVE:SYS PHAS
                                  :rem 109
131 SYS THRST:SYS MOVE:SYS THRUST:SYS MOV
    E:IF RND(.)<QI THEN8000       :rem 37
132 SYS THRST:A5=PEEK(197):POKEV+7,PEEK(V
    +7)+1:IFPEEK(V+7)>185THENPOKEV+7,185
                                  :rem 184
133 SYS890:IFA5=22ANDABS(PEEK(V+6)-PEEK(V
    +10))<25ANDPEEK(V+7)>175THENGOSUB6000
                                  :rem 183
135 XY$=STR$(100-(HIT*15.1))+"{2 SPACES}"
                                  :rem 107
140 SYS890:PRINTXX$;"ENERGY TO SHIELDS ";
    XY$                           :rem 216
145 SYS THRST:SYS PHAS:SYS THRST:IF HIT*1
    5.1>100THEN200                :rem 142
180 SYS THRST:EX=EX+1:IF EX>150THEN EX=40
    :LC=LC+1:IF LC>1 THEN 4000    :rem 140
190 SYS THRST:SYS MOVE:SYS PHAS:POKE V+6,
    EX:SYS THRST                  :rem 186
195 SYS THRST:SYS PHAS:GOTO110    :rem 157
200 PRINTXX$;"SHIELDS COLLAPSED,
    {2 SPACES}STARSHIP DESTROYED{HOME}"
                                  :rem 117
205 S=54272:POKES+24,15:POKES+5,192:POKES
    +6,129:U1=255:POKES+2,75:POKE53280,2
                                  :rem 230
210 N=190-PEEK(V+7):FOR CRASH=1 TO NSTEP4
                                  :rem 116
215 POKES+1,U1:POKES+4,17:U1=U1-1:rem 184
220 FOR DL=1TO10:NEXT:C1=(PEEK(V+7)+4):C2
    =(PEEK(V+6)+4)                :rem 136
225 IF C2>225 THENCRASH=N         :rem 159
230 POKEV+7,C1:POKEV+6,C2:NEXT     :rem 86
231 GOSUB 300                     :rem 169
234 POKEV+21,PEEK(V+21)AND(255-8) :rem 87
235 POKES+24,15:POKES+1,15:POKES+4,128
                                  :rem 203
240 POKES+5,9:POKES+1,20:POKES+4,128
                                  :rem 101
250 FORDL=1TO700:NEXT              :rem 47
260 POKES+4,129:GOTO3000          :rem 119
300 POKE 53277,PEEK(53277)AND(255-2↑3)
                                  :rem 36
301 POKE 53271,PEEK(53271)AND(255-2↑3)
                                  :rem 25
305 POKE 2043,216:POKEV+42,7:FOR DELAY=1T
    O20:NEXT:POKES+4,129          :rem 103
310 POKEV+7,PEEK(V+7)-10:POKE 53271,PEEK(
    53271)OR(2↑3):FOR DELAY=1TO10:NEXT
                                  :rem 162
312 POKES+24,7:FORD=1TO20:NEXT:POKES+24,8
    :FORD=1TO40:NEXT:POKES+24,13  :rem 62
315 POKE 53277,PEEK(53277)OR(2↑3):FOR DEL
    AY=1TO60:NEXT                 :rem 175
316 POKES+24,7:FORD=1TO40:NEXT:POKES+24,8
    :FORD=1TO20:NEXT:POKES+24,13  :rem 66
320 POKE 53271,PEEK(53271)AND(255-2↑3)
                                  :rem 26
325 POKEV+42,1:FORD=13TO1STEP-.5:POKES+24
    ,D:NEXT:POKES+4,128:POKES+24,9:RETURN
                                  :rem 151
499 REM SPRITE SUBSECTION         :rem 96
500 FOR N=13440TO13502:READA:POKEN,A:NEXT
                                  :rem 213
510 FOR N=13504TO13566:READA:POKEN,A:NEXT
                                  :rem 225
515 FOR N=13568TO13630:READA:POKEN,A:NEXT
                                  :rem 232
520 FOR N=13632TO13694:READA:POKEN,A:NEXT
                                  :rem 230
525 FOR N=13760TO13822:READA:POKEN,A:NEXT
                                  :rem 230
527 FOR N=13696TO13759:READA:POKEN,A:NEXT
                                  :rem 249
528 FOR N=13824TO13886:READA:POKEN,A:NEXT
                                  :rem 244
530 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,48,0,0,48,0,0,48,0,0    :rem 74
535 DATA 48,0,0,48,0,0,48,0,0,48,0,0,48,0
    ,0,0,0,0,0,0,0,0,0,0,0,0,0    :rem 107
540 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,96,0,1,248,255,223,255
                                  :rem 224
545 DATA 127,223,255,15,135,192,7,143,128
    ,3,159,0,15,255,0,15,255,0,7,255,128
                                  :rem 206
550 DATA 7,255,0,3,255,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                  :rem 70
555 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,224,0,1,248,0,3,252,0,3
                                  :rem 247
560 DATA 254,224,7,255,16,15,255,8,31,255
    ,132,127,255,254,255,255,255,255,255
                                  :rem 231
565 DATA 255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,7,0,0,15,128,0,31,192,0,31,192,2
                                  :rem 80
570 DATA 63,240,7,127,248,15,191,248,15,2
    23,248,31,231,252,31,251,252,63,248
                                  :rem 165
575 DATA 254,63,255,6,127,255,248,255,255
    ,254,255,255,255,255,255,255,0,0,0,0,
    0                             :rem 64
580 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,24
    8,3,255,240,15,255,248,63,255,248,0,1
    5                             :rem 222
585 DATA 192,1,255,240,1,255,224,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                  :rem 16
590 DATA 0,0                      :rem 68
592 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,255,0,0
                                  :rem 30
593 DATA 255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                  :rem 215
595 DATA 0,0,0,0,112,0,3,252,0,15,255,128
    ,23,255,64,59,254,240,61,253,248,118
                                  :rem 185
596 DATA 171,252,251,255,60,255,254,220,1
    23,253,232,119,251,248,47,255,240,15
                                  :rem 220
597 DATA 223,224,7,223,224,3,223,192,3,21
    9,192,3,219,192,7,219,224,15,219,240
                                  :rem 215
598 DATA 0,0,0                    :rem 168
600 KI=49152:FOR N=KI TO 49204:READA:POKE
    N,A:NEXT                      :rem 134
610 DATA 162,210,173,30,208,141,62,192,41
    ,68,201,68,208,7,142,208,2,142,5,208
                                  :rem 193
620 DATA 96,173,62,192,41,66,201,66,208,7
```

```
    ,142,198,2,142,3,208,96,173,62,192,41
                                        :rem 19
630 DATA 65,201,65,208,246,142,188,2,142,
    1,208,96                            :rem 113
640 BAM=49407:FOR N=BAM TO 49459:READA:PO
    KEN,A:NEXT                          :rem 17
641 DATA 162,210,173,30,208,141,62,192,41
    ,12,201,12,208,7,142,208,2,142,5,208
                                        :rem 175
642 DATA 96,173,62,192,41,10,201,10,208,7
    ,142,198,2,142,3,208,96,173,62,192,41
                                        :rem 1
643 DATA 9,201,9,208,246,142,188,2,142,1,
    208,96                              :rem 17
650 MOVE=49232:FORN=MOVETO49275:READA:POK
    EN,A:NEXT                           :rem 216
660 DATA 173,7,208,205,1,208,176,6,206,1,
    208                                 :rem 115
665 DATA 76,97,192,238,1,208,205,3,208,17
    6                                   :rem 42
670 DATA 6,206,3,208,76,111,192,238,3,208
    ,205,5,208,176,4,206,5,208,96,238
                                        :rem 71
680 DATA 5,208,96                       :rem 78
690 PHAS=49472:FOR N=PHAS TO 49531:READA:
    POKE N,A:NEXT                       :rem 199
691 DATA 165,197,201,60,240,1,96,169,129,
    141,4,212,173,7,208,174,6,208,142,12
                                        :rem 215
693 DATA 208,141,13,208,160,127,140,21,20
    8,238,12,208,173,12,208,208,13,169
                                        :rem 105
695 DATA 128,141,4,212,169,63,141,21,208,
    76,0,192,160,70,136,192,0,208,251
                                        :rem 59
697 DATA 76,93,193                      :rem 142
700 RETURN                              :rem 119
799 REM MAIN LOOP MACHINE LANGUAGE:rem 37
800 FOR N=828 TO 886:READA:POKEN,A:NEXT
                                        :rem 41
810 DATA 174,188,2,202,202,142,188,2,142,
    0,208,174,198,2,202,202,202,142,198
                                        :rem 146
815 DATA 2,142,2,208,174,208,2,202,142,20
    8,2,142,4,208,174,188,2,174,213,2,202
                                        :rem 232
820 DATA 142,213,2,142,8,208,174,211,2,20
    2,142,211,2,142,10,208,76,49,234
                                        :rem 236
850 RETURN                              :rem 125
879 REM RESET HARDWARE INTERUPT VECTOR
                                        :rem 171
880 FOR N=989TO1002:READA:POKEN,A:NEXT
                                        :rem 86
885 DATA 120,169,60,141,20,3,169,3,141,21
    ,3,88,96,32                         :rem 2
890 REM THRUSTERS (THRST)               :rem 63
891 THRST=890:FORN=890TO951:READA:POKEN,A
    :NEXT                               :rem 216
892 DATA 165,197,201,2,240,13,165,197,201
    ,7,240,19,165,197,197,60,240,25,96
                                        :rem 136
893 DATA 174,7,208,202,224,95,240,247,142
    ,7,208,96,174,7,208,232,224,190,240
                                        :rem 179
894 DATA 247,142,7,208,96,173,7,208,74,74
    ,74,24,105,25,168,174,6,208,24,32
                                        :rem 93
895 DATA 240,255,96,32                  :rem 74
899 RETURN                              :rem 138
900 POKE53280,0:POKE53281,0             :rem 238
910 GOSUB 1100                          :rem 220
920 PRINT"{CLR}":GOSUB 980              :rem 90
930 PRINT"{20 DOWN}{2}{35 I}{HOME}";
                                        :rem 202
950 PRINT"{WHT}{7 SPACES}{4 +} {4 +}
    {SHIFT-SPACE}{4 +}{SHIFT-SPACE}{+}
    {2 SHIFT-SPACE}{+}{5 SPACES}HI SCORE
    {SPACE}";                           :rem 80
952 PRINT" LEVEL{3 SPACES}{+}{2 SPACES}
    {+}{2 SPACES}{+} {+}{4 SPACES}{+}
    {2 +}{4 SPACES}"BSC                 :rem 75
954 PRINT"{2 SPACES}"FQ"{4 SPACES}{+}
    {2 SPACES}{4 +} {3 +}{2 SPACES}{3 +}
    {15 SPACES}";                       :rem 171
956 PRINT"{9 SPACES}{+}{2 SPACES}{+} {+}
    {2 SPACES}{+}{4 SPACES}{+} {+}
    {15 SPACES}";                       :rem 148
958 PRINT"{9 SPACES}{+}{2 SPACES}{+}
    {2 +} {4 +} {+} {2 +}{14 SPACES}";
                                        :rem 212
960 PRINT"{40 O}";                      :rem 147
965 FOR SV=1TO12:PRINTSPC(30)"{WHT}{RVS}
    {J}":NEXT                           :rem 146
970 PRINTSPC(31)"{9 U}{HOME}"           :rem 151
971 PRINT"{4 DOWN}"SPC(31)"{CYN}CONTROLS
    {DOWN}"                             :rem 185
972 PRINTSPC(31)"{YEL}CRSR-DN="         :rem 226
973 PRINTSPC(33)"DOWN{DOWN}"            :rem 90
974 PRINTSPC(31)"CRSR-RT="              :rem 90
975 PRINTSPC(34)"UP{DOWN}"              :rem 202
977 PRINTSPC(31)"{RVS}T{OFF}=BEAM-UP"
                                        :rem 47
978 PRINTSPC(31)"SPACEBAR=";           :rem 205
979 PRINTSPC(32)" FIRE":GOSUB9000:GOTO100
    0                                   :rem 243
980 FOR SF=1TO 20:K=(RND(X)*25+RND(Y)*400
    ):POKE 1250+K,46:NEXT               :rem 242
990 FOR CF=55296TO55555:POKECF,4:NEXT
                                        :rem 209
992 FOR CF=55556TO55855:POKECF,1:NEXT
                                        :rem 210
994 FOR CF=55856TO56295:POKECF,7:NEXT
                                        :rem 220
998 RETURN                              :rem 138
1000 RETURN                             :rem 161
1099 REM TITLE                          :rem 57
1100 FORN=1TO30:PRINT"{DOWN}";:NEXT
                                        :rem 59
1110 PRINT"{CLR}":M=1:GOSUB980          :rem 119
1200 POKE53277,PEEK(53277)OR(2↑3):FOR X=2
     50TO3 STEP-3                       :rem 24
1210 M=M+3:POKE V+6,M : POKEV+7,X       :rem 55
1220 NEXT                               :rem 4
1230 PRINT"{HOME}{7 DOWN}{WHT}";        :rem 97
1240 PRINT"{10 SPACES}{3 +} {3 +} {3 +}
     {+} {2 +}"                         :rem 96
1250 PRINT"{11 SPACES}{+}{2 SPACES}{+}
     {+} {+}{3 SPACES}{+} {+}"          :rem 125
1260 PRINT"{11 SPACES}{+}{2 SPACES}{3 +}
     {SPACE}{3 +} {3 +}"                :rem 22
1270 PRINT"{11 SPACES}{+}{2 SHIFT-SPACE}
     {+}{SHIFT-SPACE}{-}{SHIFT-SPACE}{+}
     {3 SHIFT-SPACE}{+}{SHIFT-SPACE}{+}"
                                        :rem 181
1280 PRINT"{11 SPACES}{+}{2 SPACES}{+}
     {SHIFT-SPACE}{+}{SHIFT-SPACE}{3 +}
     {SHIFT-SPACE}{+}{SHIFT-SPACE}{2 +}"
                                        :rem 242
1285 M=1                                :rem 139
1290 FOR X=3TO250STEP3:M=M+3            :rem 147
1300 POKEV+6,M:POKEV+7,X                :rem 200
```

```
1310 NEXT                                :rem 4
1320 PRINT"{2 DOWN}";                    :rem 244
1371 PRINT"{DOWN}{8 SPACES}MISSION TO DEL
     TA MINOR"                           :rem 98
1375 FOR DELAY=1TO2000:NEXT              :rem 114
1379 GOSUB7000:RETURN                    :rem 53
1380 S=54272:FORL=STOS+24:POKEL,0:NEXT:PO
     KES+5,9:POKES+6,0:POKES+24,15:rem 71
1391 POKES+1,28:POKES,49:POKES+4,17
                                         :rem 70
1392 FORN=1TO125:NEXTN                   :rem 116
1393 POKES+4,16                          :rem 68
1394 FORN=1TO70:NEXTN                    :rem 69
1395 POKES+1,28:POKES,49:POKES+4,17
                                         :rem 74
1396 FORN=1TO125:NEXTN                   :rem 120
1397 POKES+4,16:FORL=STOS+24:POKEL,0:NEXT
     :RETURN                             :rem 112
3000 POKE2041,215:POKEV+21,0             :rem 38
3100 PRINT"{HOME}{11 DOWN}{OFF}{BLK}
     {10 SPACES}{YEL}G{RED}A{WHT}M{PUR}E
     {SPACE}{GRN}O{RED}V{CYN}E{BLU}R";
                                         :rem 105
3105 POKE54296,0:LC=0:POKE198,0          :rem 251
3110 PRINTXX$"{WHT}TYPE (E) TO END, TYPE
     {SPACE}(P) TO PLAY AGAIN"           :rem 9
3200 GETA$:IF A$="E"THEN POKE198,0:SYS198
                                         :rem 53
3205 IFA$<>"P"THEN3200                   :rem 192
3210 RESTORE:POKE2043,211:POKEV+21,63:T1=
     TI                                  :rem 87
3211 PRINT"{HOME}{11 DOWN}{27 SPACES}"
                                         :rem 102
3212 PRINTXX$"{39 SPACES}"               :rem 109
3213 PRINTXX$;:PRINT" SKILL LEVEL 1-4"
                                         :rem 249
3215 GETFQ$:IF FQ$<"1"OR FQ$>"4" THEN 321
     5                                   :rem 167
3216 FQ=VAL(FQ$)                         :rem 143
3217 PRINTXX$"{39 SPACES}"               :rem 114
3218 PRINT"{HOME}{2 DOWN}{GRN}"SPC(31)"SC
     ORE{HOME}":PRINT"{DOWN}"SPC(25),"
     {7 SPACES}"                         :rem 138
3220 PRINT"{HOME}{11 DOWN}{10 RIGHT}
     {5 SPACES}.{GRN}.{3 SPACES}.
     {3 SPACES}";:T1=TI:GOSUB920:GOTO95
                                         :rem 104
4000 B2=(TI-T1)-HIT:IFB2>BSCTHENBSC=B2
                                         :rem 50
4001 PRINT"{CLR}{YEL}";:POKEV+21,8:POKE53
     280,0:FORL=STOS+24:POKEL,0:NEXT:M=1:
     N=2                                 :rem 116
4004 POKE54272,40:POKE54296,15           :rem 144
4005 POKE54273,30+N:POKES+4,17:PRINT TAB(
     N)"M{4 SPACES}N B O[Y] [Y]O[Y] OP
     {SHIFT-SPACE}OP M{2 SPACES}N"
                                         :rem 206
4006 PRINT TAB(N)" M{SHIFT-SPACE} N
     {2 SHIFT-SPACE}B{SHIFT-SPACE}[H]
     {3 SPACES}[H]{2 SPACES}[H][N] L@
     {2 SPACES}MN{3 SPACES}"             :rem 213
4007 PRINT TAB(N)"{2 SPACES}MN{3 SPACES}B
      L[P]{2 SPACES}[H]{2 SPACES}L@
     {SHIFT-SPACE}[H]M{3 SPACES}[H]
     {3 SPACES}{HOME}";                  :rem 71
4008 N=N+M:IF N>7 OR N<1 THEN M=M*(-1):N1
     =N1+1:IFN1>4THEN 4010               :rem 99
4009 FORD=1TO10:NEXT:POKES+4,16:GOTO 4005
                                         :rem 205
4010 POKES+4,16:PRINT"{5 DOWN}{YEL}
     {11 SPACES}PEGASUS REFUELED!!"
                                         :rem 221
4020 PRINT"{11 SPACES}CONTINUING ON MISSI
     ON"                                 :rem 84
4030 PRINT"{11 SPACES}TO DELTA MINOR..."
                                         :rem 180
4035 M=1                                 :rem 135
4036 FOR X=3TO250STEP3:M=M+3             :rem 148
4037 POKEV+6,M:POKEV+7,X                 :rem 210
4038 NEXT                                :rem 14
4045 M=1                                 :rem 136
4046 FOR X=250TO3STEP-3:M=M+3            :rem 194
4047 POKEV+6,M:POKEV+7,X                 :rem 211
4048 NEXT                                :rem 15
4050 FORI=1TO1000:NEXTI:GOSUB 920:GOTO 30
     00                                  :rem 24
4999 REM RELOCATE CHARACTER SET          :rem 139
5000 POKE56334,0:POKE1,51                :rem 129
5020 FOR ADD=14336TO16384                :rem 94
5030 POKE ADD,PEEK(ADD+38912):NEXT ADD
                                         :rem 159
5040 POKE1,55:POKE56334,129:POKE53272,(PE
     EK(53272)AND240)OR14:RETURN         :rem 224
6000 POKEV+44,7:SYS THRST:HIT=HIT-2
                                         :rem 175
6010 SYS THRST:POKEV+44,9:RETURN         :rem 44
7000 FOR CD=1TO30:PRINT"{DOWN}";:NEXT
                                         :rem 121
7100 PRINT"[6]{5 SPACES}CAPTAINS LOG":PRI
     NT"{5 SPACES}STARDATE "TI$          :rem 139
7150 PRINT"{5 SPACES}[15 Y]"             :rem 87
7200 PRINT"{6 SPACES}THE PEGASUS IS EN RO
     UTE TO"                             :rem 244
7210 PRINT"{5 SPACES}DELTA MINOR. OUR MIS
     SION:"                              :rem 10
7220 PRINT"{5 SPACES}AID A FEDERATION RES
     EARCH"                              :rem 217
7230 PRINT"{5 SPACES}OUTPOST IN COMBATING
      A"                                 :rem 71
7250 PRINT"{5 SPACES}MUTANT VIRUS THAT IS
     "                                   :rem 222
7270 PRINT"{5 SPACES}THREATENING THE CONT
     INUED"                              :rem 100
7280 PRINT"{5 SPACES}EXISTENCE OF THE OUT
     POST.":PRINT                        :rem 243
7300 PRINT"{6 SPACES}WHILE EN ROUTE, THE
     {SPACE}SHIP"                        :rem 119
7310 PRINT"{5 SPACES}SUFFERED DAMAGE TO T
     HE"                                 :rem 19
7320 PRINT"{5 SPACES}MAIN POWER HOUSINGS.
      AS A"                              :rem 194
7330 PRINT"{5 SPACES}RESULT, WE ARE FORCE
     D TO"                               :rem 115
7340 PRINT"{5 SPACES}MINE NEW DILITHIUM C
     RYSTALS"                            :rem 208
7350 PRINT"{5 SPACES}ON A NEARBY PLANETOI
     D. THE"                             :rem 238
7360 PRINT"{5 SPACES}PLANETOID, HOWEVER,
     {SPACE}IS"                          :rem 85
7370 PRINT"{5 SPACES}GUARDED BY NUMEROUS
     {SPACE}LIFELESS"                    :rem 14
7380 PRINT"{5 SPACES}DRONES...":PRINT
                                         :rem 191
7390 PRINT"{WHT}{5 SPACES}HIT ANY KEY TO
     {SPACE}CONTINUE"                    :rem 103
7400 POKE198,0:GOSUB1380                 :rem 126
7500 GETA$:IFA$=""THEN7500               :rem 187
7501 T1=TI:RETURN                        :rem 70
8000 POKE53269,PEEK(53269)AND250         :rem 21
8005 A1=PEEK(53278):J1=0:POKEV+3,200:POKE
     2041,214:POKE710,1:POKE53288,7
                                         :rem 31
```

```
8010 FOR N=1TO60:SYS THRST:SYS MOVE:SYS M
     OVE:SYS THRST:SYS THRST:SYS THRST
                                    :rem 141
8015 POKE2041,210:POKES+4,129       :rem 102
8020 SYSTHRST:IF(PEEK(53278)AND10)=10THEN
     POKE53288,10:POKE2041,215:GOTO200
                                    :rem 183
8025 POKE2041,214                   :rem 136
8030 SYSMOVE:SYSTHRST:SYSMOVE:SYSTHRST:NE
     XT:POKE2041,215:POKE53288,10:rem 122
8035 POKES+4,128:POKE53269,63:GOTO132
                                    :rem 136
9000 PRINTXX$"{6 UP}":PRINTSPC(35)"{WHT}
     {5 U}";                         :rem 86
9010 POKE 646,9                     :rem 254
9020 PRINTSPC(35)"{RVS}{5 SPACES}";
                                    :rem 136
9030 PRINTSPC(35)"{RVS}{5 SPACES}";
                                    :rem 137
9035 PRINTSPC(35)"{RVS}{5 SPACES}";
                                    :rem 142
9040 RETURN                         :rem 173
```

VIC Music Tutor

Requires minimum of 8K memory expansion.

(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.

```
10 DI=INT(FRE(8)/16):DIMNN%(DI),NV(DI),N$
   (DI),V$(DI)                      :rem 196
20 PRINT"{CLR}{3 DOWN}"SPC(159)"MUSIC TUT
   OR"                              :rem 50
30 FORI=1TO 2000:NEXT               :rem 223
40 PRINT"{CLR}{DOWN}DO YOU WANT TO","
   {DOWN}1-PLAYBACK AN OLD TUNE2-RECORD A
    NEW ONE"                        :rem 65
50 INPUTA:ONAGOTO760,60:GOTO40      :rem 141
60 INPUT"{DOWN}SONG NUMBER";X:PRINT"
   {DOWN}NAME OF SONG":INPUTX$:C=1:rem 24
70 XX=1:PRINT"{CLR}YOU HAVE A MAXIMUM OF
   {SPACE}"DI"NOTES"                :rem 146
80 XX=XX+1:IFXX<=3THEN85            :rem 34
83 GOTO70                           :rem 11
85 PRINT"{DOWN}NOTE #"C:INPUTN$     :rem 39
90 IFN$="£"THENC=C-1:GOTO700        :rem 161
100 IFN$="R"THENN=0                 :rem 79
110 IFN$="C"THENN=135               :rem 170
120 IFN$="CS"ORN$="DF"THENN=143     :rem 27
130 IFN$="D"THENN=147               :rem 176
140 IFN$="DS"ORN$="EF"THENN=151     :rem 30
150 IFN$="E"THENN=159               :rem 182
160 IFN$="F"THENN=163               :rem 179
170 IFN$="FS"ORN$="GF"THENN=167     :rem 44
180 IFN$="G"THENN=175               :rem 185
190 IFN$="GS"ORN$="AF"THENN=179     :rem 44
200 IFN$="A"THENN=183               :rem 171
210 IFN$="AS"ORN$="BF"THENN=187     :rem 31
220 IFN$="B"THENN=191               :rem 173
230 IFN$="C1"THENN=195              :rem 228
240 IFN$="CS1"ORN$="DF1"THENN=199:rem 139
250 IFN$="D1"THENN=201              :rem 219
260 IFN$="DS1"ORN$="EF1"THENN=203:rem 129
270 IFN$="E1"THENN=207              :rem 228
280 IFN$="F1"THENN=209              :rem 232
290 IFN$="FS1"ORN$="GF1"THENN=212:rem 136
300 IFN$="G1"THENN=215              :rem 223
310 IFN$="GS1"ORN$="AF1"THENN=217:rem 129
320 IFN$="A1"THENN=219              :rem 223
330 IFN$="AS1"ORN$="BF1"THENN=221:rem 121
340 IFN$="B1"THENN=223              :rem 221
350 IFN$="C2"THENN=225              :rem 226
360 IFN$="CS2"ORN$="DF2"THENN=227:rem 136
370 IFN$="D2"THENN=228              :rem 232
380 IFN$="DS2"ORN$="EF2"THENN=229:rem 142
390 IFN$="E2"THENN=231              :rem 229
400 IFN$="F2"THENN=232              :rem 223
410 IFN$="FS2"ORN$="GF2"THENN=233:rem 135
420 IFN$="G2"THENN=235              :rem 229
430 IFN$="GS2"ORN$="AF2"THENN=236:rem 135
440 IFN$="A2"THENN=237              :rem 227
450 IFN$="AS2"ORN$="BF2"THENN=238:rem 134
460 IFN$="B2"THENN=239              :rem 232
470 IFN$="C3"THENN=240              :rem 227
480 PRINT"NOTE VALUE"               :rem 32
490 INPUTV$                         :rem 167
500 V$(C)=V$                        :rem 90
510 IFRIGHT$(V$,1)="D"THEND=1:V$=LEFT$(V$
   ,1)                              :rem 253
520 IFV$="W"THENV=16                :rem 161
530 IFV$="H"THENV=8                 :rem 100
540 IFV$="Q"THENV=4                 :rem 106
550 IFV$="S"THENV=1                 :rem 106
560 IFV$="E"THENV=2                 :rem 94
570 IFV$="T"THENV=.5                :rem 159
580 V$(C)=V$                        :rem 98
590 IFV$="W"ORV$="H"ORV$="Q"ORV$="E"ORV$=
   "S"ORV$="T"THEN610               :rem 214
600 V=VAL(V$)                       :rem 215
610 IFD=1THENV$=V$+"D":V=V*1.5:D=0
                                    :rem 183
620 PRINT"{RVS}NOTE #"C"{2 SPACES}{RVS}VA
   LUE":PRINTN$,V$:PRINT"{RVS}{PUR}
   {6 SPACES}CORRECT Y/N{5 SPACES}{BLU}"
                                    :rem 137
630 GETT$:IFT$<>"N"ANDT$<>"Y"THEN630
                                    :rem 104
640 IFT$="N"THENPRINT"{RVS}{5 SPACES}ENTE
   R AGAIN":GOTO70                  :rem 58
650 N$(C)=N$                        :rem 80
660 NN%(C)=N:NV(C)=V:IFSS=1THENRETURN
                                    :rem 109
670 C=C+1:IFC=DI-5THENPRINT"{RVS}FIVE MOR
   E NOTES BEFOREMEMORY IS FULL"    :rem 71
680 IFC=DITHENPRINT"{RVS}LAST NOTE":C=C-1
   :GOTO700                         :rem 27
690 GOTO80                          :rem 64
700 PRINT"{CLR}{DOWN}DO YOU WANT TO HEAR
   {SPACE}ITY/N"                    :rem 12
710 GETT$:IFT$=""THEN710            :rem 121
715 IF T$<>"Y"ANDT$<>"N"THEN 710    :rem 217
720 IFT$="Y"THEN850                 :rem 70
721 GOTO1130                        :rem 152
725 INPUT"DISK OR TAPE";Q$          :rem 24
726 IF LEFT$(Q$,1)<>"D"ANDLEFT$(Q$,1)<>"T
   "THEN762                         :rem 199
727 IF LEFT$(Q$,1)="D"THENQW=8:GOTO729
                                    :rem 199
728 QW=1                            :rem 183
729 INPUT "FILENAME";X$:IF X$=""THEN729
                                    :rem 5
```

```
730 OPEN1,QW,1,X$:PRINT#1,X:PRINT#1,X$:PR
    INT#1,C                          :rem 194
740 FORI=1TOC:PRINT#1,NN%(I):PRINT#1,NV(I
    ):NEXT:CLOSE1                    :rem 160
745 IF QW=8 THEN OPEN15,8,15:INPUT#15,Z:C
    LOSE15                           :rem 248
746 IF Z<>0 THEN PRINT"{CLR}DISK ERROR!!"
    :GOTO725                         :rem 207
750 PRINT"{DOWN}SONG SAVED":STOP     :rem 168
760 INPUT"{CLR}NAME OF SONG";X$      :rem 168
762 INPUT "DISK OR TAPE";Q$          :rem 25
763 IF LEFT$(Q$,1)<>"D"ANDLEFT$(Q$,1)<>"T
    "THEN762                         :rem 200
764 IF LEFT$(Q$,1)="D"THENQW=8:GOTO770
                                     :rem 196
765 QW=1                             :rem 184
770 OPEN1,QW,0,X$                    :rem 217
780 INPUT#1,X,X$,C                   :rem 30
790 PRINT"{CLR}SONG #";X:PRINTX$     :rem 52
800 FORI=1TOC                        :rem 28
810 INPUT#1,NN%(I),NV(I)             :rem 110
820 NEXT:CLOSE1                      :rem 186
825 IF QW=8 THEN OPEN15,8,15:INPUT#15,Z:C
    LOSE15:IF Z<>0 THEN PRINT"{CLR}DISK E
    RROR!!":STOP                     :rem 206
830 GOSUB1380                        :rem 231
840 PRINT"{RVS}SONG LOADED"          :rem 95
844 FORIJ=1 TO1500:NEXT              :rem 106
845 GOTO700                          :rem 113
850 Q=1:Y=C                          :rem 111
860 PRINT"{CLR}{3 DOWN}CHOOSE TEMPO"
                                     :rem 123
870 PRINT"{DOWN}0 THRU 10":PRINT"0=FASTES
    T...10=SLOWEST"                  :rem 64
880 INPUTM                           :rem 125
890 DU=(M+3)*10                      :rem 254
900 PRINT"{DOWN}SELECT PITCH",,,"1=LOW",,
    "2=MEDIUM",,"3=HIGH":INPUTW      :rem 244
910 SP=36873+W:L=36878               :rem 218
920 PRINT"{CLR}WHEN YOU ARE READY TO HEAR
     THE SONG PRESS{3 SPACES}{RVS}RETURN"
                                     :rem 213
930 GETG$:IFG$<>CHR$(13)THEN930      :rem 22
940 POKEL,15                         :rem 170
950 FORI=QTOY                        :rem 88
960 POKESP,NN%(I):FORH=1TODU*NV(I):NEXT:P
    OKESP,0                          :rem 84
970 IFSS=1THENGOSUB1280              :rem 189
980 NEXT                             :rem 224
990 POKEL,0:POKESP,0                 :rem 225
1000 PRINT"{CLR}{3 DOWN}{PUR}{RVS}END OF
    {SPACE}SONG{BLU}":PRINT"{2 DOWN}PRES
    S {RVS}{PUR}RETURN{OFF}{BLU} TO REPL
    AYEXACTLY"                       :rem 75
1010 PRINT"{DOWN}PRESS ANY OTHER KEY TOMA
    KE CHANGES"                      :rem 62
1020 GETR$:IFR$=""THEN1020           :rem 203
1030 IFR$=CHR$(13)THEN940            :rem 136
1035 SS=0                            :rem 220
1040 PRINT"{CLR}DO YOU WANT TO HEAR ITAGA
    IN?"                             :rem 242
1050 INPUT"Y/N";W$                   :rem 38
1060 IFW$="N"THEN1130                :rem 148
1070 PRINT"{CLR}{DOWN}1-JUST PART","2-THE
     WHOLE SONG":INPUTK              :rem 147
1080 ONKGOTO1090,850                 :rem 125
1090 PRINT"{DOWN}THERE ARE";C;"NOTES.":PR
    INT"ENTER THE START AND":PRINT"ENDIN
    G NOTES."                        :rem 242
1100 INPUT"START";Q                  :rem 176
1110 INPUT"ENDING NOTE";Y            :rem 22
1120 GOTO860                         :rem 155
1130 PRINT"{CLR}DO YOU WANT TO SINGLE STE
    P THROUGH THE TUNE?Y/N":GOSUB1360
                                     :rem 107
1140 IFC$="Y"THEN1180                :rem 143
1150 IFC$="N"THENPRINT"{CLR}DO YOU WANT T
    O ADD ANYNOTES Y/N":GOSUB1360:rem 32
1160 IFC$="Y"THENC=C+1:GOTO70        :rem 192
1170 IFC$="N"THENPRINT"{DOWN}DO YOU WANT
    {SPACE}TO SAVE{3 SPACES}THE TUNE
    {2 SPACES}Y/N?":GOTO1250         :rem 168
1180 PRINT"{CLR}PRESS F1 TO PLAY NOTE":PR
    INT"{DOWN}IF YOU WANT TO CHANGE THAT
     NOTE PRESS F7."                 :rem 11
1190 PRINT"THEN ENTER REPLACEMENTNOTE AND
     VALUE":SS=1                     :rem 77
1200 PRINT"{DOWN}PRESS F2 TO ADD A NEW NO
    TE IN"                           :rem 14
1210 PRINT"{DOWN}PRESS F5 TO STOP THE
    {2 SPACES}SINGLE NOTE MODE"      :rem 149
1220 PRINT"{RVS}{DOWN} HIT ANY KEY TO CON
    T. "                             :rem 116
1230 GETI$:IFI$=""THEN1230           :rem 191
1240 GOTO 1070                       :rem 200
1250 GOSUB1360                       :rem 18
1260 IFC$="Y"THEN725                 :rem 102
1270 END                             :rem 161
1280 PRINT"NOTE #"I                  :rem 62
1290 PRINT"{RVS}{RED}{2 SPACES}NOTE
    {2 SPACES}{OFF}","{RVS} VALUE {OFF}
    {BLU}"                           :rem 67
1300 PRINTN$(I),V$(I)                :rem 157

1310 GETU$:IFU$<>CHR$(133)ANDU$<>CHR$(135
    )ANDU$<>CHR$(136)ANDU$<>CHR$(137)THE
    N1310                            :rem 205
1320 IFU$=CHR$(133)THENRETURN         :rem 3
1330 IFU$=CHR$(137)THENGOTO1900      :rem 43
1340 IFU$=CHR$(135)THEN1000          :rem 232
1350 CT=C:C=I:PRINT"{RVS}ENTER REPLACEMEN
    T":GOSUB80:C=CT:RETURN           :rem 59
1360 INPUTC$:IFC$<>"Y"ANDC$<>"N"THEN1360
                                     :rem 71
1370 RETURN                          :rem 171
1380 FORI=1TOC                       :rem 80
1390 IFNN%(I)=135THENN$(I)="C":GOTO1760
                                     :rem 205
1400 IFNN%(I)=143THENN$(I)="CS(DF)":GOTO1
    760                              :rem 242
1410 IFNN%(I)=147THENN$(I)="D":GOTO1760
                                     :rem 202
1420 IFNN%(I)=151THENN$(I)="EF(DS)":GOTO1
    760                              :rem 245
1430 IFNN%(I)=159THENN$(I)="E":GOTO1760
                                     :rem 208
1440 IFNN%(I)=163THENN$(I)="F":GOTO1760
                                     :rem 205
1450 IFNN%(I)=167THENN$(I)="FS(GF)":GOTO1
    760                               :rem 3
1460 IFNN%(I)=175THENN$(I)="G":GOTO1760
                                     :rem 211
1470 IFNN%(I)=179THENN$(I)="AF(GS)":GOTO1
    760                               :rem 3
1480 IFNN%(I)=183THENN$(I)="A":GOTO1760
                                     :rem 206
1490 IFNN%(I)=187THENN$(I)="BF(AS)":GOTO1
    760                              :rem 255
1500 IFNN%(I)=191THENN$(I)="B":GOTO1760
                                     :rem 199
1510 IFNN%(I)=195THENN$(I)="C1":GOTO1760
                                     :rem 254
```

```
1520 IFNN%(I)=199THENN$(I)="CS1(DF1)":GOT
     O1760                              :rem 98
1530 IFNN%(I)=201THENN$(I)="D1":GOTO1760
                                        :rem 245
1540 IFNN%(I)=203THENN$(I)="EF1(DS1)":GOT
     O1760                              :rem 88
1550 IFNN%(I)=207THENN$(I)="E1":GOTO1760
                                        :rem 254
1560 IFNN%(I)=209THENN$(I)="F1":GOTO1760
                                        :rem 2
1570 IFNN%(I)=212THENN$(I)="FS1(GF1)":GOT
     O1760                              :rem 95
1580 IFNN%(I)=215THENN$(I)="G1":GOTO1760
                                        :rem 2
1590 IFNN%(I)=217THENN$(I)="AF1(GS1)":GOT
     O1760                              :rem 97
1600 IFNN%(I)=219THENN$(I)="A1":GOTO1760
                                        :rem 249
1610 IFNN%(I)=221THENN$(I)="BF1(AS1)":GOT
     O1760                              :rem 80
1620 IFNN%(I)=223THENN$(I)="B1":GOTO1760
                                        :rem 247
1630 IFNN%(I)=225THENN$(I)="C2":GOTO1760
                                        :rem 252
1640 IFNN%(I)=227THENN$(I)="CS2(DF2)":GOT
     O1760                              :rem 95
1650 IFNN%(I)=228THENN$(I)="D2":GOTO1760
                                        :rem 2
1660 IFNN%(I)=229THENN$(I)="EF2(DS2)":GOT
     O1760                              :rem 101
1670 IFNN%(I)=231THENN$(I)="E2":GOTO1760
                                        :rem 255
1680 IFNN%(I)=232THENN$(I)="F2":GOTO1760
                                        :rem 2
1690 IFNN%(I)=233THENN$(I)="FS2(GF2)":GOT
     O1760                              :rem 103
1700 IFNN%(I)=235THENN$(I)="G2":GOTO1760
                                        :rem 255
1710 IFNN%(I)=236THENN$(I)="AF2(GS2)":GOT
     O1760                              :rem 94
1720 IFNN%(I)=237THENN$(I)="A2":GOTO1760
                                        :rem 253
1730 IFNN%(I)=238THENN$(I)="BF2(AS2)":GOT
     O1760                              :rem 93
1740 IFNN%(I)=239THENN$(I)="B2":GOTO1760
                                        :rem 2
1750 IFNN%(I)=240THENN$(I)="C3"   :rem 188
1760 IFNV(I)=16THENV$(I)="W":GOTO1890
                                        :rem 159
1770 IFNV(I)=8THENV$(I)="H":GOTO1890
                                        :rem 98
1780 IFNV(I)=4THENV$(I)="Q":GOTO1890
                                        :rem 104
1790 IFNV(I)=2THENV$(I)="E":GOTO1890
                                        :rem 91
1800 IFNV(I)=1THENV$(I)="S":GOTO1890
                                        :rem 96
1810 IFNV(I)=.5THENV$(I)="T":GOTO1890
                                        :rem 148
1820 IFNV(I)=24THENV$(I)="WD":GOTO1890
                                        :rem 223
1830 IFNV(I)=12THENV$(I)="HD":GOTO1890
                                        :rem 206
1840 IFNV(I)=6THENV$(I)="QD":GOTO1890
                                        :rem 171
1850 IFNV(I)=3THENV$(I)="ED":GOTO1890
                                        :rem 157
1860 IFNV(I)=1.5THENV$(I)="SD":GOTO1890
                                        :rem 13
1870 IFNV(I)=.75THENV$(I)="TD":GOTO1890
                                        :rem 21
1880 V$(I)=STR$(NV(I))             :rem 206
1890 NEXT:RETURN                    :rem 43
1900 C=C+1:FORJ=CTOI+1STEP-1:NN%(J)=NN%(J
     -1):NV(J)=NV(J-1)             :rem 150
1910 N$(J)=N$(J-1):V$(J)=V$(J-1):NEXT:CT=
     C:C=I:PRINT"{RVS}ENTER ADDITIONAL NO
     TE"                           :rem 246
1920 GOSUB80:C=CT:RETURN            :rem 31
```

Vocab Builder

(Article on page 84.)

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.

```
5 SYS65517:IFPEEK(781)=40THENPRINT"{WHT}"
  :GOTO10                           :rem 149
6 PRINT"{BLK}"                      :rem 151
10 PRINTCHR$(14):DIMA%(100),W$(100),D$(10
   0),WR$(100):D=0                  :rem 198
30 PRINT"{CLR}{7 DOWN}{RVS}ENTER YOUR NAM
   E{OFF}":INPUTNM$:IFLEN(NM$)=0THEN30
                                    :rem 144
35 IFLEN(NM$)>13THENNM$=LEFT$(NM$,13)
                                    :rem 162
50 PRINT"{CLR}{RVS}"TAB(INT((22-LEN(NM$))
   /2)-1)NM$;"'S{OFF}":PRINTTAB(3)"{RVS}V
   OCABULARY TEST{OFF}"             :rem 91
55 N$="":A$="":B$="":DA$="":X%=0:I=0:J=0:
   FORQ=1TO100:A%(Q)=0:W$(Q)="":D$(Q)=""
                                    :rem 103
60 WR$(Q)="":NEXT:PRINTSPC(9);"{2 DOWN}ME
   NU":PRINTSPC(2);"{2 DOWN}1) ENTER NEW
   {SPACE}WORDS"                    :rem 254
80 PRINTSPC(2);"{DOWN}2) STUDY YOUR WORDS
   ":PRINTSPC(2);"{DOWN}3) TAKE A TEST"
                                    :rem 130
100 PRINTSPC(2);"{DOWN}4) END":PRINT"
    {2 DOWN}{RVS} PRESS 1-4 TO CONTINUE
    {OFF}"                          :rem 195
120 GETZ$:C$=Z$:V=VAL(Z$):IFZ$=""OR(V<1OR
    V>4)THEN120                     :rem 116
130 IFV=4THEN190                    :rem 179
140 INPUT"{2 DOWN}{RVS}D{OFF}ISK OR {RVS}
    T{OFF}APE";Z$:IFZ$<>"T"ANDZ$<>"D"THEN
    PRINT"{4 UP}":GOTO130           :rem 121
150 A$="{LEFT}{22 SPACES}":IFZ$="D"THEND=
    1                                :rem 6
160 ONVGOTO230,810,810              :rem 226
190 END                             :rem 113
230 PRINT"{CLR}{6 DOWN}ENTER # OF NEW WOR
    DS "                            :rem 14
235 INPUTX%:IFX%>100ORX%<1THEN230   :rem 11
250 FORI=1TOX%                      :rem 85
260 PRINT"{CLR}{3 DOWN}ENTER WORD"I:INPUT
    W$(I)                           :rem 17
265 IFLEN(W$(I))=0THENPRINT"{6 UP}":GOTO2
    60                              :rem 79
270 PRINT"{DOWN}ENTER DEFINITION":INPUTD$
    (I)                             :rem 174
275 IFLEN(D$(I))=0THENPRINT"{3 UP}":GOTO2
    70                              :rem 139
```

```
280 NEXT                                :rem 217
300 PRINT"{CLR}":FORI=1TOX%             :rem 239
310 PRINT" WORD"I"{LEFT}:";W$(I)        :rem 213
320 PRINT"{DOWN} DEFINITION:"D$(I):PRINT"
    {DOWN}"                             :rem 56
330 IF(I/3)=INT(I/3)ANDX%=3THEN360
                                        :rem 205
332 IF(I/3)=INT(I/3)ANDX%<>ITHENGOSUB2000
                                        :rem 203
340 NEXT                                :rem 214
360 PRINT"{DOWN}{RVS}ANY CORRECTIONS(Y/N)
    ? {OFF}"                            :rem 184
380 GETZ$:IFZ$=""OR(Z$<>"Y"ANDZ$<>"N")THE
    N380                                :rem 111
390 IFZ$="Y"THEN420                     :rem 72
400 IFZ$="N"THEN520                     :rem 54
420 I=0:INPUT"{CLR}{4 DOWN}WHICH ENTRY";I
                                        :rem 58
435 IFI=0ORI>X%THENPRINT"{4 UP}":GOTO420
                                        :rem 211
440 PRINT"{2 DOWN}WORD"I"{LEFT}:";W$(I)
                                        :rem 251
450 PRINT"{DOWN}DEFINITION:";D$(I):rem 91
470 PRINT"{2 DOWN}ENTER WORD"I:INPUTW$
                                        :rem 214
480 W$(I)=W$                            :rem 105
490 PRINT"{DOWN}ENTER DEFINITION":INPUTD$
                                        :rem 24
500 D$(I)=D$                            :rem 60
510 GOTO300                             :rem 98
520 INPUT"{2 DOWN}TODAY'S DATE: ";DA$
                                        :rem 196
540 PRINT"{CLR}{8 DOWN}{5 SPACES}{RVS}PLE
    ASE WAIT"                           :rem 134
550 PRINT"{2 DOWN}{RVS}I'M SAVING YOUR WO
    RDS {OFF}{2 DOWN}"                  :rem 182
560 N$="TEST "+DA$                      :rem 162
565 IFD=1THENN$="TEST "+DA$+",S,W"
                                        :rem 136
570 OPEN15,8,15:OPEN1+D,1+7*D,1+D,N$:INPU
    T#15,A$,B$                          :rem 67
572 IFA$="63"THENCLOSE(1+D):CLOSE15:GOTO5
    20                                  :rem 50
575 IFA$<>"00"THENPRINTB$:FORI=1TO3000:NE
    XT:CLOSE(1+D):CLOSE15:GOTO50        :rem 27
580 PRINT#(1+D),X%:FORI=1TOX%           :rem 223
590 PRINT#(1+D),W$(I):PRINT#(1+D),D$(I)
                                        :rem 137
600 NEXTI:CLOSE(1+D):CLOSE15:GOTO50
                                        :rem 173
810 PRINT"{CLR}{DOWN} {RVS}A TEST MADE JU
    ST FOR{OFF}":PRINTTAB(INT((22-LEN(NM$
    ))/2)-1)"{RVS}"NM$"{OFF}"           :rem 157
820 PRINT"{2 DOWN}ENTER THE TEST DATE ":I
    NPUTDA$                             :rem 61
870 PRINT"{CLR}{5 DOWN}{5 SPACES}{RVS}PLE
    ASE WAIT"                           :rem 89
880 PRINT"{DOWN}{2 SPACES}{RVS}LOADING TH
    E WORDS{OFF}"                       :rem 148
890 N$="TEST "+DA$                      :rem 168
895 IFD=1THENN$="TEST "+DA$+",S,R"
                                        :rem 137
900 OPEN15,8,15:OPEN1+D,1+7*D,2*D,N$:INPU
    T#15,A$,B$                          :rem 64
902 IFA$="62"THENCLOSE1+D:CLOSE15:GOTO810
                                        :rem 223
905 IFA$<>"00"THENPRINTB$:FORI=1TO3000:NE
    XT:CLOSE(1+D):CLOSE15:GOTO50        :rem 24
910 INPUT#(1+D),X%:FORI=1TOX%           :rem 223
920 INPUT#(1+D),W$(I)                   :rem 128
930 IFW$(I)=""THENGOTO950               :rem 199
940 GOTO960                             :rem 117
950 I=X%:GOTO970                        :rem 180
960 INPUT#(1+D),D$(I)                   :rem 113
970 NEXTI                               :rem 40
980 CLOSE(1+D):CLOSE15                  :rem 30
1030 FORI=1TOX%                         :rem 130
1040 A%=X%*RND(1)+1:IFI=1THENA%(I)=A%:GOT
     O1070                              :rem 94
1050 FORJ=1TOX%                         :rem 133
1060 IFA%=A%(J)THENJ=X%:NEXT:GOTO1040
                                        :rem 222
1065 NEXT                               :rem 11
1070 A%(I)=A%:NEXT                      :rem 228
1110 IFC$="2"THENGOTO1510               :rem 155
1120 PRINT"{CLR}{DOWN}{RVS} ";NM$;"'S QUI
     Z {OFF}"                           :rem 153
1130 N%=0                               :rem 165
1140 FORI=1TOX%                         :rem 132
1150 PRINT"{2 DOWN}"                    :rem 186
1160 N=A%(I)                            :rem 83
1170 PRINT"DEFINITION: ";:PRINTD$(N)
                                        :rem 70
1180 WO$="":PRINT"{DOWN}ENTER THE WORD":I
     NPUTWO$:IFWO$=""THEN1180           :rem 109
1190 IFWO$=W$(N)THENPRINT"{DOWN}CORRECT !
     ":FORZ=1TO1500:NEXT:GOTO1230       :rem 67
1200 PRINT"{DOWN}SORRY, THE WORD WAS:":PR
     INTW$(N):FORZ=1TO1500:NEXT         :rem 222
1210 N%=N%+1                            :rem 67
1220 WR$(N)=W$(N)                       :rem 136
1230 PRINT"{CLR}":NEXTI                 :rem 236
1250 PRINT"{6 DOWN}YOU GOT"N%"WRONG":PRIN
     T"OUT OF"X%:GOSUB1900:IFN%=0THEN2100
                                        :rem 222
1260 FORI=1TO3000:NEXT                  :rem 70
1270 PRINT"{CLR}{2 DOWN}THE WORD(S) THAT
     {SPACE}YOU":PRINT"GOT WRONG ARE:"
                                        :rem 83
1280 J=1:PQ=0:FORI=1TOX%                :rem 195
1290 IFWR$(I)=""THEN1310                :rem 55
1300 PQ=PQ+1:PRINT"{DOWN}";WR$(I)       :rem 93
1305 IFPQ<>5*JTHEN1310                  :rem 16
1306 J=J+1:PRINT"{DOWN} {RVS}ANY KEY TO C
     ONTINUE{OFF}":POKE198,0            :rem 119
1307 GETZ$:IFZ$=""THEN1307              :rem 235
1308 PRINT"{CLR}":NEXTI:GOTO50          :rem 202
1310 NEXTI:GOSUB2000:GOTO50             :rem 161
1510 PRINT"{CLR}{RVS} HIT THE BOOKS "
                                        :rem 129
1520 N%=0                               :rem 168
1530 FORI=1TOX%                         :rem 135
1540 PRINT"{2 DOWN}":N=A%(I)            :rem 130
1550 J%=1:J1%=1                         :rem 240
1560 PRINT"DEFINITION: ";:PRINTD$(N)
                                        :rem 73
1570 WO$="":PRINT"{DOWN}ENTER THE WORD":I
     NPUTWO$                            :rem 99
1580 IFWO$=W$(N)THENPRINT"{DOWN}CORRECT !
     ":GOTO1790                         :rem 134
1590 IFLEFT$(WO$,2)=LEFT$(W$(N),2)THENGOT
     O1610                              :rem 171
1600 GOTO1680                           :rem 207
1610 IFJ1%>1THENGOTO1750                :rem 155
1620 PRINT"{DOWN}YOU'RE CLOSE":PRINT"TRY
     {SPACE}AGAIN"                      :rem 70
1630 J1%=J1%+1                          :rem 163
1640 IFW$(N)=WR$(N)THENGOTO1670         :rem 83
1650 WR$(N)=W$(N)                       :rem 143
1670 PRINT"{2 DOWN}":GOTO1560           :rem 0
1680 IFJ%>2THENGOTO1750                 :rem 114
1690 PRINT {DOWN}SORRY, ";NM$:PRINT"YOU'R
```

```
    E NOT EVEN CLOSE":PRINT"{DOWN}TRY AG
    AIN"                              :rem 77
1691 FORZ=1TO2500:NEXT:PRINT"{CLR}":rem 1
1700 PRINT"{2 DOWN}":J%=J%+1          :rem 108
1710 IFW$(N)=WR$(N)THENGOTO1740       :rem 79
1720 WR$(N)=W$(N)                     :rem 141
1740 GOTO1560                         :rem 209
1750 PRINT"{DOWN}SORRY, THE WORD WAS:":PR
    INT"{DOWN} "W$(N):FORZ=1TO3000:NEXT:
    PRINT"{CLR}"                      :rem 216
1760 IFW$(N)=WR$(N)THENGOTO1790       :rem 89
1770 WR$(N)=W$(N)                     :rem 146
1780 N%=N%+1                          :rem 79
1790 FORTD=1TO1500:NEXT:PRINT"{CLR}":NEXT
    I:GOTO1250                        :rem 59
1900 IFN%=0THENPRINT"{2 DOWN}{RVS}GREAT J
    OB":RETURN                        :rem 213
1910 N=10-(N%/X%*10)                  :rem 223
1920 IFN<6.5THENPRINT"{DOWN}BETTER STUDY
    {SPACE}MORE":RETURN               :rem 187
1930 IFN<8THENPRINT"{DOWN}A LITTLE MORE S
    TUDY":PRINT"NEEDED":RETURN        :rem 84
1940 IFN<9THENPRINT"{DOWN}NOT BAD":RETURN
                                      :rem 131
1950 IFN<9.5THENPRINT"{DOWN}GOOD JOB":RET
    URN                               :rem 51
1960 IFN<10THENPRINT"{DOWN}VERY GOOD"RETU
    RN                                :rem 42
1970 RETURN                           :rem 177
2000 PRINT"{DOWN}{RVS} ANY KEY TO CONTINU
    E{2 SPACES}"                      :rem 15
2001 GETMT$:IFMT$=""THEN2001          :rem 105
2002 PRINT"{CLR}":RETURN              :rem 66
2100 GOSUB2000:GOTO50                 :rem 221
```

Power BASIC

(Article on page 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 1:
Hi-Res Screen Dump— 64 Version

```
3 INPUT"{CLR}WIDTH 1 OR 2";WI$        :rem 29
4 IF VAL(WI$)<1OR VAL(WI$)>2 THEN3
                                      :rem 147
5 POKE2,VAL(WI$)                      :rem 186
10 I=52224                            :rem 230
20 READ A:IF A=256 THEN 40            :rem 54
25 PRINT"{CLR}ENTERING DATA":POKE646,A
                                      :rem 183
30 POKE I,A:I=I+1:CH=CH+A:GOTO 20:rem 123
40 IF CH<>60660 THENPRINT"ERROR IN DATA S
   TATEMENTS":END                     :rem 78
50 PRINT"SYS 52224 TO START DUMP":END
                                      :rem 172
52224 DATA 173,0,221,41,3,73          :rem 87
52230 DATA 3,160,6,10,136,208         :rem 135
52236 DATA 252,141,132,3,173,24       :rem 241
52242 DATA 208,41,8,240,9,24          :rem 97
52248 DATA 169,32,109,132,3,141       :rem 249
52254 DATA 132,3,169,0,32,189         :rem 152
52260 DATA 255,169,4,170,160,255      :rem 48
52266 DATA 32,186,255,32,192,255      :rem 56
52272 DATA 162,4,32,201,255,176       :rem 247
52278 DATA 3,76,61,204,76,32          :rem 109
52284 DATA 205,169,8,32,210,255       :rem 254
52290 DATA 169,13,32,210,255,162      :rem 41
52296 DATA 0,169,1,141,198,205        :rem 206
52302 DATA 169,0,141,199,205,169      :rem 50
52308 DATA 0,141,200,205,169,199      :rem 42
52314 DATA 141,201,205,32,225,255     :rem 79
52320 DATA 208,3,76,32,205,138        :rem 196
52326 DATA 72,152,72,32,41,205        :rem 195
52332 DATA 104,168,104,170,173,205
                                      :rem 136
52338 DATA 205,45,203,205,240,12      :rem 32
52344 DATA 173,202,205,13,198,205     :rem 91
52350 DATA 141,202,205,76,143,204     :rem 81
52356 DATA 173,198,205,73,255,45      :rem 62
52362 DATA 202,205,141,202,205,14     :rem 73
52368 DATA 198,205,173,198,205,201
                                      :rem 158
52374 DATA 128,240,20,24,173,199      :rem 50
52380 DATA 205,105,1,141,199,205      :rem 38
52386 DATA 173,200,205,105,0,141      :rem 30
52392 DATA 200,205,76,93,204,173      :rem 46
52398 DATA 202,205,9,128,224,45       :rem 1
52404 DATA 144,10,173,202,205,41      :rem 26
52410 DATA 31,9,128,141,202,205       :rem 235
52416 DATA 142,207,205,166,2,142      :rem 37
52422 DATA 206,205,168,32,210,255     :rem 87
52428 DATA 152,206,206,205,208,246
                                      :rem 144
52434 DATA 174,207,205,169,1,141      :rem 43
52440 DATA 198,205,169,0,141,202      :rem 40
52446 DATA 205,56,173,199,205,233:rem 106
52452 DATA 6,141,199,205,173,200      :rem 43
52458 DATA 205,233,0,141,200,205      :rem 28
52464 DATA 206,201,205,173,201,205
                                      :rem 130
52470 DATA 201,255,240,3,76,93        :rem 201
52476 DATA 204,224,45,176,31,24       :rem 253
52482 DATA 173,199,205,105,7,141      :rem 51
52488 DATA 199,205,173,200,205,105
                                      :rem 149
52494 DATA 0,141,200,205,232,169      :rem 36
52500 DATA 199,141,201,205,169,13     :rem 89
52506 DATA 32,210,255,76,93,204       :rem 251
52512 DATA 169,13,32,210,255,32       :rem 242
52518 DATA 231,255,96,173,201,205     :rem 97
52524 DATA 41,7,141,204,205,173       :rem 242
52530 DATA 201,205,74,74,74,168       :rem 255
52536 DATA 185,146,205,133,251,185
                                      :rem 152
52542 DATA 172,205,133,252,24,165     :rem 91
52548 DATA 251,109,204,205,133,251
                                      :rem 140
52554 DATA 165,252,105,0,133,252      :rem 38
52560 DATA 24,173,132,3,101,252       :rem 237
52566 DATA 133,252,173,199,205,41:rem 104
52572 DATA 7,73,7,168,200,169         :rem 164
52578 DATA 0,56,42,136,208,252        :rem 207
52584 DATA 141,203,205,24,173,200     :rem 83
52590 DATA 205,101,252,133,252,173
                                      :rem 136
52596 DATA 199,205,41,248,168,138:rem 121
52602 DATA 72,120,162,52,134,1        :rem 186
52608 DATA 177,251,162,55,134,1       :rem 254
52614 DATA 88,168,104,170,152,45      :rem 53
```

```
52620 DATA 203,205,141,205,205,96 :rem 84
52626 DATA 0,64,128,192,0,64      :rem 102
52632 DATA 128,192,0,64,128,192    :rem 0
52638 DATA 0,64,128,192,0,64      :rem 105
52644 DATA 128,192,0,64,128,192    :rem 3
52650 DATA 0,64,0,1,2,3           :rem 88
52656 DATA 5,6,7,8,10,11          :rem 155
52662 DATA 12,13,15,16,17,18       :rem 96
52668 DATA 20,21,22,23,25,26       :rem 94
52674 DATA 27,28,30,31,256          :rem 9
```

Program 2:
Hi-Res Screen Dump— VIC Version

```
5 POKE56,14:CLR                          :rem 123
6 INPUT"{CLR}WIDTH 1 OR 2";WI$:IF VAL(WI$
  )<1OR VAL(WI$)>2THEN 6                 :rem 188
7 POKE2,VAL(WI$)                         :rem 188
10 I=3584                                :rem 187
20 READ A:IF A=256 THEN 40                :rem 54
30 POKE I,A:I=I+1:CH=CH+A:GOTO 20:rem 123
40 IF CH<>35292THENPRINT"ERROR IN DATA":S
   TOP                                   :rem 184
50 PRINT"ENTER SYS 3584 TO DUMP SCREEN":S
   TOP                                   :rem 160
3584 DATA 169,16,141,132,3,169,0  :rem 44
3591 DATA 32,189,255,169,4,170,160
                                         :rem 154
3598 DATA 255,32,186,255,32,192,255
                                         :rem 212
3605 DATA 162,4,32,201,255,176,3  :rem 34
3612 DATA 76,34,14,76,5,15,169    :rem 207
3619 DATA 8,32,210,255,169,13,32  :rem 42
3626 DATA 210,255,162,0,169,1,141 :rem 82
3633 DATA 104,15,169,0,141,105,15 :rem 78
3640 DATA 169,0,141,106,15,169,159
                                         :rem 145
3647 DATA 141,107,15,32,225,255,208
                                         :rem 190
3654 DATA 3,76,5,15,138,72,152    :rem 204
3661 DATA 72,32,14,15,104,168,104 :rem 84
3668 DATA 170,173,111,15,45,109,15
                                         :rem 142
3675 DATA 240,12,173,108,15,13,104
                                         :rem 131
3682 DATA 15,141,108,15,76,116,14 :rem 90
3689 DATA 173,104,15,73,255,45,108
                                         :rem 154
3696 DATA 15,141,108,15,14,104,15 :rem 85
3703 DATA 173,104,15,201,128,240,20
                                         :rem 171
3710 DATA 24,173,105,15,105,1,141 :rem 71
3717 DATA 105,15,173,106,15,105,0 :rem 78
3724 DATA 141,106,15,76,66,14,173 :rem 94
3731 DATA 108,15,9,128,224,22,144 :rem 88
3738 DATA 10,173,108,15,41,63,9   :rem 249
3745 DATA 128,141,108,15,142,113,15
                                         :rem 183
3752 DATA 166,2,142,112,15,168,32 :rem 87
3759 DATA 210,255,152,206,112,15,208
                                         :rem 237
3766 DATA 246,174,113,15,169,1,141
                                         :rem 146
3773 DATA 104,15,169,0,141,108,15 :rem 86
3780 DATA 56,173,105,15,233,6,141 :rem 90
3787 DATA 105,15,173,106,15,233,0 :rem 87
3794 DATA 141,106,15,206,107,15,173
                                         :rem 189
3801 DATA 107,15,201,255,240,3,76 :rem 81
3808 DATA 66,14,224,22,176,31,24  :rem 42
3815 DATA 173,105,15,105,7,141,105
                                         :rem 131
3822 DATA 15,173,106,15,105,0,141 :rem 75
3829 DATA 106,15,232,169,159,141,107
                                         :rem 249
3836 DATA 15,169,13,32,210,255,76 :rem 97
3843 DATA 66,14,169,13,32,210,255 :rem 93
3850 DATA 32,231,255,96,173,105,15
                                         :rem 143
3857 DATA 74,74,74,168,185,62,15  :rem 69
3864 DATA 133,253,185,83,15,133,254
                                         :rem 201
3871 DATA 173,105,15,41,7,73,7    :rem 201
3878 DATA 168,200,169,0,56,42,136:rem 104
3885 DATA 208,252,141,109,15,172,107
                                         :rem 245
3892 DATA 15,177,253,45,109,15,141
                                         :rem 150
3899 DATA 111,15,96,0,160,64,224   :rem 48
3906 DATA 128,32,192,96,0,160,64   :rem 48
3913 DATA 224,128,32,192,96,0,160 :rem 92
3920 DATA 64,224,128,16,16,17,17   :rem 43
3927 DATA 18,19,19,20,21,21,22    :rem 192
3934 DATA 22,23,24,24,25,26,26    :rem 193
3941 DATA 27,27,28,256             :rem 72
```

BEFORE TYPING...

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

Now there's PASCAL for the Commodore 64. And at a mini price.

Kyan PASCAL is a *full-featured implementation* of PASCAL – not a subset! With these features:

- ☐ high-performance compiler and comprehensive set of error messages for faster debugging
- ☐ speeds *up to 40 times faster* than Commodore Basic
- ☐ built-in full screen editor
- ☐ support for files, pointers, record types, sets and ordinal types used in sophisticated data structures.
- ☐ plus, a comprehensive tutorial manual illustrating the speed and power of PASCAL through sample programs.

Kyan PASCAL lets you develop programs on your Commodore 64 that are transportable to most mainframes. So it's perfect for students and working at home.

Try it for 15 days.

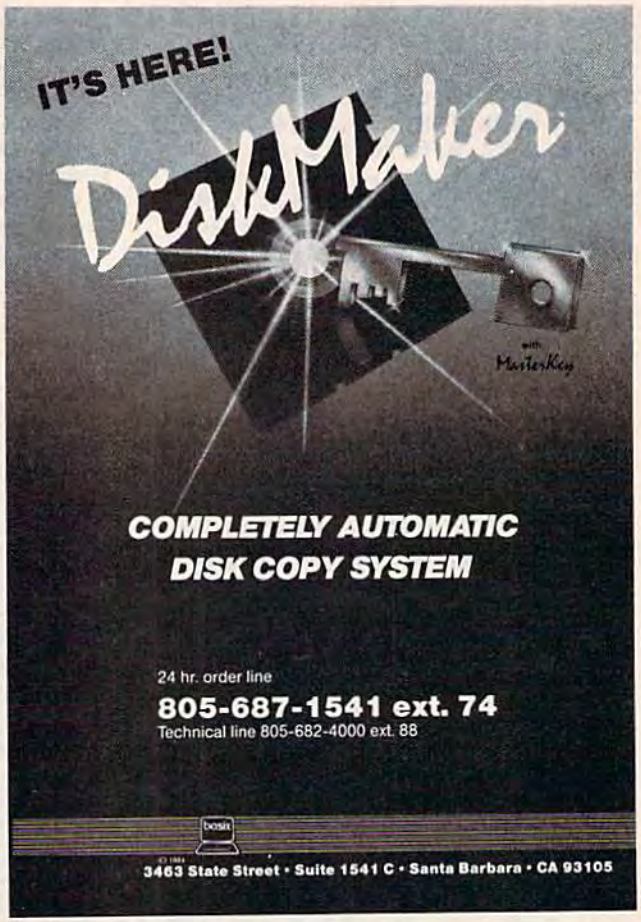
Order Kyan PASCAL, and if you're not completely satisfied, return the diskette and manual within 15 days for a full refund.

Kyan PASCAL with complete tutorial manual $49.95
(Add $4.50 per copy for postage and handling. $9.50 outside North America. California residents add 6.5% sales tax.)

To order, call 415/775-2923.
(VISA and MasterCard accepted.)

Or send check or money order to: ***Kyan Software, Suite 183, 1850 Union Street, San Francisco, CA 94123.***
Sorry, no C.O.D. orders accepted.

Kyan



DON'T

SPEND ANOTHER DIME ON ANY OTHER GAME UNTIL YOU EXPERIENCE
-the Challenge-
-the Suspense-
-the Power of-

SALVAGE

Graphic Adventure Game Series

Salvage I

Explore the sunken S.S. Commodore. Can you defy the curse and recover the valuable treasures below?

Salvage II

Unlock the secrets of an evil Alien Race from Outer Space and return with just two items-a valuable gem . . . and your life!

Salvage III

Conquer the Ultimate Adventure Challenge! Expose the traiterous spy who sabotaged the ship and murdered its crew for millions in GOLD! Computer Generated Graphics in every program shows your salvage boat, the sunken ship, diver location and more.

-for Commodore 64 and Vic $27.95 DISK $24.95 CASSETTE Order all three for only $65.00 and we will send you our $27.95 Action Arcade BATTLESHIP game absolutely FREE! - send check or money order to

DP *Dynastar Productions*

P.O. Box 508, Toano, VA. 23168 **804/564-0087**

Dealer Inquiries Invited

Better Than Original Equipment

This CP Power Supply for the Commodore 64 and VIC-20 is definitely better and more reliable than the one that originally came with your computer. It works dependably, year-in and year-out. Unlike the others, this CP Power Supply is serviceable, NOT disposable.

Computer Place (213) 325-4754
23914 Crenshaw Blvd. Torrance, CA 90505
Dealer Inquiries Welcome

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

UNIQUE HARDWARE
For Your Commodore or Vic

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

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

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

1. **The Interference Stopper...** For Vic-20 and Commodore 64. A new kit that installs in minutes with two simple solder connections. Best results when combined with #2, 3, or 4 below. Absolutely stops 90% of the RF interference on your screen.
$15.95

2. **The Color Sharpener...** Use if your "old 64" is hooked up to a TV. Just plug into the monitor plug, and the color and contrast immediately improve. Dramatically. Crisp letters. Great graphics.
$18.95

3. **The NEW Color Sharpener CABLE...** Use if your "old 64" is hooked up to a monitor. A new 2 prong cable, with the Color Sharpener built in. All the benefits of #2, on your monitor.
$24.95

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

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

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

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

Why Blank "Cheat" Sheets?

Because They're Better Blank

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

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

12 for **$15.95**

Is Your Commodore Disk Drive Hot and Bothered?

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

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

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

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

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

Order Today!

Rush me the following:

Qty.	Item	Amount
____	Commodore 64 Interference Stopper @ $15.95	$ ____
____	Vic-20 Interference Stopper @ $15.95	$ ____
____	Color Sharpener @ $18.95	$ ____
____	NEW Color Sharpener Cable @ $24.95	$ ____
____	The Monitor Improver *$24.95	$ ____
____	The Reset Switch @ $9.95	$ ____
Dust Covers		
____	Commodore Dust Cover @ $9.95	$ ____
____	Vic-20 Dust Cover @ $9.95	$ ____
____	1541 Disk Dust Cover @ $8.95	$ ____
____	Dataset Dust Cover @ $7.95	$ ____
Cheat Sheets		
____	Sets of 12 Keyboard Cheat Sheets @ $15.95	$ ____
____	2 Packs (24 Sheets) @ $24.95	$ ____
Muffin Fans		
____	Muffin Fans @ $54.95	$ ____
____	Muffin Fans with Surge Protector @ $79.95	$ ____
	Shipping & Handling	$ 2.00
	Sub Total	$ ____
	5% State Tax (Wisconsin Residents only)	$ ____
	TOTAL	$ ____

☐ Check or Money Order enclosed
☐ Charge to my VISA or MasterCard

VISA # _____
MasterCard # _____
Inner Bank # _____
Expiration Date _____

Signature _____

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

Bytes & Pieces, Inc.
550 N. 68th Street,
Wauwatosa, WI 53213
(414) 257-1214
Dealer Inquiries Invited

"The Rabbit"
for your VIC 20 or CBM 64

$39.95

If you own a VIC 20 or a CBM 64 and have been concerned about the high cost of a disk to store your programs on . . . worry yourself no longer. Now there's the RABBIT. The RABBIT comes in a cartridge, and at a much, much lower price than the average disk. And speed . . . this is one fast RABBIT. With the RABBIT you can load and store on your CBM datasette an 8K program in almost 30 seconds, compared to the current 3 minutes of a VIC 20 or CBM 64, almost as fast as the 1541 disk drive.
The RABBIT is easy to install, allows one to Append Basic Programs, works with or without Expansion Memory, and provides two data file modes. The RABBIT is not only fast but reliable. (The Rabbit for the VIC 20 contains an expansion connector so you can simultaneously use your memory board, etc.)

MAE
NOW THE BEST FOR LESS!

Please for your own protection consider the MAE first before you buy that other assembler. We've had numerous customers who wasted their money on some cheaper off brand assembler tell us how much better the MAE is.
The most powerful Macro Assembler/Editor available for the Commodore 64 and other CBM/PET computers, and also for the ATARI 800/XL and Apple II/IIE.
MAE includes an Assembler, Editor, Word Processor, Relocating Loader, and more all for just $59.95.
We could go on and describe the MAE but we thought you would like to read our customers' comments. The following are actual unedited comments from correspondence about the MAE:

"Excellent Development Package."
"Compares to DEC and INTEL."
"My Compliments to Carl Moser and EHS."
"It is a superb program."

TELSTAR 64 — "A Star is Born"
Sophisticated Terminal Communications Cartridge for the 64.

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

Don't settle for less than the best!

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

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

Machine Language Monitor Cartridge
for the CBM 64

More than 20 commands allow you to access the CBM 64's Microprocessors Registers and Memory Contents. Commands include assemble, disassemble, registers, memory, transfer, compare, plus many more.
Someday every CBM 64 owner will need a monitor such as this.
Cartridge and Manual — **$24.95**

Eastern House

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

VISA
MasterCard

NEW

COMPUTER ROAD ATLAS

TAKE TRIPS WITH COMPUTER LISTINGS SHOWING THE BEST ROUTE

ROADSEARCH MAP

ROADSEARCH IS A COMPUTERIZED ROAD ATLAS that helps you plan trips. Enter the departing city and the destination city. **ROADSEARCH** computes the shortest route or other routes more suitable to your needs. Printout includes detailed driving directions, mileage, time, fuel and more.

ROADSEARCH contains a roadmap of 406 cities/road junctions and 70,000 road miles. **ROADSEARCH-PLUS** (extra cost) also contains a **ROADMAP DEVELOPMENT SYSTEM** which lets you customize your roadmap. Add local roads, favorite shortcuts, new destinations, etc. **ROADSEARCH-PLUS** lets you add up to 50 towns/road junctions anywhere in North America.

EASY TO USE. Back-up copies allowed. Specify Apple II/IIe or Commodore-64. All Columbia Software carries a **15 DAY MONEYBACK GUARANTEE.**

STOP USING THE "OLD FASHIONED" METHOD of tracing routes on a map. It is tedious, time consuming, and inaccurate. Use your computer to plan your next route. The printed output is an excellent companion on any trip.

ORDER ROADSEARCH OR ROADSEARCH-PLUS from your dealer or directly from Columbia Software. **ROADSEARCH-PLUS** is $74.95 and **ROADSEARCH** is only $34.95. Add $1.50 for shipping and handling. MD residents add 5% state tax.

MAIL COUPON OR CALL TODAY

Columbia Software (301) 997-3100
Box 2235W, Columbia, MD 21045

Please send me ( ) **ROADSEARCH-PLUS** for $74.95 ( ) **ROADSEARCH** for $34.95 for ( ) Apple II/IIe ( ) C-64 disk

Add $1.50 shipping. If I am not satisfied, I may return it for a full refund.

( ) check ( ) Master Charge ( ) VISA

Card Number Expiration Date

NAME

ADDRESS

CITY STATE ZIP

REAL SPEECH SYNTHESIZER

Only
$89.95
Complete

The best speech synthesizer available because it talks in **your own voice!** Speaks in any language—even singing! You can even add special effects. Easily programs in BASIC using new commands. Children find it utterly fascinating. Imagine putting your own voice (or someone else's) into education, business, and game programs. These are just a few of the many reasons why the Voice Master system was selected at CES as one of the most innovative consumer electronic products of 1984!

Available for most personal computers. The Commodore 64 version will reproduce speech without any extra hardware! Software license available at no extra cost. Word recognition software available separately.

SEE YOUR DEALER NOW FOR A VOICE MASTER DEMONSTRATION

Or call us at (503) 342-1271 for a **telephone demonstration.** Please include $4.00 shipping/handling when ordering from the factory. Visa, MC, MO, or check. Specify disk or tape. Call or write for further information **and receive a FREE audio demo tape!**

COVOX INC.
675-D Conger St., Eugene, OR 97402
Telex 706017

Lyco Computer Marketing & Consultants

SAVE ON THESE IN-STOCK PRINTERS

MANNESMANN
SPIRIT 80 ..... $299.00
MTL-160L $559.00
MTL-180L $775.00

NEC
NEC8023 ..... $369.00
NEC8025 ..... $699.00

CITOH
GORILLA GX-100 ..... $169.00
GORILLA SERIAL $199.00
PROWRITER II $575.00
8600 ..... $899.00
SHEET FEEDER $425.00
620 ..... $929.00
630 ..... $1699.00
8510 SP $499.00
8510 SCP $559.00
A10 LETTER QUAL $499.00
PROWRITER 8510 ..... $335.00
PRINTMASTER $1199.00
STARWRITER $929.00

PANASONIC
1090 ..... $279.00
1091 ..... $309.00
1092 ..... $459.00

BLUE CHIP
M12010 ..... $279.00
M12010 C-64 ..... $279.00
D4015 ..... $1399.00

LEGEND
880 ..... $259.00
1000 ..... $279.00
1200 ..... CALL
1500 ..... CALL
1018 ..... CALL

EPSON
RX-80 ..... $269.00
RX-80FT $299.00
FX-80 ..... $439.00
FX-100 ..... $639.00

OKIDATA
80 ..... $199.00
82A $299.00
83A $569.00
84 ..... $959.00
92 ..... $409.00

RITEMAN
INFO RUNNER ..... CALL

JUKI
6100 ..... CALL

STAR
MICRONICS
GEMINI 15X $379.00
DELTA 10 ..... $449.00
DELTA 15 ..... $525.00
RADIX 10 ..... $575.00
RADIX 15 ..... $675.00
POWERTYPE ..... $CALL
SWEET P (Model 100) ... $549.00
STX80 ..... $149.00

GEMINI 10X
..$259.00

PRINTING PAPER
3000 SHEETS FANFOLD $42.75
1000 SHEETS FANFOLD $19.75
1000 SHEETS LETTER ... $21.95
200 SHEETS LETTER $8.99
150 RAG STATIONERY ... $10.99
MAILING LABELS (1 in) ... $9.75
14x11 1000 FANFOLD ... $24.75

SUB LOGIC-64
Flight Simulator II $32.75
Night Mission Pinball $22.75

TIMEWORKS-64
Electronic Checkbook $18.75
Star Battle $18.75
Data Manager $18.75
Wall St. Manager $18.75

EPYX-64
Jumpman $27.75
Jumpman Jr. $27.75
Pitstop $27.75
Gateway to Apshai $27.75
Temple of Apshai $27.75

PROFESSIONAL SOFTWARE
Word Pro 64/Spell $62.50

PERSONAL PERIPHERALS-64
Super Sketch Graphics Tablet $39.95
Joystick $7.75

RESTON SOFTWARE-64
Miner 2049er $27.95

KOALA-64
Koala Pad $69.95
Geometric Design $22.50

STRATEGIC SIMULATIONS-64
Knights of the Desert $26.75
Eagles $26.75
Computer Baseball $26.75
Ringside Seat $26.75

EASTERN HOUSE-64
Rabbit Rom $34.75
TelStar 64 ..... $39.95
Assembler Rom $19.95

SCARBOROUGH-64
Mastertype $27.75
Songwriter $27.75
PHI Beta Filer $32.75

ADVENTURE INTERNATIONAL-64
Arex $25.00
Sea Dragon $25.00
Stratos $25.00

BRODERBURD-64
Bank St. Writer $39.95
Choplifter $27.95
Seafox $22.95
Spare Change $22.95
Lode Runner $22.95

CARDCO-64
C/?G Printer Interface ... $69.75
C/?B Printer Interface ... $39.75
CK/1 Numeric Keypad ... $34.75
DC/1 Cassette Recorder $39.75
CB/5 5-Slot Expansion .. $54.00
CR/1 Light Pen $29.75
CE/1 Cassette Interface . $29.75
C/02 Write Now $34.75
D/01 Mail List $26.75
D/04 Spell Now $26.75

MICROPROSE-64
Solo Flight $21.75

ALIEN GROUP-64
Voice Box $99.00

PARKER-64
Q Bert $29.95
Frogger $29.95

ATARISOFT-64
Pacman $32.75
Donkey Kong $32.75
Dig Dug $32.75
Defender $32.75
Centipede $32.75

HESWARE-64
Microsoft Multiplan $59.95
Omni Writer $39.95
Turtle Toyland Jr $21.95
Time & Money Mgr $39.95
Paint Brush $12.95
Hes Games $24.95
The Factory $21.95
Minn. Fats Pool $19.95

INFOCOM-64
Enchanter $32.75
Infidel $32.75
Planetfall $32.75
Witness $32.75
Sorcerer $32.75

SPINNAKER-64
Alphabet Zoo $23.75
Kindercomp $23.75
Kids On Keys $23.75
Facemaker $23.75
Delta Drawing $26.75
Fraction Fever $23.75
Cocmic Life $23.75
Up For Grabs $26.75

SYNAPSE-64
Blue Max $22.75
Shamus $22.75
Shamus II $22.75
Zeppelin $22.75
Zaxxon $22.75

CONTINENTAL-64
Home Accountant $44.75
Tax Advantage $35.75
C-64 Users Encyclopedia $12.50

DISKETTES

ELEPHANT
5¼" SSSD $16.75
5¼" SSDD $19.95
5¼" DSDD $25.95

MAXELL
5¼" MD-1 ..... $24.75
5¼" MD-2 ..... $34.75

CERTRON CASSETTES
CC-10 (12) $15.99
CC-20 (12) $17.99

MONITORS

GORILLA
12" GREEN $82.00
12" AMBER $89.00

TAXAN
100 GREEN $115.00
105 AMBER $125.00
210 RGB COMPOSITE ... $269.00
400 RGB MEDRES $295.00
410 RGB HI RES $349.00
420 RGB HI RES (IBM) $475.00
121 GREEN $145.00
122 AMBER $149.00

SAKATA
SC100 COLOR ... $229.00
SG1000 GREEN $99.00
SA AMBER $109.00

ZENITH
ZVM 122A AMBER $95.00
ZVM 123G GREEN $85.00
ZUM 124IBM-PC $129.00
ZUM COLOR $275.00
ZUM 133 RGB $389.00
ZUM 135 RGB $449.00
ZUM 136 ..... $589.00

AMDEK
VIDEO 300 ..... $139.00
VIDEO 300A $149.00
COLOR 1+ $269.00
COLOR 11+ $399.00

NEC
JB1260 GREEN $99.00
JB1201 GREEN $145.00
JB1205 ..... AMBER $145.00
JC1215 COLOR $269.00
JC1216 ..... $399.00

MasterCard VISA

TO ORDER

CALL TOLL FREE
800-233-8760
Customer Service 1-717-327-1825 Jersey Shore PA 17740

or send order to
Lyco Computer
P O Box 5088

POLICY
In-stock items shipped within 24 hours of order. Personal checks require four weeks clearance before shipping. No deposit on C.O.D. orders. Free shipping on prepaid cash orders within the continental U.S. PA residents add sales tax. All products subject to availability and price change. Advertised prices show 4% discount offered for cash; add 4% for Master Card or Visa. DEALER INQUIRIES INVITED

COMMODORE 64 SOFTWARE

Access
Beach-Head ..... (D)23.95
Neutral Zone ..... (D)23.95
Raid over Moscow ..... (D)26.95
Master Composer ..... (D)26.95
Atari Soft All Titles ..... Call
Logo ..... (D)59.95
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
Nevada Cobol ..... (D)49.95
Nevada Fortran ..... (D)49.95

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
Magic Desk ..... (D)54.95
Zork (1-3) ..... (D)30.95
Suspended ..... (D)37.95
Starcross ..... (D)30.95
Deadline ..... (D)37.95
Voice Synthesis ..... (C)59.95
Public Domain All Titles ..... (D)8.95
Accounting Series (ea) ..... (D)49.95

Continental
Home accountant ..... (D)50.75
Tax Advantage ..... (D)40.60
FCM ..... (D)76.50

Creative Software
Household Fin ..... (D,T)19.95
Home Inv. ..... (D,T)19.95
Loan Analyzer ..... (D,T)19.95

Cymbal
English/Spanish ..... (D)34.95
Science ..... (D)34.95
History/Geography ..... (D)34.95
Music Theory ..... (D)34.95
Mathematics ..... (D)34.95
Entertainment Tonight
Trivia ..... (D)25.95

Epyx
Pitstop ..... (C)27.00
Temple of Apshai ..... (D)27.00
Jumpman ..... (D,T)27.00
Gateway to Apshai ..... (D)27.00
Jumpman Jr. ..... (C)27.00
Oil Baron ..... (D)39.95
Impossible Mission ..... (D)25.95
Greatest Baseball ..... (D)25.95
Summer Games ..... (D)29.95
Silicone Warrior ..... (D)25.95

Hes Software
Hes Mon 64 ..... (C)27.00
Turtle Graphics II ..... (C)40.60
Time Money Mgr. ..... (D)51.00
Omniwriter ..... (D)24.20
Multiplan ..... (D)59.95
Hes Writer 64 ..... (C)30.45
Gridrunner ..... (C)20.95
64 Forth ..... (C)49.95
HES Games '84 ..... (D)27.95
Minnesota Fats Pool ..... (C)20.45
HES Modem I ..... (D)46.95

Microprose
Hellcat Ace ..... (D)23.95
Solo Flight ..... (D)26.95
Nato Commander ..... (D)26.95
F-15 Eagle ..... (D)36.95

subLogic
Flight SimulatorII ..... (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 ..... (D,C)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 ..... (D,C)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 ..... (D)31.95

Synapse
Ft. Apocalypse ..... (D,C)23.70
Shamus ..... (D,C)23.70
Blue Max ..... (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
RFD 1985 ..... (D)29.95
Germany 1985 ..... (D)34.95
Golf ..... (D)29.95
Broadsides ..... (D)31.95

Electronic Arts
One on One ..... (D)37.00
Music Construction ..... (D)37.00
Sky Fox ..... (D)37.00

Miscellaneous
Doodle ..... (D)31.95
Word Pro 3+ ..... (D)76.95
Trivia Mania ..... (D)33.95
Paper Clip /w Spell Pack ... (D)89.95
Orical ..... (D)79.95
Musicalc ..... (D)41.95
Mr. Robot ..... (D)26.95

64

Hardware
Commodore 64
Executive 64
1541 Disk Drive
1530 Datasette
MPS 801 Printer
1650 Automodem
1702 Color Monitor
1600 Modem
64 Prog. Ref. Guide
BASF 21.95
Soft Box (10) 23.80

Joysticks
Wico
Command Ctrl ... 19.50
Wico "Boss" 13.50
Wico Red Ball 20.75
Wico Power Grip . 20.75

Cardo
Card+G 68.25
Card B 37.85
Cass. Interface ... 33.25
5 Slot Expansion. . 60.25
Numeric Keypad . 30.25
GEMINI Printers
O'SULLIVAN Furniture
MAXTRON
Datasettes

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 call on shipping with a minimum of $10.00. All prices are subject to change. Credit and exchange only please. For return authorization call (214) 759-8937.

ATARI, APPLE, IBM, SOFTWARE LISTS AVAILABLE
For further information and inquires call 214-759-8937
FOR ORDERS ONLY CALL TOLL FREE
Outside Texas **1-800-527-8698**
Inside Texas **1-800-442-8717**

SoftWare Warehouse Outlet

P.O. BOX 2511 LONGVIEW, TX. 75606

Only 10% of Software we carry is listed on this ad. Call if there is a title you need for all computers. We are here to serve you!

DI-SECTOR Version 2.0

*EVERYTHING YOU EVER WANTED IN A DISK UTILITY . . . AND MORE!! FOR THE COMMODORE 64**

- ★ Fully automatic back-up of almost any protected disk.
- ★ Copy files (PRG, SEQ, RND) with full screen editing.
- ★ Three minute back-up of standard disks (even many protected disks).
- ★ Format a disk in ten seconds.
- ★ View and alter sector headers.
- ★ Remove errors from any track/sector.
- ★ Edit sectors in HEX, ASCII — even assembler.
- ★ Create errors on any track/sector (20, 21, 22, 23, 27, 29) instantly.
- ★ Drive/64 MON, even lets you write programs inside your 1541.
- ★ All features are fully documented and easy to use.
- ★ None of our copy routines ever makes the drive head "kick."

Yes, you get all of this on one disk for this low price!

$39.95

. . . WRITE OR PHONE . . .

STARPOINT SOFTWARE

Star Route 10 Gazelle, CA 96034 (916) 435-2371

VISA or MASTERCARD

add $3 for C.O.D.

*Commodore 64 is a registered trademark of Commodore Business Machines

ROBOTS

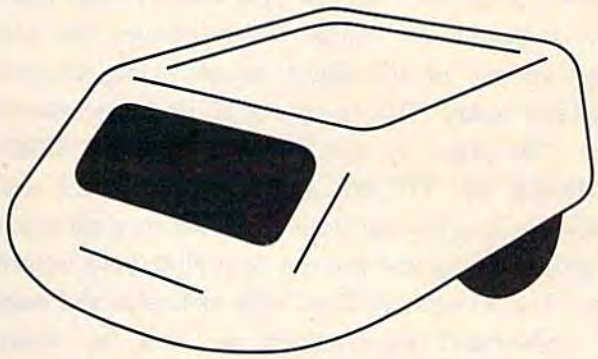
Let NOMAD introduce you to the AGE OF PERSONAL ROBOTS
THE NOMAD ROBOT! AFFORDABLE! FUN! EDUCATIONAL!

IS NOMAD INTELLIGENT?
YES! Your C-64 or VIC is NOMAD's brain.

CAN NOMAD MOVE ABOUT?
YES! NOMAD accurately travels forward, reverse, and pivots left and right using industrial type stepper motors.

CAN NOMAD SEE?
YES! NOMAD has ultrasonic sight, ranging and motion detection capabilities.

DOES NOMAD COME WITH SOFTWARE?
YES! NOMAD comes complete with ready to run software including a friendly robot language for creating your own programs.



$179.95

HOME CONTROL

HOME SECURITY AND ENERGY MANAGEMENT PRODUCTS

VIController. Wireless home control using BSR or Leviton receiver modules. Software included. $69.95

COMclock/AB Autoloads and runs program at power-up. Battery backed up clock. (C-64 only) $69.95

COMsense. Provides 4 open/close and 2 analog inputs. Great for security $69.95

SUPER SCHEDULE PLUS Control software utilizing above three products. (C-64 only) $19.95

SPECIAL PACKAGE PRICE **$199.95**

SPEECH

COMVOICE! THE EASY TO USE SPEECH SYNTHESIZER FOR THE C-64

SPEECH IS AS EASY AS 1-2-3!

1) Plug COMvoice in.
2) Turn computer on.
3) Type SPEAK "HELLO, HOW ARE YOU"

AS EASY TO USE AS A PRINT STATEMENT!

$99.95

W/EXTERNAL SPEAKER $139.95

EDUCATIONAL DEMO & AARDVARK ADVENTURE SOFTWARE AVAILABLE

GENESIS COMPUTER CORPORATION • P.O. BOX 152 • HELLERTOWN, PA 18055

DEALER INQUIRIES INVITED • **(215) 861-0850** • MC & VISA Accepted

ADVERTISEMENT

ATTENTION ALL COMMODORE 64, VIC 20, and PET OWNERS!

A complete self-tutoring BASIC programming course is now available. This course starts with turning your computer on, to programming just about anything you want! This course is currently used in both High School and Adult Evening Education classes and has also formed the basis of teacher literacy programs. Written by a teacher, who after having taught the course several times, has put together one of the finest programming courses available today. This complete 13 lesson course of over 220 pages is now available for the COMMODORE 64, VIC 20, and PET computers and takes you step by step through a discovery approach to programming and you can do it all in your leisure time! The lessons are filled with examples and easy to understand explanations as well as many programs for you to make up. At the end of each lesson is a test of the information presented. Furthermore, ALL answers are supplied to all the questions and programs, including the answers to the tests. Follow this course step by step, lesson by lesson, and turn yourself into a real programmer! You won't be disappointed!

We will send this COMPLETE course to you at once for just $19.95 plus $2.00 for shipping and handling (U.S. residents, please pay in U.S. funds). If you do not live in the U.S. or Canada, please add $5.00 for shipping and handling (and pay in U.S. funds). If you are not COMPLETELY satisfied, then simply return the course within 10 days of receipt for a FULL refund.

Fill in the coupon or send a facsimile.

NAME: _____

ADDRESS: _____

CITY: _____

PROV./STATE: _____

POSTAL/ZIP CODE: _____

CHECK DESIRED COURSE: COMMODORE 64 ☐ VIC ☐ PET ☐

Send Cheque or Money Order to:
Brantford Educational Services
68 Winding Way,
Brantford, Ontario,
Canada, N3R 3S3

Complete course:	$19.95
Postage and hand:	$2.00
Total:	$21.95

Software Discounters of America

For Orders Only 1-800-225-SOFT
Inquiries and PA. 412-361-5291

ACCESS
Beach Head (T or D) .$23
Master Composer (D) $26
Neutral Zone (T or D) $23
Raid Over Moscow (T or D) $26
ACTIVISION
Beamrider (D) $25
Pitfall (D) $25
ARTWORX
Bridge 4.0 (T or D) ...$16
Monkeymath (T or D) $18
Strip Poker (D) $21
BATTERIES INCLUDED
Home Organizer Series:
Audio-Video Catalog (D) $21
Checkbook (D) $21
Home Inventory (D)$21
Recipes (D) $21
Stamps (D) $21
Paperclip (D) $59
The Consultant (D) ..$67
BRODERBUND
Bank St. Writer (D) ..$43
Castles Dr. Creep (D) $21
Championship Loderunner (D) ...$23
Choplifter (R) $25
Drol (D) $23
Loderunner (D) $23
Mask of the Sun (D) ..$25
Operation Whirlwind (D) $25
CBS
Big Bird's Spc. Del (D) $23
Decimals (T or D)....$19
Ducks Ahoy (R) $23
Fractions (T or D) ...$19
Match Wits (D).....$19
Math Mileage (D)....$19
Mastering SAT (D)...$99
Mult/Div (T or D).....$16
Murder by Dozen (D) .$23
Peanut Butter Panic (D) $23
CONTINENTAL
FCM (D) $33
Home Accountant (D)$47
DATASOFT
Bruce Lee (D).....$23
Dallas Quest (D) $23
Letter Wizard (D) $33
Pooyan (T/D) $19
DESIGNWARE
Creature Creator (D) .$21
Math Maze (D) $25
Spellicopter (D) $25
Trap-A-Zoid (D) $25
EPYX
Dragonriders Pern (D)$25
FAX (D).....$19
Gateway Apshai (R) .$25
Impossible Mission (D) $23
Jumpman (T or D) ...$25
9 to 5 Typing (D).....$25
Oil Barons (D) $39
Pitstop (R) $25
Puzzlepanic (D) $23
Summer Games (D)..$25
Temple Apshai (T or D) $25
Worlds Greatest Baseball (D).....$23
FISHER PRICE
Alpha Build (R) $19
Dance Fantasy (R) ..$19
Hop Along Counting (R) $19
Linking Logic (R)....$19
Memory Manor (R) ..$19
Number Tumbler (R) .$19
Sea Speller (R).....$19
Up & Add'em (R) $19
FIRST STAR
Astro Chase (T or D) .$19
Bristles (T or D) $19
Flip Flop (T or D) $19
HBJ
Computer SAT (D)...$59
HES
HES Modem I $49
HES Modem II.....call
HES Games 84 .....$23
Multiplan (D) $65
Paint Brush (R) $18
Synthesound (D) $17
Time/Money Mgr (D) .$19
INFOCOM
Deadline (D).....$25
Enchanter (D) $33
Infidel (D) $33
Planetfall (D) $33
Sea Stalker (D).....$25
Sorcerer (D) $33
Suspended (D).....$25
Witness (D) $33
Zork I, II, III (D) $25
KOALA
Touch Tablet w/Painter (D) $65
Touch Tablet w/Painter (R) $75
Muppet Learning Keyboard.....call
Paint-A-Rhyme (D) ..$21
Spider Eater (D).....$21
MICROLAB
Death In Caribbean (D) $23
Dino Eggs (D).....$23
English SAT I (D) $19
Math SAT I (D) $19
MICROPROSE
Hellcat Ace (D) $19
NATO Commander (D) $23
Solo Flight (T or D) ..$23
Spitfire Ace (D) $19
PARKER BROS
Frogger (R).....$29
Popeye (R) $33
Q-Bert (R).....$33
PRECISION SOFTWARE
Superbase 64 (D)....$67
RESTON
Miner 2049er (R) $25
SCARBOROUGH
Mastertype (D or R) ..$25
Songwriter (D) $25
SIERRA ON-LINE
Championship Boxing (D) $19
Homeword (D) $43
Oil's Well (D) $19
Quest for Tires (d) ...$23
Ultima II (D) $39
Ulysses (D).....$23
Wizard & Princess (D)$21
Wiz Type (D).....$23
SPINNAKER
Adventure Creator (R)$25
Aegean Voyage (R) ..$25
Aerobics (D).....$29
Alphabet Zoo (R) $21
Cosmic Life (R) $21
Delta Drawing (R) ...$25
Facemaker (R).....$21
Fraction Fever (R) ...$21
Grandma's House (D)$21
Hey Diddle (D) $19
Kids on Keys (R) $21
Kidwriter (D) $21
Kindercomp (R).....$18
Most Amazing Thing (D) $25
Story Machine (R) ...$25
Trains (D).....$25
SSI
Battle Normandy (D) .$25
Combat Leader (D) ..$25
Computer Baseball (D).....$25
Cosmic Balance (D) .$25
Eagles (D) $25
50 Mission Crush (D) $25
Fortress (D) $23
Germany 1985 (D) ...$39
Knights of Desert (D) $25
Pro Tour Golf (D) $25
RDF 1985 (D) $23
Ringside Seat (D) ...$25
Tigers In Snow (D)...$25
SUBLOGIC
Flight Simulator II (D)$35
Night Mission Pinball (D) $21
SYNAPSE
Blue Max (T or D)....$21
Ft. Apocalypse (T or D) $21
Necromancer (T or D) $21
Pharoah's Curse (T or D) $21
Sentinel (T or D).....$21
Slamball (T or D) $21
Zaxxon (T or D) $25
Zepellin (T or D).....$21
TIMEWORKS
Data Manager (T or D)$17
Data Manager 2 (D) ..$33
Money Manager (T or D) $17
Spellbound (T or D) ..$17
Word Writer (D) $33
TRILLIUM
Amazon (D) $27
Dragonworld (D) $27
Fahrenheit 451 (D) ..$27
Rendevous with Rama (D) $27
Shadow Keep (D)....$27
TRONIX
Chaterbee (D) $25
Pokersam (D).....$19
S.A.M. (D).....$39
Suicide Strike (D) ...$23
WAVEFORM
Musicalc 1 (D) $35
Musicalc 2 (D) $23
Musicalc 3 (D) $23
ACCESSORIES
Alien Group Voice Box.....call
BASF SS, DD ...$17 Box
Cardco Accessories..call
Commodore Dust Covers $6
Compuserve Starter Kit.....$25
Disk Drive Cleaner ...$9
Disk Case (Holds 50) $12
Monitor Cable $7
Sakata 13" Color Monitor $239
Surge Protector w/6 Outlets.....$39
Verbatim Disks .....call
Wico Boss $13
Wico Bat Handle....$19
Wico Trackball $29

P.O. Box 278 — Dept. CG • Wildwood, PA 15091

Ordering and Terms: Orders with cashier check or money order shipped immediately Personal/company checks, allow 3 weeks clearance. No C.O.D.'s. VISA/MASTERCARD accpeted with no additional charge for orders shipped to continental U.S.A. **Shipping: Continental U.S.A.** — Orders under $100 add $3; free shipping on orders over $100. PA residents add 6% sales tax. **AK, HI, FPO-APO** — add $5 on all orders. **INTERNATIONAL** — add $10 or 15% of order whichever is greatest. Defective merchandise will be replaced with same merchandise — NO CREDITS! Return must have authorization number (412) 361-5291. Prices subject to change without notice.

PC GALLERY

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

Toll FREE (800) 992-3300

For Information Call (609)596-1944

Call us for reliable service, experience and affordable prices!

$Call

CMD 64 ..... $Call

1541 $Call

1520 Color Plotter $109
MPS 801 Printer $205
1530 Datasette $64
1650 Auto Modem $86
1702 Monitor ..... $CALL
Int'l Soccer $21
1011 RS 232
Terminal Interface $42
1211 Super Expander $39
EZ Back up $25
Magic Voice Speech $49.95

Repeat of a Sellout
1526 PRINTER
$269

Special of the Month
ELEPHANT DISK
SS/SD $16.00 SS/DD $17.00

WORD PROCESSING

Bank Street Writer $45
Paper Clip $69
Easy Script $45
Word Pro 64 ..... $58
EZ Spell $19

PROGRAMMING SERIES

Assembler 64 ..... $36
Logo $52
Pilot $39
Simon Basic $44
Screen Editor $19
Nevada Fortran $39
Nevada Cobol $39

ACCOUNTING

Home Accountant $45
Tax Advantage $45
General Ledger $35
A/R, A/P, Payroll $35

DATA BASES

Code Writer $64
Mirage Database Mgr. $79
Super Base 64 ..... $77

Special of the Month
The Consultant
Formerly Delphi Oracle
$65.00

Special of the Month
Gemini 10X
$266

PRINTERS

EPSON ..... $ CALL
OKIDATA ..... $CALL
Silver Reed ..... $Call

STAR

Gemini 10X $266
Gemini 15X $366
Powertype $395

SURGE PROTECTOR

4 Outlet $37.95

MSD DISK DRIVE

SD1 Disk Drive $335
SD2 Dual Disk Drive ..... $Call

CARDCO

*G Printer Interface $64
*Numeric Keypad $29.50
Light Pen $31
5 Slot Expansion $49
Datasette $47
TYMAC Connection $75

SPREAD SHEET

Multiplan $68
Calc Results $34
Practicalc $38
Koala Pad $65

Special of the Month
SUBLOGIC
Flight Simulator
$34.95

ELECTRONIC ARTS

One on One $29.95
Mule $29.95

ATARISOFT

Donkey Kong $29
Pole Position $3E

INFOCOM

Zork I, II, III ..... $29 ea.
Suspended $29
Starcross $29
Deadline $29
Introductory Price
Zaxxon $29.95

COMPUTER COVERS

Features heavy duty canvas with vinyl interior - waterproof.

Reg. $15.95 **$6.99**

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

Ordering & Payment Policy

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

Shipping

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

Catalog

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

STOP PLAYING GAMES

NEW Disk Commodore 64

- Calculate odds on HORSE RACES with ANY COMPUTER using **BASIC**.
- SCIENTIFICALLY DERIVED SYSTEM really works. TV Station WLKY of Louisville, Kentucky used this system to predict the odds of the 1980 Kentucky Derby. See *Popular Computing* (February, 1984) for a review of this program. This system was written and used by computer experts and is now being made available to home computer owners. This method is based on storing data from a large number of races on a high speed, large scale computer. 23 factors taken from the "Daily Racing Form" were then analyzed by the computer to see how they influenced race results. From these 23 facts, ten were found to be the most vital in determining winners. NUMERICAL PROBABILITIES of each of these 10 factors were then computed and this forms the basis of this REVOLUTIONARY NEW PROGRAM.
- SIMPLE TO USE: Obtain "Daily Racing Form" the day before the races and answer the 10 questions about each horse. Run the program and your computer will print out the odds for all horses in each race. COMPUTER POWER gives you the advantage!
- YOU GET
  1) Program on cassette or disk.
  2) Listing of BASIC programs for use with any computer.
  3) Instructions on how to get the needed data from the "Daily Racing Form."
  4) Tips on using the odds generated by the program.
  5) Sample form to simplify entering data for each race.

MAIL COUPON OR CALL TODAY

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

Yes, I want to use my computer for FUN and PROFIT. Please send me "Play the Horses" for **$29.95**. Circle the cassette you need: VIC-20, Color Computer, TRS-80, Sinclair Timex 1000, Atari, Commodore 64 (disk or cassette), Apple (disk), IBM (disk).

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

Card No. _____ Exp. date _____

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

START USING YOUR COMPUTER FOR FUN and PROFIT!

When was that article?

Find it fast with *PcDex*™ and *PcDex Quarterly*™!

Magazine resource guides for Commodore 64,™ VIC-20,™ and PET/CBM® computers

Computing magazines getting out of hand? Can't locate a needed software or hardware review? That valuable programming utility? Your magazines are a valuable source of programs, articles, columns, letters and reviews. **PcDex** provides fast, easy access to this valuable resource!

. . . at last!

THE COMPLETE MAGAZINE REFERENCE COMPANION

COMPREHENSIVE Indexes the 12 most popular Commodore and general microcomputer magazines. Over 6000 cross-referenced entries covering January 1982 thru March 1984 *Compute! Compute's! Gazette, Commander, Commodore, Power/Play, Run, Creative Computing,* and more!

VERSATILE Six separate indexes: subject, title, programs, software reviews, hardware reviews, and table of contents. Includes updates and revisions.

VALUE Locate valuable "free" programs for a wide range of games and applications, or reviews that can prevent costly purchasing errors. No more time wasted searching stacks of back issues for that item you can't find. **Go directly to it!** All for $14.95.

Want to stay up-to-date? Subscribe to *PcDex Quarterly*™! Receive 4 issues including an annual cumulation for only $17.95

ALTACOM, *INC.* P.O. Box 19070, ALEXANDRIA, VA 22320

☐ Please send me a copy of **PcDex** at $14.95 plus $2.00 shipping/handling*
☐ Please enter my subscription to **PcDex Quarterly** (4 issues) for $17.95*

Payment enclosed: $_____ (VA residents add 6% sales tax)

Charge my: ☐ VISA ☐ MasterCard

Card no. _____ Exp. _____

Mr./Ms. _____

Address _____

City _____ State/ZIP _____

*Outside the U.S. add $5.00 Please send additional information ☐

FOR FASTER SERVICE CALL 703-683-1442

COMMODORE TRADEMARKS ACKNOWLEDGED DEALER INQUIRIES INVITED

PRODUCT MART

ditto

COPY DISKS AUTOMATICALLY

$39.95

- Copies 99% of currently available Commodore 64 disks.
- Supports 1 or 2 1541 drives.
- Takes approximately 25 minutes.
- Easy to use—menu driven.
- Currently available—future updates $17.
- ditto will even backup ditto.

ORDERS

800-762-5645

Hours 10-6 Mon.-Sat.

CARDINAL SOFTWARE

13646 Jefferson Davis Highway
Woodbridge, VA 22191

GIFTS THAT TEACH

Tutorial cartridges for Commodore 64

- ★ States and Capitals
- ★ Counties (specify state)
- ★ U.S. Mountains, Rivers & Lakes
- ★ Oceans, Seas & Waterways
- ★ Western Hemisphere Countries
- ★ African Countries
- ★ Asian Countries
- ★ European Countries
- ★ Human Skeletal System (Ages 8-16)

Clear graphics and color sprites.
Each cartridge $35.00 + $2.00 S&H
Checks & money orders accepted.

S. M. Thorpe Co.
1229 W. 4000 N.
Pleasant View, Utah 84404

C-64 DS™

Color Digital Oscilloscope *$249*

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

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

call: 206-641-2141

$249

Prices subject to change without notice

SIGN OF THE SPHINX

Aficionados of the arcane, disciples of doom, students of the sinister and masters of the macabre have a new home-in the future dark age of a dying Earth.

Werewolf Software is at it again - serving up full-strength doses of adventure, mystery and weirdness in our new adventure, Sign of the Sphinx.

From the dark shrines of bizarre cults to the ruins of ancient and modern civilizations, you can trace fragments of forgotten knowledge to the final secret at the Crimson Altar.

Text adventure on disk for the Commodore 64. $23 postpaid from:

Werewolf Software
109 Minna Street, Suite 353
San Francisco, CA 94105

HAVE YOU GOT THE 1541 BLUES?

WE'VE GOT THE PERMANENT FIX!!

Send us your sick 1541 and a check for $69.95 plus 5.00 shipping and we'll not only align your drive, we'll fix it so it will STAY THAT WAY.

We guarantee it for 6 months.

We also service the complete COMMODORE COMPUTER LINE with a 48-hour turn-around time.

TRIAD COMPUTERS
3068 Trenwest Dr.
Winston-Salem, NC 27103 (919)765-0433

TUTOR with your COMPUTER

Educational SOFTWARE tutors your child, step-by-step. Teacher-created, educationally sound, *remarkably effective!*
For

- **MATH • LANGUAGE ARTS**
- **SOCIAL STUDIES**
- **PRESCHOOL**

For Apple II+, IIe, Commodore 64, TRS-80 Models III & IV.

DAYBREAK

Send for **FREE** brochure:
DAYBREAK SOFTWARE
(Div. of Educational Activities)
1951 Grand Avenue, Dept. CG
Baldwin, NY 11510

1-800-645-3739 (U.S.) 516-223-4666 (N.Y., AK., HW.)

BOARDWALK BLACKJACK

Play Blackjack Atlantic City Style using 2,4,6 or 8 decks.
BW Blackjack simulates playing with real decks of cards, including shuffling and cutting.

Allows you to practice card counting and betting strategies.
HIT - SPLIT - DOUBLE - STAND.

Commodore version - $19.95
IBM version - $24.95
Add $2.00 for shipping.
NJ residents add 6% sales tax.

To order specify version and send check or money order to:

DETAIL TECHNOLOGIES, INC.
3490 U.S. Route 1
Princeton, NJ 08540

DUANES COMPUTER CORNER

(518) 372-1615 / 393-7330

COMMODORE 64 $195.00

ALL C-64 SOFTWARE

• Access • Academy
• Broderbund • Codewriter
Comprehensive • EPYX
MUSE • HES • Timeworks
Spinnaker • Scarborough
Timeworks • Screenplay

40% Off Mfg. Sugg. List

BMC Monitors		Panasonic Printers	
• JJ1201 Y+	84.95	KXP1090	289.95
• 12EUN Hi Res	89.95	KXP1091	349.95
• 9191V + Color	249.95	KXP1092	399.95

Fuji Diskettes MD1 19.95 box of 10
Super Special MD2 24.95 box of 10

Write for our free catalog
3901 State St., Schenectady, N.Y. 12304

CARTRIDGE-MAKER-64™

Now you can make your own Commodore 64 Video Game and Program Cartridges with CARTRIDGE-MAKER-64.

Simple and Easy to use

1. Insert CARTRIDGE-MAKER-64 into the Commodore 64 cartridge expansion port.
2. Insert a blank CARTRIDGE-64 to be programmed into CARTRIDGE-MAKER-64.
3. Turn on the Commodore 64 and run the CARTRIDGE-MAKER-64 program to create a cartridge of your program or video game.

With CARTRIDGE-MAKER-64 you can:

- copy from cartridge to cartridge
- copy your cartridge to cassette or diskette
- erase and reuse cartridges again

CARTRIDGE-MAKER-64	$ 129.00
CARTRIDGE-ERASER	$ 59.00
CARTRIDGE-64 (16K)	$ 25.00
Cartridge-Maker Kit (Includes: Cartridge-Maker-64, Cartridge-Eraser and 1 Blank Cartridge-64 (16K))	$ 189.00
Purchase of 5 Blank Cartridge-64 (16K each)	$115.00

Phone Orders call: Toll Free 800-233-3251 Ext. 1050
PA Residents call 215-363-8440
VISA-MASTERCARD-C.O.D.-CHECK-MONEY ORDER

CARTRIDGE-MAKER-64 & CARTRIDGE-64 - are TM of Custom Programming Group, Inc.
COMMODORE 64 - is a Registered TM of Commodore Business Machines, Inc.

Dealer and Distributor Inquiries Invited

Cartridge Maker-64 is not intended for piracy of copyright materials.

CPG
Custom Programming Group, Inc.

47 Marchwood Road • Suite 2A • Exton, PA 19341 • (215) 363-8440

ADVERTISERS INDEX

You asked for them . . . CARDCO has them!

LQ/1 & LQ/3 LETTER QUALITY PRINTERS
In response to your demand and need, CARDCO now has available "Commodore-ready "LETTER QUALITY PRINTERS". Just plug them in and print.

Offering standard friction-feed and optional tractor-feed, the CARDCO PRINTERS come complete with built-in interfacing for all Commodore Personal Computers, as well as compatible input for PC, PC jr., TRS-80 and other personal computers.

LQ/1 13" carriage, 15 CPS

LQ/3 11" carriage, 13 CPS

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

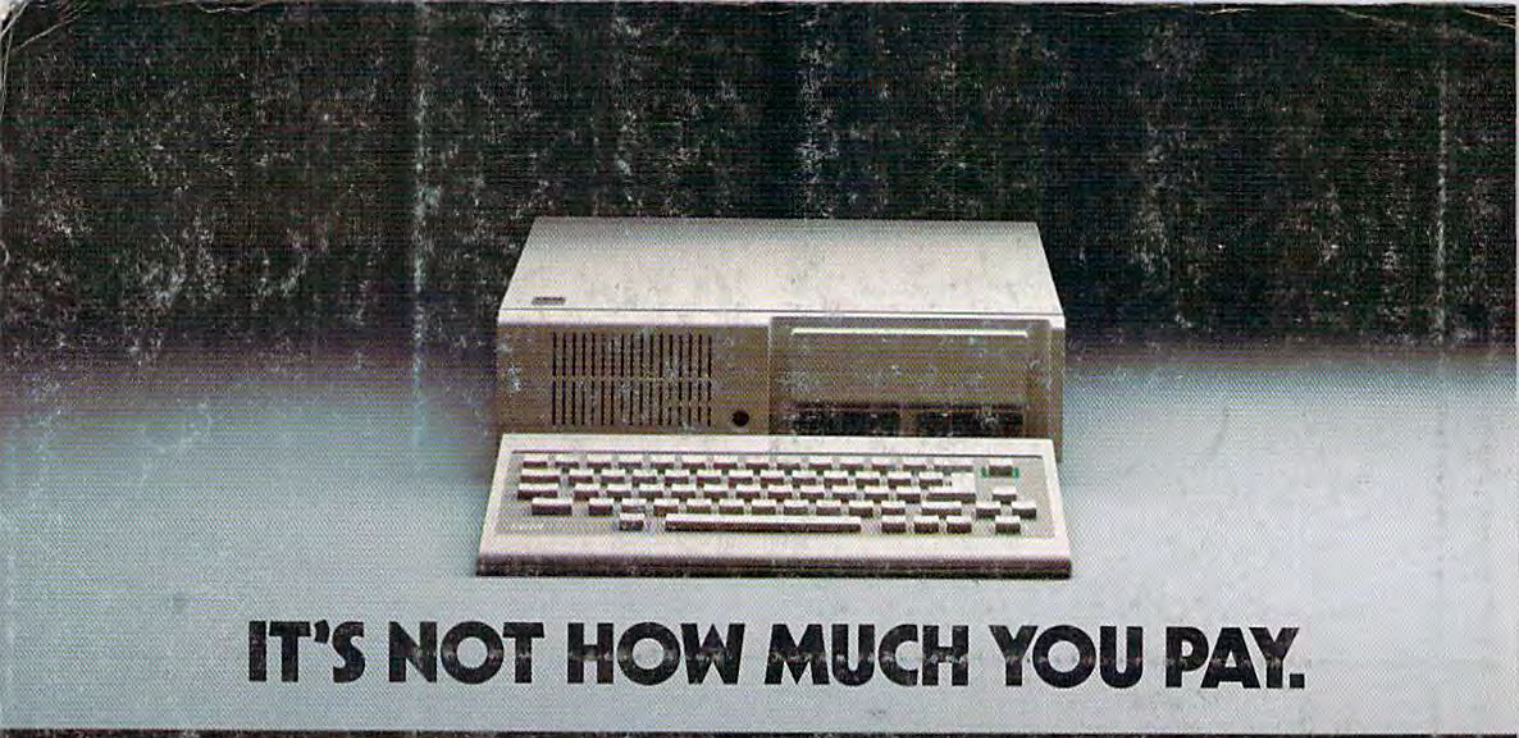
"WRITE NOW" WORD PROCESSOR SOFTWARE
An excellent time saver, CARDCO OFFERS THE CØ2 "Write Now" program with built-in 80 column display. You see exactly what will print. All special codes can be transmitted to printers maintaining justification. Easy full-screen editing; works with any printer.

"MAIL NOW" MAILING LIST SOFTWARE
CARDCO D/Ø1 quickly (in memory) sorts by zip, category, name and state; fully compatible with "WRITE NOW". Other fine features include: User-oriented; menu-driven operation; each disk supports 600 entries. Format can print single, double or triple labels across.

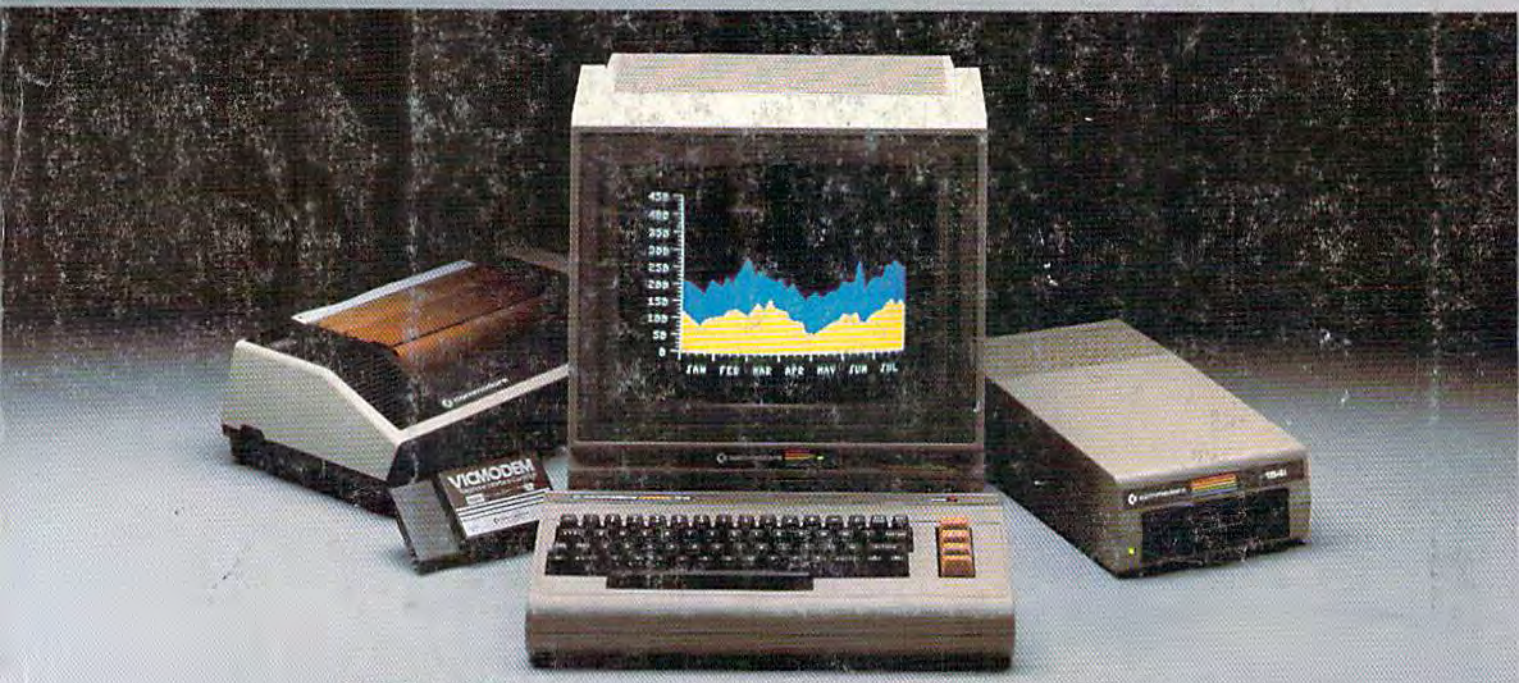
All Cardco Products are available at your local dealers. Write for illustrated literature.

cardco, inc.

300 S. Topeka Wichita, Kansas 67202 (316) 267-6525
"The world's largest manufacturer of Commodore accessories."



IT'S NOT HOW MUCH YOU PAY.



IT'S HOW MUCH YOU GET.

The computer at the top has a 64K memory.

It has the initials I, B, and M. And you pay for those initials—about $669.

The Commodore 64™ has a 64K memory.

But you don't pay for the initials, you just pay for the computer: $215. About one third the price of the IBM PCjr.™

The Commodore 64 also has a typewriter-type keyboard with 66 typewriter-type keys. (Not rubber chicklet keys like the IBM PCjr.)

It has high resolution graphics with 320 x 200 pixel resolution, 16 available colors and eight 3-dimensional sprites.

It has 9-octave high fidelity sound.

The Commodore 64 is capable of running thousands of programs for home and office. And if you add a printer or color monitor, disk drive and a modem—all together it just about equals the price of the IBM PCjr all alone. With no peripherals.

So you can buy a computer for a lot of money.

Or buy a lot of computer for the money.

COMMODORE 64

IT'S NOT HOW LITTLE IT COSTS,
IT'S HOW MUCH YOU GET.